

Universidad de las Ciencias Informáticas

Facultad 7



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

**Diseño del Módulo de Facturación del Sistema de
Información Hospitalaria alas HIS**

Autores: Danais Castillo Borrero

Diuber Estanque Díaz

Tutores: Ing. Adnier Guerrero Barea

Ing. Yenisel Valido Pérez

Ciudad de la Habana, julio de 2010

“Año 52 de la Revolución”

*“La inteligencia consiste no sólo en el conocimiento,
sino también en la destreza de aplicar los conocimientos en la práctica.”*

Aristóteles

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 2 días del mes de julio del año 2010.

Danais Castillo Borrero

Firma del autor

Diuber Estanque Díaz

Firma del autor

Adnier Guerrero Barea

Firma del tutor

Yenisel Valido Pérez

Firma del tutor

Datos de Contacto

Síntesis del tutor: Ing. Yenisel Valido Pérez

Correo electrónico: yenisel@uci.cu

Graduada de Ingeniería Informática en el Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE) y la Universidad de Cienfuegos “Carlos Rafael Rodríguez”. Profesora de la Universidad de las Ciencias Informáticas, pertenece al departamento de Ingeniería y Gestión de Software de la facultad 7. Posee categoría docente de profesor asistente. En cuatro años de labor docente, ha impartido las asignaturas de Ingeniería de Software y se desempeña como jefa de la misma en la facultad siete. Profesora del Área Temática Sistemas de Apoyo a la Salud.

Síntesis del tutor: Ing. Adnier Guerrero Barea

Correo electrónico: abarea@uci.cu

Graduado en la Universidad de Ciencias Informáticas como Ingeniero en Ciencias Informáticas. Especialista del Centro de Desarrollo de la Informática Médica (CESIM) y profesor adjunto a la facultad 7.

Resumen

El presente trabajo tiene como objetivo diseñar el módulo de Facturación del Sistema de Información Hospitalaria alas HIS; a partir de las necesidades existentes en los hospitales de controlar las prestaciones asociadas a un paciente y gestionar el cobro de las mismas. Este sistema mejora la gestión de la información en el área de facturación de las instituciones hospitalarias.

Para diseñar este sistema se utilizó la metodología de software Proceso Unificado de Desarrollo (RUP), la cual se apoyó en el Lenguaje de Modelado Unificado (UML) y en la Notación para el Modelado de Procesos del Negocio (BPMN); y tuvo como modelo rector del proceso de desarrollo al Modelo de Capacidad y Madurez (CMMI). Además, para la creación del prototipo no funcional del sistema fue utilizado el entorno de desarrollo Eclipse, unido a Java como lenguaje de programación y PostgreSQL como Sistema Gestor de Base de Datos. Junto a estos elementos se incorporan frameworks, componentes, motores de reglas de negocio y sistemas para la administración de bases de datos. Todos estos elementos giran alrededor de la arquitectura de software Modelo Vista Controlador (MVC) propuesta para el sistema.

El módulo de Facturación promete muchos beneficios para los usuarios, y la optimización del trabajo en los procesos de facturación. Se espera un aumento en la velocidad y precisión de la gestión de costes y facturas. Y además, se podrá mejorar el manejo de los datos y la generación de reportes para las predicciones financieras y la planificación hospitalaria.

Palabras Claves:

Sistema de Información Hospitalaria, Facturación, Factura.

Tabla de Contenidos

RESUMEN	IV
TABLA DE CONTENIDOS	V
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1. CONCEPTOS BÁSICOS RELACIONADOS CON EL DOMINIO DEL PROBLEMA	6
1.2. SISTEMAS AUTOMATIZADOS QUE INCLUYEN EN SU SOLUCIÓN EL MÓDULO FACTURACIÓN	7
1.2.1. Sistemas nacionales.....	7
1.2.2. Sistemas internacionales.....	8
1.3. TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR	12
1.3.1. Aplicaciones web	13
1.3.2. Tecnologías libres.....	13
1.3.2.1. Capa de presentación.....	14
1.3.2.2. Capa de negocio	15
1.3.2.3. Capa de modelo	17
1.3.2.4. Tecnologías horizontales.....	18
1.3.3. Herramientas	18
1.3.4. Sistema de gestión de base de datos	19
1.3.5. Herramienta de modelado	19
1.3.6. Gestión de proyecto.....	20
1.3.7. Patrón de arquitectura	23
1.3.8. Entornos distribuidos	23
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	25
2.1. OBJETO DE ESTUDIO	25
2.1.1. Objetivos estratégicos de la organización	25
2.1.2. Flujo actual de los procesos involucrados en el campo de acción.....	26
2.1.3. Causas que originaron la situación problemática.....	27
2.1.4. Propuesta del sistema	28
2.2. INFORMACIÓN QUE SE MANEJA	28
2.3. MODELADO DEL NEGOCIO	30

2.3.1.	Definición de roles	30
2.3.2.	Diagramas de procesos del negocio	31
2.4.	ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA	33
2.4.1.	Requisitos funcionales del sistema.....	33
2.4.2.	Requisitos no funcionales del sistema	34
2.4.2.1.	Usabilidad.....	35
2.4.2.2.	Seguridad	35
2.4.2.3.	Rendimiento	35
2.4.2.4.	Soporte	36
2.4.2.5.	Hardware	37
2.4.2.6.	Software	37
2.4.2.7.	Restricciones de diseño.....	37
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.....	38
2.5.	DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA	38
2.5.1.	Definición de actores	39
2.5.2.	Diagrama de casos de uso	39
2.5.2.1.	Diagramas de casos de uso del sistema	39
2.5.2.2.	Descripción textual de casos de uso significativos del sistema	40
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	45	
3.1.	DESCRIPCIÓN DE LA ARQUITECTURA. FUNDAMENTACIÓN	45
3.2.	ANÁLISIS DE POSIBLES IMPLEMENTACIONES, COMPONENTES O MÓDULOS YA EXISTENTES Y QUE PUEDAN SER REHUSADOS	46
3.3.	MODELO DE ANÁLISIS	46
3.3.1.	Diagramas de clases de análisis	46
3.4.	MODELO DE DISEÑO	48
3.4.1.	Fundamentación del uso de patrones	48
3.4.2.	Estructura del diseño	48
3.4.3.	Diagramas de clases del diseño.....	50
3.4.4.	Descripción de las clases	51
3.4.5.	Diagramas de interacción	53
3.5.	MODELO DE DATOS	56

3.5.1. Descripción de las tablas.....	57
CONCLUSIONES.....	60
RECOMENDACIONES	61
REFERENCIAS BIBLIOGRÁFICAS.....	62
BIBLIOGRAFÍA.....	64
GLOSARIO DE TÉRMINOS	66
ANEXOS.....	67
1. ANEXO: DIAGRAMAS DE CLASES DEL DISEÑO.....	67
2. ANEXO: DIAGRAMAS DE SECUENCIA DEL DISEÑO	69

Introducción

El desarrollo informático ha sido considerado como el principal impulsor de este milenio, no sólo porque ha roto barreras del conocimiento antes intocables, sino también porque ha impulsado el avance de las ciencias tradicionales hacia nuevos caminos. Este desarrollo puso punto final a los esquemas clásicos de la organización, trazando nuevas pautas hacia un control de la información abierto, dinámico y orientado a procesos. Lo que trae como consecuencia que en la actualidad sea la informática, la herramienta tecnológica más influyente en la economía y la sociedad.

La medicina como rama social dirigida al mejoramiento físico y psicológico de los individuos, no ha estado exenta a este proceso de innovación. Con el transcurso del tiempo y el surgimiento de nuevos adelantos y necesidades médicas, se ha trabajado en función de alcanzar soluciones imprescindibles para obtener la satisfacción y realización personal del paciente.

Las opciones viables que ofrece la tecnología; y las semejanzas existentes entre la medicina y la informática, al ser estas dos ciencias que centran su trabajo en mejorar la calidad de vida; permite que ambas ramas encuentren un punto común de intercambio. La confluencia de estas áreas evoluciona en un nuevo campo interdisciplinario y cada vez más, el término “informática” se ubica en una posición adyacente a la medicina, hasta convertirse en ciencia la informática aplicada a la salud.

El aumento en número y complejidad de las especialidades médicas, la necesidad de mejorar las prestaciones al paciente y los problemas en el manejo del gran cúmulo de información, fueron los detonadores del uso de la informática en todos los sectores de los servicios sanitarios. Es por ello, que el uso del software como herramienta de apoyo al médico en la toma de decisiones se convirtió en un hecho para la práctica clínica.

De este modo han sido creados en todo el mundo sistemas informáticos especializados en distintas áreas de la salud, y estos a su vez han automatizado desde un documento de historia clínica, hasta complicados resultados de imágenes de alta resolución. Ya hoy, se pueden encontrar indistintamente soluciones de software en consultorios médicos, policlínicos de urgencias, clínicas u hospitales; dejando estas a su paso, profundas huellas de calidad y excelencia en su funcionamiento.

Sin embargo, muchas veces estos sistemas no integran en sí mismos una amplia variedad de especialidades y debido a que la infraestructura de un hospital es muy abarcadora, fueron engendradas nuevas dificultades al tener que mantener un conjunto de sistemas realizando disímiles

tareas a la vez. Esta es la causa que originó una nueva alternativa, los Sistemas de Información Hospitalaria (HIS).

Un HIS es un sistema que tiene como núcleo central para su ejecución a los hospitales, instituciones donde se maneja constantemente un gran flujo de información de cada uno de los pacientes atendidos. El objetivo principal de esta aplicación es realizar un control y manejo de toda la documentación e información que allí se administra, pues la misma genera una inagotable fuente de conocimientos que generalmente se pierde porque está muy dispersa, o no está disponible cuando se necesita.

Son muchos los países del mundo que han integrado como solución informática en la salud a los HIS. Cuba, como parte de las potencias tecnológicas en la medicina, se está introduciendo en este proceso de transformación. Por lo que la Universidad de las Ciencias Informáticas (UCI), institución de estudios informáticos de esta nación; ha tomado la iniciativa de realizar un Sistema de Información Hospitalaria.

Este sistema incluye un grupo de módulos que automatizan áreas dedicadas al Almacén Hospitalario, Consulta Externa, Hospitalización, Emergencias, Admisión, Laboratorio, Anatomía Patológica, Epidemiología, Banco de Sangre, Citas, Bloque Quirúrgico, Enfermería, Farmacia y Estadísticas. Estos módulos integrados como un todo único, administran información clínica, administrativa, epidemiológica y científica importante para mejorar la calidad de la atención al paciente en el hospital.

La automatización de las prestaciones ofrecidas al paciente es muy importante para llevar a cabo un servicio de excelencia, buscando siempre el máximo de calidad en el proceso de atención. Esta es la causa por la que es frecuente que los sistemas dirijan su desempeño a especialidades que trabajan directamente con el individuo. Sin embargo, dentro de una entidad hospitalaria e invisible para el hospitalizado, existen áreas que son pilares fundamentales para el sustento de un hospital. Enmarcadas en este círculo se pueden encontrar la Contabilidad, Estadísticas, Planificación y la Facturación, siendo esta última, de relevante importancia por las distintas variables e implicaciones que trae para las instituciones.

El sector de facturación, es el encargado de controlar administrativamente, todas las operaciones financieras que se realizan en el hospital alrededor del paciente, partiendo de la llegada de este al centro en busca de una atención clínica.

En instituciones donde para recibir los servicios médicos estos deban ser pagados, una de las principales tareas de la facturación es determinar los gastos realizados por el paciente en la institución, así como el modo de pago de los mismos. Por ello, previamente a la atención hospitalaria del individuo se establece con este, la forma en que va a realizar el pago; dándosele la oportunidad de hacerlo a través de una empresa proveedora de seguros médicos y en caso de que no pertenezca a ninguna de ellas, se le permite pagar en efectivo o a través de un plan de plazos. En esta última opción, se tiene en cuenta si el ingreso monetario que recibe el paciente puede sufragar los gastos asumidos.

Luego de haber determinado el capital que puede dedicar el paciente a los servicios médicos, se procede a la confección de la factura. Esta varía en tipo teniendo en cuenta su valor en factura pro-forma, copia y duplicado; clasificándose también en ordinaria, rectificativa y recapitulativa, de acuerdo con la operación y función que va a realizar.

A medida que se cumple la ayuda clínica, los especialistas gestionan las tarifas de cada prestación o servicio recibido por el paciente en el hospital, desde los costos de las citas médicas, productos farmacéuticos y procedimientos, hasta elementos logísticos. Enviándose al final de la atención, la factura de gastos a la empresa proveedora de seguros o al paciente en ausencia de esta.

Todas las acciones dirigidas al paciente en un hospital, son previamente valoradas por un médico u otro especialista, quien de acuerdo con su experiencia, calificación y especialidad, recibe honorarios financieros por los resultados obtenidos. Estos honorarios también deben ser procesados por el hospital a través de la gestión de facturas.

El área de facturación es además, quien gestiona las indemnizaciones del hospital. Es posible, que durante los procedimientos realizados al paciente ocurra una negligencia médica o un error no previsto, en estos casos el centro debe ser capaz de resarcir los daños e indemnizar a la persona por las afectaciones ocasionadas.

Evidentemente, los procesos de facturación requieren de un gran esfuerzo y precisión, sin embargo, un elevado número de instituciones médicas realizan estas actividades utilizando métodos tradicionales que ya resultan poco efectivos. Esto trae como consecuencia que los mismos se realicen de forma engorrosa dada la cantidad de información acumulada, obteniéndose al final resultados poco productivos y un bajo rendimiento en esta área de la institución.

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.

Un elemento significativo, es que las labores de facturación son realizadas por un grupo de especialistas quienes diariamente recolectan documentos y datos de los gastos de los pacientes en las disímiles áreas del hospital; teniendo en cuenta las distintas prestaciones ofrecidas. Esta actividad además de extenuante deja el campo abierto a los posibles errores humanos precisamente en un sector que requiere de total precisión.

La buena facturación, es también importante para el paciente, quien se ve afectado cuando el trabajo se realiza manualmente, pues 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. En otras circunstancias, donde el paciente recibe la atención médica gratuita el proceso no pierde importancia; para la institución continúa aportando los beneficios antes mencionados y en el caso particular del atendido, es una vía irrefutable para conocer los gastos ocasionados y sufragados.

Dada esta situación se ha determinado como **problema a resolver**: ¿Cómo facilitar la gestión de la información relacionada con los procesos en el área de facturación de las instituciones hospitalarias?

En correspondencia con el problema, el **objeto de estudio** está enfocado en el proceso de gestión de la información de las instituciones hospitalarias. Para la realización del estudio del problema planteado se ha delimitado como **campo de acción**, al proceso de gestión de la información en el área de facturación de las instituciones hospitalarias.

Para solucionar los problemas mencionados, se define como **objetivo general** diseñar el módulo de Facturación del Sistema de Información Hospitalaria alas HIS, para facilitar la gestión de la información en esta área de las instituciones hospitalarias.

Con la intención de alcanzar el objetivo propuesto se han trazado distintas **tareas de la investigación**, entre estas se encuentran:

- Aplicar la arquitectura y tecnologías definidas para el Sistema de Información Hospitalaria alas HIS.
- Modelar los procesos del negocio definidos en la investigación utilizando la metodología de desarrollo del software RUP.
- Definir los requerimientos del software para el módulo Facturación.
- Realizar el análisis del módulo de Facturación, según la metodología de desarrollo de software RUP.
- Proponer un diseño flexible basado en estándares y patrones.
- Elaborar el prototipo no funcional del módulo de Facturación para el Sistema de Información Hospitalaria alas HIS.

El presente trabajo de diploma presenta un carácter investigativo y está concebido en tres capítulos como se expone a continuación:

- Capítulo 1: Fundamentación Teórica: Se estudian los conceptos esenciales y se analizan los fundamentos teóricos del tema, las características de la facturación, los problemas que motivan la investigación, los sistemas ya existentes en el campo y los aportes de la facturación electrónica como resultado de la informática en la salud.
- Capítulo 2: Características del Sistema: Se detallan las características generales y específicas del sistema, definición de los procesos que intervienen en el mismo, actores, trabajadores, procesos del negocio; requisitos funcionales y no funcionales; actores y casos de uso del sistema.
- Capítulo 3: Análisis y Diseño del Sistema: Se describe el sistema a través de diagramas de clases del análisis y diseño. Propuesta del estándar a aplicar y de la estrategia para su aplicación. Se realiza la presentación de un prototipo no funcional de la solución.

Capítulo 1: Fundamentación Teórica

La necesidad de la creación de un Sistema de información Hospitalaria (HIS), y de incluir dentro de este, un área dedicada a la facturación, puede estar afectada por la existencia de acepciones desconocidas. El siguiente capítulo contiene un breve compendio de terminologías de uso indispensable en los procesos de facturación. Además, se realiza una descripción de las distintas herramientas de diseño y desarrollo que serán utilizadas para apoyar la constitución del sistema; así como de las metodologías de trabajo y patrones de arquitectura y diseño que servirán de guía para la formación estructural del mismo.

1.1. Conceptos básicos relacionados con el dominio del problema

El término **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. (1)

La **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. (2)

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

- **Factura ordinaria:** aquella que documenta una operación comercial o un servicio.
- **Factura rectificativa:** aquella que documenta correcciones de una o más facturas anteriores, o bien devoluciones de productos, envases y embalajes o comisiones por volumen.
- **Factura recapitulativa:** 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.

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. (3)

La **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.

1.2. Sistemas automatizados que incluyen en su solución el módulo Facturación

Al ser la facturación un proceso que requiere de precisión en una institución hospitalaria, han sido muchos los sistemas informáticos que han incluido soluciones que apoyan en mayor o menor medida, la gestión de facturas. La mayoría de estos son productos internacionales que se especializan más en esta área, sin embargo, también existen productos nacionales que de alguna manera proponen una alternativa de apoyo.

1.2.1. Sistemas nacionales

Gestión Económica (BKMIS): Sistema económico compuesto por tres módulos principales que pueden operarse de manera integrada o de forma independiente. (4)

- Contabilidad General Integral: Cuenta con el manejo de activos fijos, cierre de períodos normales, caja y banco, préstamos bancarios recibidos y otorgados.
- Administración Empresarial: Incluye un inventario de almacenes, ofertas, facturación configurable, cuadro diario, entre otras opciones.

- Recursos Humanos y Nómina: Cuenta con plantilla ocupacional, control de datos de nómina, información de cuadros, retenciones, entre otros. (5)

Gestión Económica (VINDSOR): Sistema financiero que proyecta, maneja y rastrea con rapidez una amplia gama de futuras transacciones bancarias, registra las recepciones relacionadas con cobros y pagos y administra el efectivo, así como la ejecución del presupuesto aprobado. (6)

Brinda las prestaciones siguientes:

- Proyecciones financieras: Proyecta, maneja y rastrea con rapidez una amplia gama de futuras transacciones bancarias.
- Distribución financiera: Registra recepciones relacionadas con los cobros y pagos.
- Administración de efectivos: Captura transacciones bancarias relacionadas con los registros de efectos por cobrar, por pagar, ingresos bancarios, entre otros.
- Presupuesto: Detalla el presupuesto aprobado y obtiene informes a diferentes niveles. (7)

Estos sistemas fueron desarrollados por la Empresa Nacional de Software (DESOFT S.A.), y están actualmente en uso en la Empresa de Suministros Médicos de Cuba (EMSUME). Los mismos son utilizados para apoyar el proceso de gestión de la facturación por compraventa de productos médicos dentro de esta organización, y aunque no se dedican a la facturación por servicios; han servido como punto de apertura en el proceso de apoyo para la gestión económica-financiera en la salud.

1.2.2. Sistemas internacionales

Mediplus 9.5: Es un software que incluye un módulo para la facturación. El mismo, utiliza como gestor de base de datos MS Access y está basado en la arquitectura cliente-servidor. Este software permite la gestión de las facturas realizadas en una institución hospitalaria por concepto de servicios de urgencias, consulta, diagnóstico, exámenes, farmacia y logística. Permite la gestión de factura online con detalles de pagos totales, reembolsos, así como notas de crédito. También permite el registro de pacientes con detalles de los datos personales del mismo, además de la admisión, consultor, cirujano y otras modalidades; con el objetivo de obtener los datos para la gestión de cobros. Sus facilidades permiten una mejor atención al paciente, asignando para la tarea menos personal y obteniendo resultados mucho más alentadores en términos de eficiencia y precisión.

SAM Facturación Médica v 7.0: Es una aplicación de escritorio desarrollada para consultorías médicas. La misma permite la gestión de las historias clínicas de los pacientes, así como de las facturas de los mismos. Este software aunque bastante abarcador en cuanto a operaciones, sólo es

Capítulo 1: Fundamentación Teórica

capaz de manipular archivos de un médico. El sistema permite obtener informes de los pacientes y de las transacciones de estos, incluyendo además reportes de las operaciones de las empresas aseguradoras. Contiene funcionalidades para la gestión de las cuentas por cobrar, de las facturas, estados de cuentas, informes financieros y registros de las entidades aseguradoras. SAM Facturación Médica, además, tiene una base de datos robusta que soporta una gran cantidad de fichas médicas.

SAM Médico Más v 7.0: Es una aplicación de escritorio que contiene las mismas funcionalidades que el software SAM Facturación Médica v 7.0. En cambio, este sistema está realizado para organizaciones que dispongan de más de un médico, permitiendo así, el registro de 99 usuarios como máximo. Entre las funcionalidades que ofrece el sistema, le son agregadas la posibilidad de pagar al médico un por ciento del monto de facturas, si se le va a pagar por visitas realizadas o por un salario estándar.

Consulta Médica v 1.21: Es un software que permite llevar la gestión de pacientes, historias clínicas, citas y permite la elaboración de distintos reportes de información hospitalaria. Es un sistema pequeño, con opciones de interfaz limitada y está preparado para funcionar en una red local, de modo que varios médicos puedan compartir la misma base de datos. Incluye un módulo de facturación minimizado, capaz de gestionar la realización y edición de las facturas realizadas.

Antamedia Medical Software v 1.1.0: Sistema hospitalario que permite la programación de citas médicas, la obtención de registros médicos electrónicos y la facturación. Gestiona eficientemente la información perteneciente a cada paciente de la institución hospitalaria donde se encuentra instalado. Es muy factible para la calendarización de las citas de los pacientes, incluyendo opciones para el recordatorio automático de las mismas. El módulo de facturación del software permite la gestión de pagos, con facilidades en el tipo de moneda para realizar la retribución. Además, permite la administración de gastos e ingresos de la institución, pudiéndose obtener reportes con información detallada de las cuentas, así como del flujo de efectivo.

El sistema puede ser utilizado en distintos tipos de instituciones médicas incluso si son de gran tamaño. Es capaz de proporcionar múltiples niveles de seguridad, con la opción de gestionar las cuentas de usuarios y los niveles de acceso. Proporciona una interfaz sencilla, con una alta usabilidad, permitiendo que sea muy fácil de aprender y utilizar, tanto en una oficina, como en varias instituciones distribuidas por una ciudad.

Net Clínicas: Sistema de gestión clínica muy utilizado en la actualidad. El mismo puede ser utilizado en cualquier centro de atención sanitaria, es configurable y adaptable a las necesidades de

los usuarios y el cliente. Permite gestionar distintas áreas que van desde el control de historias clínicas hasta el registro de facturas. Contiene una agenda de excelencia para archivar citas y eventos, gestiona toda la documentación del centro, tiene editores de formularios y bases de datos de los pacientes. Dispone del envío automático de mensajes recordatorios de los eventos a realizar.

Net Clínicas permite la eficiente facturación de prestaciones médicas en sus distintas modalidades. Permite la gestión de mutuas, de pagos a través de estas, de pagos anticipados y abonos, así como de recibos y facturas. Puede realizar el cobro de facturas por domiciliación.

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.

Tesis CM: Es una pequeña aplicación de escritorio, dedicada a solucionar las problemáticas de los centros hospitalarios. Este sistema de información permite la elaboración de presupuestos, la programación de agendas, la confección de informes con datos de los pacientes atendidos y las historias clínicas de estos. Dispone de un área dedicada a la elaboración de presupuestos, emisión de facturas y estadísticas de las actividades tanto económicas como asistenciales. El módulo de facturación tiene funcionalidades dedicadas al:

- Registro de conceptos a facturar.
- Emisión de facturas diferidas.
- Registro de cobros.
- Registro de honorarios.
- Liquidación y eliminación de facturas de honorarios.
- Traspaso de apuntes contables.

Salus: Software de gestión hospitalaria definido para cualquier centro de salud, el mismo tiene dos versiones, una dedicada a los hospitales y clínicas y la segunda versión creada para los centros médicos y policlínicos. Está concebido para tres grandes áreas de gestión: agendas, historia clínica y facturación. En el área de facturación el sistema tiene funcionalidades que le permiten la facturación a particulares y mutuas, el control de deudas, presupuestos y costes, el reporte de informes económicos

Capítulo 1: Fundamentación Teórica

y de rentabilidad, y el control de stocks y proveedores. Además, el sistema permite la liquidación de comisiones automáticas a los profesionales del centro y a consultores externos.

Michigan: Sistema de información hospitalario, de gran utilización en Argentina. El mismo cuenta con una serie de módulos dedicados a la gestión hospitalaria desde el punto de vista administrativo, facilitando en gran medida resolver las dificultades y ralentizaciones que causa la realización manual de estos procesos. Entre sus módulos se encuentran los dedicados a la gestión de las historias clínicas, turnos, admisiones, estadísticas, laboratorio, entre otros.

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.

Todos los sistemas existentes en la actualidad son muy buenos, y capaces de gestionar la facturación parcial o totalmente en distintos centros hospitalarios. Sin embargo, estos tienen propiedades que imposibilitan su implantación, en determinadas instituciones que también necesitan la automatización de sus áreas o que los vuelven ineficientes bajo determinadas condiciones. La siguiente tabla muestra algunas de las características que dificultan la utilización de dichas aplicaciones.

Nombre	Software Propietario	Aplicación Web	Aplicación Escritorio	Plataforma	Año de lanzamiento en el mercado.
Mediplus 9.5	X		X	Windows	2006
SAM Facturación Médica V7.0	X		x	Windows	2009
Consulta Médica v1.21	Shareware		x	Windows	2001
Antamedia Medical Software 1.1.0	X		x	Windows	2007
Net clínicas	X		x	Windows	2009
SAM Médico Más V 7.0	X		x	Windows	2009
TESIS CM	X		x	Windows	2009
Salus	X		x	Windows	1995

Capítulo 1: Fundamentación Teórica

Michigan	X		x	Windows	1998
Gestión económica: VINDSOR	N/D	N/D	N/D	Windows	N/D
Gestión económica: BKMIS	N/D	N/D	N/D	Windows	N/D

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

Todos los sistemas como ya se ha expuesto (Ver tabla 1) son aplicaciones de escritorio, esto tiene como desventaja que el despliegue de las mismas sea engorroso, pues deben ser instaladas en cada estación de trabajo donde vayan a ser utilizadas. Estas aplicaciones están fuertemente ligadas a una plataforma de trabajo, provocando que los sistemas carezcan de portabilidad.

Por otro lado, los sistemas de facturación que actualmente se encuentran en el mercado, fueron creados bajo licencia propietaria, lo que acarrea consigo que para la obtención de estos, se exijan altos costes; que no se limitan sólo al pago del software por cada puesto de trabajo donde se vaya a instalar, sino que se incluye además, el precio de la licencia del gestor de base de datos y otros servidores necesarios.

Además, estas aplicaciones en su mayoría no gestionan toda la información necesaria en la facturación; ya sea la gestión de facturas, honorarios, manejo de reportes, control del balance financiero u otras funcionalidades. También es importante destacar, que algunos de estos sistemas se dedican únicamente a la gestión de facturas, o incluyen la facturación en un sistema que gestiona otras áreas, pero sin integrar todas las ramificaciones sectoriales que pueden encontrarse en una institución hospitalaria. Esto trae como consecuencia que la facturación sea vista como una actividad independiente, cuando esta es en realidad, un proceso que se desarrolla como un todo que está en estrecha relación con todos los elementos de un centro hospitalario.

1.3. Tendencias y tecnologías actuales a considerar

El estudio de las tendencias mundiales y las tecnologías informáticas actualmente existentes, es un paso importante en el inicio del desarrollo de un proyecto. Es este el momento de definir cuáles serán las herramientas y técnicas a utilizar, para crear un software con la calidad reglamentada y que cumpla con las condiciones predeterminadas por el equipo.

La situación actual de los sistemas informáticos dirigidos a la facturación, exigen la creación de soluciones que estén libres de las licencias propietarias, de modo que con el uso del software libre los

costes sean minimizados, optimizando los resultados. Además, se necesita el uso de tecnologías que garanticen que el software sea configurable y multiplataforma.

1.3.1 Aplicaciones web

Desarrollar un proyecto como una aplicación web, ofrece una diversidad de ventajas que le permiten al mismo cumplir con muchas de las condiciones que se necesitan para un buen funcionamiento. Una aplicación web no es más que un sistema usado por usuarios a través de la red hasta acceder a un servidor web, utilizando una arquitectura cliente/servidor. Esto brinda como facilidad que el sistema pueda ser actualizado cuanto sea necesario de modo rápido y limpio, sin tener que modificarlo en cada estación de trabajo.

En una aplicación web el proceso de instalación es muy sencillo, pues se realiza sólo en el servidor y los usuarios tendrán acceso a este a través del navegador sin ocuparse de todo el proceso. Estas aplicaciones además, son muy provechosas ya que los sistemas no dependen de una plataforma específica, por lo que las mismas pueden variar sin complicaciones.

Una aplicación web tiene también sus desventajas respecto a las aplicaciones de escritorio, pues son sistemas que necesitan de una conexión permanente y veloz a la red, situación que en todos los casos no se asegura. Las mismas tampoco se realizan en tiempo real, pues cuando se efectúa el intercambio con el usuario necesitan de un tiempo de respuesta del servidor.

1.3.2 Tecnologías libres

El coste de producción, despliegue y utilización de un sistema, es una característica importante que determina la factibilidad del mismo. Por ello, actualmente este tema es ampliamente tratado, ya que el software muy costoso aún si es el mejor, generalmente no es accesible para todo el que lo necesite por el coste de las licencias propietarias. Por esta razón, el desarrollo de las tecnologías libres es analizado con profundidad; y la construcción de las aplicaciones utilizando herramientas, plataformas y lenguajes que pertenezcan a este movimiento tecnológico está brindando muy buenas perspectivas. De esta manera, la propuesta de realización de una solución de software utilizando el lenguaje liberado Java brinda excelentes ventajas; teniendo en cuenta además, que este es un lenguaje orientado a objetos, de programación multiplataforma y que en comparación a otros lenguajes tiene disímiles facilidades.

1.3.2.1. Capa de presentación

Los componentes del nivel de usuario, proporcionan la interfaz visual que los clientes utilizarán para ver la información y los datos. En este nivel, los componentes son responsables de solicitar y recibir servicios de otros componentes del mismo nivel o del nivel de servicios de negocio. Es muy importante destacar que, a pesar de que las funciones del negocio residen en otro nivel, para el usuario es transparente la forma de operar.

Java Server Faces (JSF)

Java Server Faces (JSF) es un framework para la construcción de interfaz de usuario que realiza la programación a través de componentes y basada en eventos. Es muy flexible, permitiendo incluso crear componentes personalizados. Resuelve los problemas de duplicidad al pulsar un botón dos veces y posee gran cantidad de implementaciones y componentes. Permite desarrollar rápidamente aplicaciones de negocio dinámicas sin necesidad de utilizar JavaScript.

RichFaces

Es una librería de componentes ricos para JSF, y además posee un framework avanzado para la integración sencilla de las funcionalidades Ajax dentro del desarrollo de las aplicaciones del negocio. Es capaz de trabajar con cualquier implementación de JSF y con la mayoría de sus bibliotecas de componentes sin ninguna configuración adicional. RichFaces incluye el soporte a skins (estilos), así como los beneficios de JSF, ciclo de vida, validaciones y conversiones.

Ajax

Ajax es el acrónimo para Asynchronous JavaScript + XML, esta es una tecnología asíncrona donde, los datos adicionales se requieren al servidor y se cargan en segundo plano sin la necesidad de recargar completamente la página. JavaScript es el lenguaje interpretado con el que normalmente se efectúan las funciones de llamada de Ajax. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML. Esta técnica es válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, pues está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

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. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones automáticas al servidor y controlar cualquier evento de usuario. Ajax4jsf permite dotar a las aplicaciones JSF de un contenido mucho más profesional con muy poco esfuerzo. (8)

Facelets

Es un framework web de código abierto bajo la licencia Apache y tecnología alternativa de controlador de vista de JSF. Facelets soporta todos los componentes de interfaz de usuario JSF y construye su propio árbol de componentes, lo que refleja el punto de vista de una aplicación de este tipo. El mismo, permite que Java Server Pages (JSP) y JSF puedan funcionar conjuntamente en una misma aplicación web. Además, su trabajo está basado en plantillas y tiene una fácil composición de componentes. Incluye funciones para expresiones, así como también permite la creación de librerías de componentes. Facelets permite en general el desarrollo amigable para el diseñador gráfico.

XHTML

El Lenguaje de Marcado de Hipertexto Extensible (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 herramientas basadas en XML. XHTML extiende HTML 4.0 combinando el mostrar datos con las sintaxis de HTML y describir los datos con las de XML. Su objetivo es avanzar de tal forma que se pueda lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas.

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.

1.3.2.2. Capa de negocio

Seam

Seam es una potente plataforma de desarrollo de código abierto realizada para construir aplicaciones de Internet en Java. La misma integra tecnologías como Ajax, Java Server Faces (JSF), Java Persistence (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.

Seam ha sido diseñado desde cero para eliminar la complejidad, tanto en la arquitectura como los niveles 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. (9)

Drools

Es un Sistema de Gestión de Reglas de Negocio (BRMS) con una inferencia de encadenamiento hacia adelante, basado en un motor de reglas estándar dentro de la plataforma Java, más bien conocido como un sistema de reglas de producción, mediante una mejor aplicación del algoritmo Rete. Para definir las reglas emplea XML y permite adaptarse a la semántica de un determinado dominio definiendo un esquema que la represente. Drools es una implementación del Java Rule Engine API (JSR-94), estándar para su motor de reglas de negocio y framework para la construcción, mantenimiento, y la ejecución de las políticas de negocio en una organización, una aplicación o servicio. Es además un modelo eficaz de algoritmo para la aplicación de sistemas de reglas de producción. (10)

Java Business Process Manager (JBPM)

Para la realización de un sistema es fundamental la utilización de algunas de las herramientas existentes de administración de procesos de negocio, una de estas es JBPM. Este no es más que un sistema de gestión de flujo de trabajo que posee como responsabilidad principal, mantener el estado de las ejecuciones de procesos. Este modelo de estado, se basa en un gráfico con nodos y transiciones, donde los nodos y las transiciones son los principales ingredientes de una definición del proceso.

Esta tecnología cuenta con un lenguaje para graficar y administrar flujos de procesos partiendo de tareas, acciones automatizadas, estados de espera para comunicación asíncrona y una amplia lista de facilidades que permiten obtener como resultado final un sistema de gran flexibilidad.

Para obtener un sistema lo suficientemente flexible se procede a la utilización de Jboss Drool como marco de trabajo que se integra con facilidad a Jboss JBPM, el cual pretende cumplir con todas las funcionalidades de un Sistema de Administración de Reglas de Negocio (BRMS).

1.3.2.3. Capa de modelo

Hibernate

Esta es una herramienta de mapeo objeto-relacional para la plataforma Java, que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación mediante archivos declarativos XML; 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.

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.

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.

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 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.

1.3.2.4. Tecnologías horizontales

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.

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 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.

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.

1.3.3 Herramientas

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. (11)

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, (X) HTML, Seam, Smooks, JBoss ESB o JBoss Portal, entre otros.

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.

1.3.4 Sistema de gestión 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. (12)

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. (13) 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. (14)

1.3.5 Herramienta de modelado

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. (15)

1.3.6 Gestión de proyecto

Modelo de Capacidad y Madurez (CMMI)

El Capability Maturity Model Integration (CMMI) es un marco de referencia que las organizaciones pueden emplear para mejorar sus procesos de desarrollo, adquisición, y mantenimiento de productos y servicios. El mismo, es una nueva generación de modelos de madurez, que basado en los principios de calidad total, propone un conjunto de buenas prácticas que enseñan qué hacer para implantar procesos productivos más efectivos.

Este modelo plantea la concepción de 5 posibles niveles de madurez que puede alcanzar una organización dependiendo del grado de sofisticación de sus procesos. A su vez, cada nivel, con excepción del inicial queda caracterizado por un conjunto de áreas de procesos que agrupan prácticas que, al ser ejecutadas colectivamente, permiten cumplir con algún objetivo que es considerado importante para el modelo. Estos pueden ser visualizados de forma continua o de forma escalonada, aunque esta última es la más común. (16)

Los niveles de CMMI son:

1. Inicial.
2. Administración básica de proyectos.
3. Proceso estandarizado.
4. Proceso administrado cuantitativamente.
5. Proceso en mejora continua.

El nivel 2 de CMMI tiene como objetivo conseguir que en los proyectos de la organización haya una buena gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados. De esta manera, se ayuda a que los proyectos se ejecuten y gestionen de acuerdo con

los planes de la organización, a que se conozca el estado de los elementos de trabajo de este y a que estos elementos satisfagan las especificaciones, estándares y objetivos. Dentro de este nivel se incluyen 7 áreas de trabajo las cuales se relacionan estrechamente: Administración de requerimientos (REQM), Planificación de proyectos (PP), Monitoreo y control de proyecto (PMC), Medición y análisis (MA), Aseguramiento de la calidad de productos y procesos (PPQA), Administración de la Configuración (CM) y Administración de acuerdos con proveedores (SAM).

La Universidad de las Ciencias Informáticas (UCI), en busca de mejorar la eficiencia y calidad de los procesos de desarrollo de software y la gestión de requerimientos, ha adoptado como estrategia la aplicación en la organización del nivel anteriormente especificado. Trabajando dentro del mismo, en el área de procesos de Administración de Requerimientos (REQM).

El área de proceso de Administración de requerimientos (REQM) tiene como propósito mantener bajo control los requerimientos que el producto a desarrollar deberá satisfacer. Las prácticas incluidas aquí apuntan a que los requerimientos no sólo estén claramente identificados, sino también que todos los involucrados en el proyecto estén de acuerdo en su significado. Un tema fundamental planteado en esta área de proceso es que cualquier cambio realizado a los requerimientos se efectúe de manera controlada y que el resto de los artefactos del proyecto se mantengan consistentes.

Un aspecto importante en esta área es la trazabilidad bidireccional; pues cuando los requerimientos son correctamente administrados se debería estar en condiciones de relacionar cuál ha sido el origen de los requerimientos, cuál es la relación entre los requerimientos de bajo nivel y los de alto nivel, cuáles son los artefactos relacionados con los requerimientos, y cuáles componentes del producto implementan cada requerimiento. Esta práctica es sumamente importante para poder realizar un buen análisis de impacto ante posibles cambios, y fundamental para poder determinar si el alcance del proyecto ha sido cubierto o no. (17)

El modelo CMMI es una alternativa para definir qué hacer durante el desarrollo de un proyecto, sin embargo, este no propone cómo realizarlo. Es de este modo, que CMMI se apoya en metodologías que suministren esta información, teniendo en cuenta que la UCI hasta el momento ha interactuado con el Proceso Unificado de Desarrollo (RUP), se ha mantenido esta metodología de desarrollo de software. De esta manera, el trabajo en conjunto de CMMI como plan de mejora, y RUP como metodología de apoyo de este, guían a la organización para la realización del producto. RUP por su parte utiliza UML como lenguaje para modelar todos los artefactos de los flujos de trabajo y en algunos casos BPMN para modelar los procesos del negocio.

Proceso Unificado de Desarrollo (RUP)

Es una infraestructura 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. (18)

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. (19)

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

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). (20)

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. (21)

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. (22)

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.

1.3.7 Patrón de arquitectura

Modelo – Vista – Controlador (MVC)

La arquitectura 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 (HTTP, consola de comandos, email, etc.). 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. (23)

1.3.8 Entornos distribuidos

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.

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.

El estudio realizado en el presente capítulo generó una visión de los conceptos fundamentales que rodean a los procesos de facturación hospitalaria. A partir de este análisis, se inició la búsqueda de distintas aplicaciones que automatizan estos procesos, sin embargo, ninguno de los sistemas estudiados arrojó resultados satisfactorios debido a sus características específicas. Por lo que se impuso la necesidad de estudiar algunas tecnologías existentes que propicien el desarrollo de la solución a la problemática planteada. Por lo que se decidió utilizar Java, junto al gestor de base de datos PostgreSQL, Visual Paradigm como herramienta de modelado y RUP como metodología de desarrollo de software. Estos aspectos proveen la base de conocimientos necesaria para llevar a cabo la elaboración del sistema.

Capítulo 2: Características del Sistema

El presente capítulo posee como objetivo principal realizar una descripción del funcionamiento del negocio vigente y por consiguiente, mostrará la propuesta de automatización dado el análisis efectuado. Dentro del mismo, será detallado el flujo actual de los procesos del negocio, definiendo los individuos involucrados en este; así como las reglas de negocio existentes. Además, serán definidos los requerimientos del sistema que darán paso al modelado de los casos de uso de este último, planteándose de esta manera la posible solución a la problemática expuesta.

2.1. Objeto de estudio

2.1.1. Objetivos estratégicos de la organización

El objetivo básico de la facturación en un hospital es identificar acertadamente las prestaciones brindadas al paciente durante la atención y su correspondencia con las tarifas indicadas de manera oficial. Así como gestionar las contrataciones y cobros llevados a cabo en cada una de las unidades de la institución. Por lo que, constituye un objetivo estratégico, alcanzar la disponibilidad y accesibilidad a los datos esenciales generados como consecuencia de las necesidades del paciente y las soluciones a estas.

Objetivos estratégicos generales.

- Contribuir al tratamiento seguro y efectivo de pacientes, al contar con un sistema que satisfaga la demanda continua y creciente, que permita llevar a cabo la gestión de la información disponible, ofrecer una respuesta rápida sobre el terreno y que esté basado en el uso de tecnologías de la información innovadoras.
- Identificar la necesidad de realizar la facturación en una institución hospitalaria, lo cual permite el control y la predicción del capital financiero en la misma.
- Formular, documentar y llevar a cabo un registro de todas las facturas realizadas en el hospital.
- Mejorar los métodos de recolección de la información y de cálculo de costos para hacer más factible los procesos de facturación.

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

El sector de facturación en una institución hospitalaria es el responsable de gestionar toda la información referente a los servicios y medios consumidos por un paciente, a partir de la llegada del mismo al hospital. Dentro de esta área existen especialistas encargados de recolectar y registrar toda la información de las prestaciones ofrecidas por la institución para la posterior facturación. Es por ello que, para obtener una factura final se realizan distintos procesos:

Registrar prestaciones ambulatorias: El proceso de registrar las prestaciones ambulatorias es realizado por un especialista en facturación. Este dirige y procesa toda la información de los servicios proporcionados al paciente durante la estancia del mismo en la institución, siempre y cuando esta se realice de forma ambulatoria. El paciente puede recibir servicios de emergencia, donde el especialista registra todas las prestaciones, las cuales van desde la realización de exámenes, cirugías, transfusiones, hasta la posible realización de una autopsia al paciente.

Ambulatoriamente el paciente puede además solicitar una consulta médica; de la cual debe ser registrada la tipología, durabilidad y los datos específicos. Existe también la posibilidad de que el paciente sólo se acerque a la institución en busca de servicios, aunque estos no hayan sido orientados dentro de la misma. Es de este modo, que pueden ser registrados exámenes realizados al paciente de modo externo a la institución como pruebas, cirugías o una autopsia en caso de una defunción fuera del hospital.

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.

Registrar prestaciones de internados: Este proceso es dirigido por el especialista encargado de la facturación, el mismo recibe y procesa toda la información de las prestaciones ofrecidas a un paciente, a partir del ingreso de este al hospital hasta su posterior egreso. Durante la estancia, al paciente se le ofrecen servicios de cirugía, exámenes de imagenología o laboratorio, transfusiones, medicamentos y estudios generales; los cuales son registrados por el especialista en la hoja de facturación, junto a los detalles del alojamiento.

Todos estos servicios pueden ser registrados de modo independiente, o sea, por lista de precios, o en forma de módulos agrupando varias prestaciones en una sola. Luego de este registro, toda la información es valuada utilizando un listado de precios de las prestaciones, finalizando las actividades con la emisión de la factura a la caja del hospital.

Capítulo 2: Características del Sistema

Realizar facturación: El proceso de administración de caja es dirigido por el encargado de caja, quien obtiene la hoja de factura. Esta hoja es revisada para su posterior aceptación o anulación; en el primero de los casos se procede a la creación de la factura y en el segundo se devuelve esta para su rectificación o desecho. La factura puede ser ordinaria (factura simple), rectificativa (factura ya realizada que tuvo una previa rectificación) o recapitulativa (factura que contiene un conjunto de facturas). El encargado tiene potestad para efectuar la creación o modificación de la misma, hasta realizar la emisión final.

Generar reportes: El encargado de caja es además quien debe emitir todos los reportes referentes a las prestaciones y facturas realizadas en la institución hospitalaria; con el objetivo de obtener una base de conocimientos para la planificación y predicción financiera. De modo que se emiten reportes de:

- Precio por prestaciones.
- Consumo por paciente.
- Facturas por paciente.
- Facturas.
- Prestaciones por área.

2.1.3. Causas que originaron la situación problemática

La gestión financiera es una de las bases fundamentales en cualquier organización de la sociedad, debido a que la contabilización de todos los gastos e ingresos producidos, son vitales para el éxito. Las instituciones hospitalarias no están alejadas de esta situación, tanto las públicas como las privadas, sin importar si brindan servicios gratuitos o no, administran el capital financiero que se transfiere en ellas.

Hasta hace algunas décadas esta administración se podía realizar con efectividad de forma manual y utilizando métodos tradicionales, pues los hospitales ofrecían una menor cantidad de servicios y la población que se atendía en ellos era más escasa. Sin embargo, en la actualidad una institución hospitalaria ofrece disímiles servicios que divergen en complejidad y costo. Además, la cantidad de pacientes atendidos en las instituciones es mayor cada día y estos exigen un aumento en la garantía de la calidad hospitalaria.

Capítulo 2: Características del Sistema

Todos estos elementos provocan que las cuentas dentro de la institución hospitalaria sean más complejas y que la cantidad de estas que se manejan en un día sea mucho más grande. Además, actualmente las instituciones requieren una gestión de la facturación con mayor precisión, así como una mayor velocidad en los resultados para obtener el conocimiento y el control indispensable para una buena realización del trabajo. Por ello, hacer el trabajo manual, almacenando la información en formato duro, se ha convertido en un problema y una forma poco factible de facturar.

2.1.4. Propuesta del sistema

Para mejorar el manejo de la información financiera dentro de un hospital, se propone diseñar un sistema automatizado que gestione los procesos de facturación dentro de las instituciones hospitalarias; con el fin de garantizar mayor precisión y eficiencia en los mismos. 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.

2.2. Información que se maneja

Factura: Es el elemento legal que se emite luego de finalizada la atención al paciente. La misma contiene: número de factura, fecha de expedición, nombre del paciente, dirección domiciliaria, carné de identidad, datos del emisor, domicilio emisor, prestaciones del paciente, precio de las prestaciones, tipo impositivo (IVA), fecha de prestación del servicio, estado.

Hoja de Factura: Registro que contiene la información personal del paciente: nombre, apellidos, carné de identidad, dirección domiciliaria. Recoge además, la información de las distintas prestaciones y servicios ofrecidos al paciente durante su estancia en el hospital.

Listado de Precios: Documento que contiene todas las prestaciones que se ofrecen en el hospital y el precio de cada una de ellas para la realización de la facturación al paciente.

Registro de Egreso: El registro contiene la información referente a los egresos que se han realizado en la institución como son tipo de egreso, motivos, fecha en que se realizó y causas del mismo. A partir de este se puede determinar la fecha y causa de egreso de un paciente.

Registro de Pacientes de Admisión: Registra la información de los pacientes ingresados, la fecha en que ocurre el ingreso, motivos, servicio que atiende al paciente, diagnóstico y ubicación del paciente. Permite obtener los datos del paciente a ingresar y la información del alojamiento.

Capítulo 2: Características del Sistema

Reporte de Cirugía: Contiene la información de las cirugías realizadas, lo que permite conocer si las mismas han sido o no de emergencias, y si el resultado de las mismas ha sido el esperado. Esta información es importante porque permite conocer si al paciente se le realizó alguna cirugía para registrar los datos de la misma.

Reporte de Transfusiones: Recoge toda la información referente a una solicitud de transfusión. Contiene una sección de datos del paciente con el nombre y apellidos de este, número de historia clínica, edad, diagnóstico, hemoglobina, tipo y factor de la sangre de este. Esta información permite determinar si al paciente se le realizó una transfusión y obtener los datos de la misma.

Hoja de Emergencia: Recoge la información de los pacientes que se atendieron por emergencia. La misma contiene en caso de que se conozca, los datos personales del paciente, la fecha, motivo de la consulta, ubicación y nivel de gravedad. En caso de que el paciente sea desconocido se administra una descripción personal, sexo, datos personales del acompañante, gravedad y motivo de la consulta. Este objeto permite obtener los datos del paciente en emergencia.

Indicaciones del Médico: Estas indicaciones contienen los datos generales del paciente, los datos generales del médico que lo atendió, la conducta tomada, el interrogatorio al paciente, los medicamentos indicados, y el diagnóstico final. Este documento es necesario para obtener la información de los servicios ofrecidos al paciente.

Reporte de Exámenes de Anatomía Patológica: Este documento recoge toda la información referente a los exámenes que se realizan en el área de anatomía patológica del hospital. El mismo incluye los campos: número de solicitud, tipo de examen, tipo de muestra, datos personales del paciente y fecha de realización. El contenido de este informe permite obtener los posibles exámenes patológicos que se haya realizado el paciente, así como las especificidades de estos.

Stock de Enfermería: En este documento se registran los datos de la indicación médica, con el nombre del médico, datos del paciente, el servicio al que pertenece, fecha y hora del cumplimiento del tratamiento. De los medicamentos indicados, se guardará el nombre, tipo, dosis, vía, volumen en caso de ser líquido. Además, se recoge por turno, el nombre del enfermero que le da cumplimiento a la indicación, con la fecha, hora y observaciones.

Reporte de Defunción: Este documento contiene un compendio informativo del fallecimiento de un paciente, y la realización de una autopsia si esta se solicita. Contiene los datos: nombre del

Capítulo 2: Características del Sistema

fallecido, fecha y hora de muerte, número de autopsia, médico de guardia, persona que autoriza, parentesco, servicio remitente, establecimiento del remitente, nombre del establecimiento. Con el mismo se pueden obtener la confirmación de una realización de autopsia y los detalles de esta.

2.3. Modelado del negocio

El flujo de trabajo Modelado del negocio tiene como objetivo llegar a un mejor entendimiento de la organización donde se va a implantar el producto. Esto es necesario para asegurar que el producto sea algo útil, que encaje del mejor modo posible en la organización; y que exista un marco común para los desarrolladores, los clientes y los usuarios finales.

2.3.1. Definición de roles

Un trabajador del negocio define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo.

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 realización del modelo de negocio a través de procesos con la notación BPMN es muy efectiva, debido a que esta notación aporta una mayor visibilidad de las actividades que se realizan y ayuda a lograr un mejor entendimiento del flujo de trabajo existente entre las áreas hospitalarias.

Mediante los elementos gráficos de la notación BPMN, se pueden construir los diagramas de procesos. Así, existen eventos de inicio y fin, que indican una acción lógica en el flujo del proceso; y subprocessos y dentro de ellos. Hay actividades atómicas que se relacionan por un flujo de secuencia. Las bifurcaciones determinan la ramificación del flujo a través de decisiones. Y los artefactos brindan la información de los documentos o registros que se requieren y se producen en cada actividad.

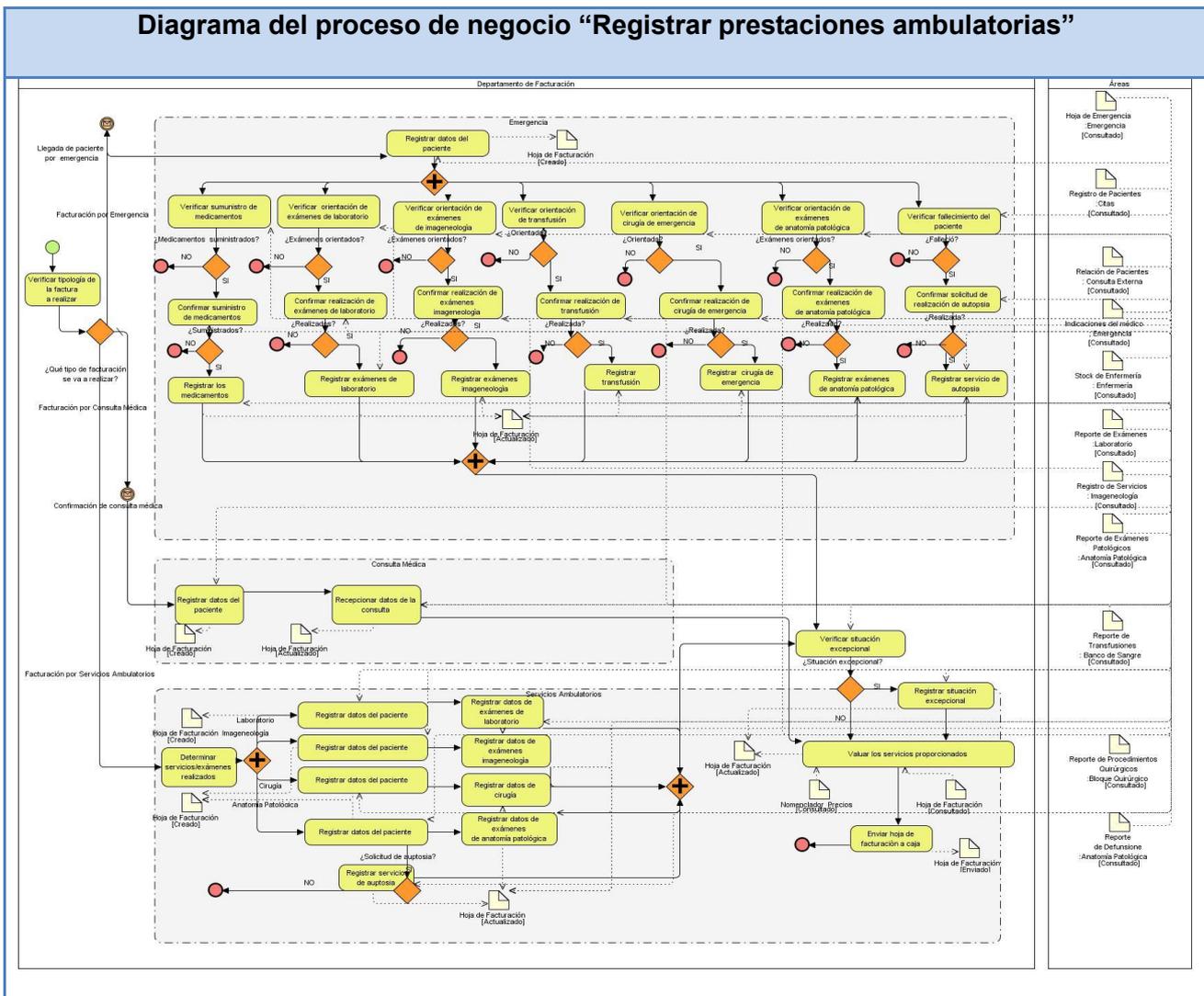


Ilustración 1: Diagrama del proceso del negocio "Registrar prestaciones ambulatorias".

2.4. Especificación de los requisitos del sistema

Los requerimientos de software, posibilitan determinar la solución tecnológica a algún problema existente, logrando como resultado, un informe detallado de los requerimientos necesarios para la solución a dicho problema. Estos se clasifican en funcionales y no funcionales, donde la primera clasificación corresponde a las especificaciones del sistema, mientras que los últimos expresan las cualidades que el producto debe tener.

2.4.1. Requisitos funcionales del sistema

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir y que definen el comportamiento interno del software. Dentro de las acciones identificadas como requisitos funcionales para el desarrollo del sistema se pueden tener en cuenta las que se listan a continuación:

- RF1 Seleccionar la historia clínica de pacientes pendientes a facturar.
- RF2 Crear una nueva factura al paciente seleccionado.
- RF3 Ver detalles de la factura creada.
- RF4 Modificar la factura creada.
- RF5 Eliminar la factura creada.
- RF6 Buscar factura.
- RF7 Ver datos de una factura previamente creada.
- RF8 Buscar datos de paciente.
- RF9 Ver datos de paciente.
- RF10 Establecer el estado de emisión de una factura.
- RF11 Buscar las prestaciones ofrecidas al paciente por cada área.
- RF12 Buscar otros consumos ofrecidos al paciente.
- RF13 Registrar las prestaciones ofrecidas al paciente por cada área.
- RF14 Buscar el precio de las prestaciones ofrecidas al paciente.

- RF15 Registrar otros consumos ofrecidos al paciente.
- RF16 Valuar las prestaciones y los consumos brindados al paciente.
- RF17 Exportar factura.
- RF18 Establecer las prestaciones a facturar por cada área en la institución.
- RF19 Registrar el precio de las prestaciones de la institución.
- RF20 Modificar el precio de las prestaciones de la institución.
- RF21 Seleccionar las áreas de la institución.
- RF22 Crear moneda de trabajo.
- RF23 Ver detalles de una moneda de trabajo.
- RF24 Buscar monedas de trabajo existentes.
- RF25 Establecer moneda de trabajo.
- RF26 Modificar moneda de trabajo.
- RF27 Eliminar moneda de trabajo.
- RF28 Generar reporte de facturas.
- RF29 Generar reporte de precio por prestaciones de la institución.
- RF30 Generar reporte de prestaciones por área.
- RF31 Generar reporte de consumo por paciente.
- RF32 Generar reporte de facturas por paciente.
- RF33 Exportar los reportes generados.

2.4.2. Requisitos no funcionales del sistema

Los requisitos no funcionales son propiedades o cualidades que debe cumplir un sistema para satisfacer un contrato o una especificación. Están regidos por las necesidades del usuario para poder resolver un problema o conseguir un beneficio determinado. Existen múltiples categorías de requisitos

no funcionales que un producto puede cumplir, y que permiten marcar la diferencia ante otros, por la aceptación del usuario.

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.
- 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 salvadas 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.

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.

- 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 desplegados que faciliten y aceleren su utilización. La entrada incorrecta de datos será detectada e informada al usuario.

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.

2.5. Definición de los casos de uso del sistema

El modelo de casos de uso del sistema es uno de los artefactos más importantes generados en la fase de Inicio de RUP, el mismo representa, bajo la forma de acciones y reacciones, el

Capítulo 2: Características del Sistema

comportamiento del sistema desde el punto de vista del usuario, permitiendo de esta forma el establecimiento de un acuerdo entre clientes y desarrolladores sobre las condiciones y requerimientos que debe cumplir el sistema.

2.5.1. Definición de actores

Un actor es cualquier entidad externa al sistema que demanda una funcionalidad de este; puede ser también una abstracción de un software que interactúa de alguna manera con el sistema.

Actores del Sistema	
Actor del sistema	Funcionalidad
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 3: Actores del sistema.

2.5.2. Diagrama de casos de uso

Luego de haber sido definidos, los requisitos del sistema son expresados como casos de uso, teniendo en cuenta que estos son actividades atómicas que ofrecen un resultado definido para cada acción que realiza el usuario. El modelo de casos de uso del sistema documenta el comportamiento del sistema desde el punto de vista del usuario, permitiendo representar las funciones que se desean en el sistema, el entorno del sistema, y las relaciones entre ellos.

2.5.2.1. Diagramas de casos de uso del sistema

Proceso: Realizar facturación

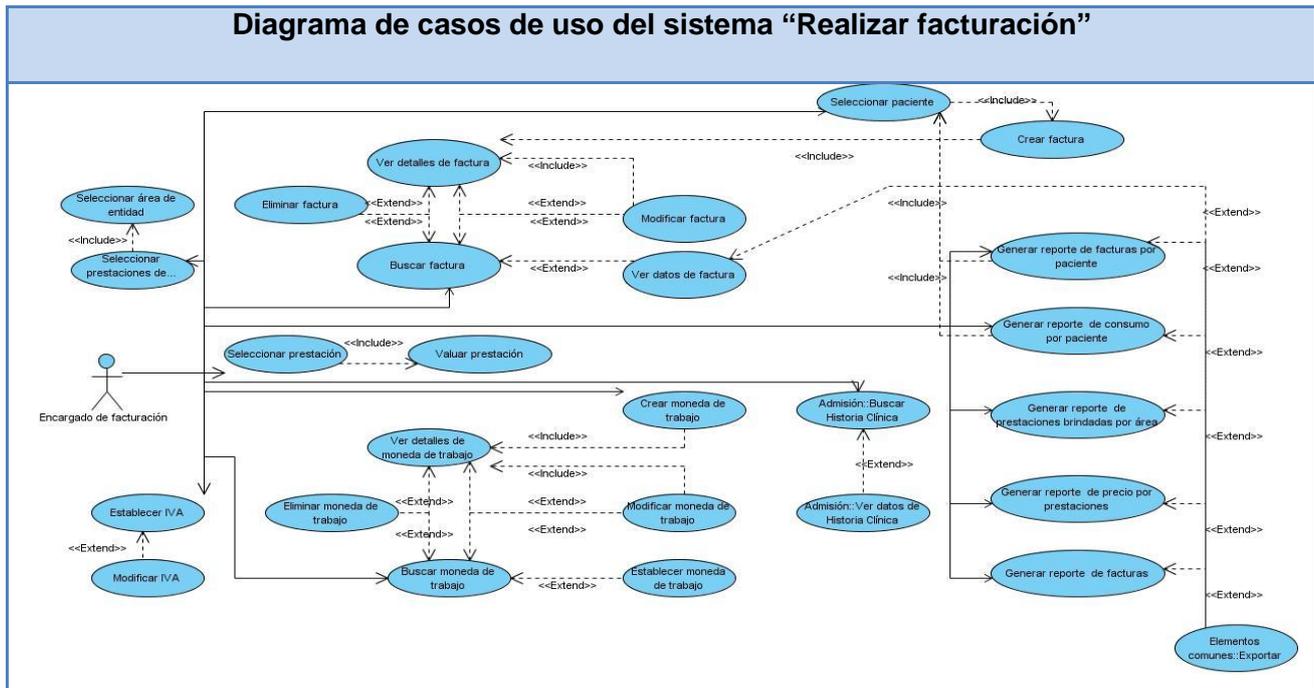


Ilustración 4: Diagrama de casos de uso del sistema “Realizar facturación”.

2.5.2.2. Descripción textual de casos de uso significativos del sistema

CU Crear factura	
Objetivo	Crear una factura hospitalaria al paciente.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor accede a la opción Crear factura, el sistema permite seleccionar el paciente al cual se le crea la factura y obtiene los datos personales del mismo, el sistema crea la factura hospitalaria, el caso de uso termina.
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	La atención hospitalaria al paciente ha sido registrada.
Postcondiciones	Se creó la factura.
R.F. Asociados	RF2, RF11, RF12

Tabla 4: Caso de uso “Crear factura”.

CU Buscar factura

Capítulo 2: Características del Sistema

Objetivo	Buscar una factura de un paciente.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor accede a la opción Buscar factura, el sistema brinda la posibilidad de introducir los criterios de búsqueda para localizar la factura, el actor introduce los datos que considera como criterios para realizar la búsqueda, el sistema busca y muestra las facturas que cumplan con los criterios, el caso de uso termina.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	La factura ha sido creada con anterioridad.
Postcondiciones	La factura fue encontrada.
R.F. Asociados	RF6

Tabla 5: Caso de uso "Buscar factura".

CU Eliminar factura	
Objetivo	Eliminar la factura deseada.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor selecciona una factura y accede a la opción Eliminar factura, el sistema elimina la factura, el caso de uso termina.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	La factura ha sido creada.
Postcondiciones	La factura fue eliminada.
R.F. Asociados	RF5

Tabla 6: Caso de uso "Eliminar factura".

CU Crear moneda de trabajo	
Objetivo	Crear la moneda de trabajo para facturar.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor selecciona la opción Crear moneda de trabajo, el sistema brinda la posibilidad de introducir los datos para incorporar

Capítulo 2: Características del Sistema

	el nuevo tipo de moneda de trabajo, el actor introduce los datos de la moneda de trabajo, el sistema registra la información, el caso de uso termina.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	No existe.
Postcondiciones	El tipo de moneda fue registrado.
R.F. Asociados	RF22

Tabla 7: Caso de uso "Crear moneda de trabajo".

CU Buscar moneda de trabajo	
Objetivo	Modificar la moneda de trabajo para facturar.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor accede a la opción Buscar moneda, el sistema brinda la posibilidad de introducir los criterios de búsqueda para localizar la moneda, el actor introduce los datos que considera como criterios para realizar la búsqueda, el sistema busca y muestra las monedas que cumplan con los criterios, el caso de uso termina.
Complejidad	Media.
Prioridad	Secundario.
Precondiciones	El tipo de moneda ha sido insertado.
Postcondiciones	El tipo de moneda fue encontrado.
R.F. Asociados	RF24, RF23

Tabla 8: Caso de uso "Buscar moneda de trabajo".

CU Establecer moneda de trabajo	
Objetivo	Establecer la moneda de trabajo para facturar.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor accede a la opción Buscar moneda, el sistema muestra las monedas existentes, el actor selecciona la moneda deseada, el caso de uso termina.
Complejidad	Baja.
Prioridad	Crítico.

Capítulo 2: Características del Sistema

Precondiciones	Los tipos de monedas han sido insertados.
Postcondiciones	El tipo de moneda fue establecido.
R.F. Asociados	RF25

Tabla 9: Caso de uso “Establecer moneda de trabajo”.

CU Valuar prestaciones	
Objetivo	Valuar prestaciones del paciente.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor oprime la opción Valuar prestaciones, el sistema muestra las prestaciones existentes, el actor selecciona la prestación deseada, el sistema muestra la opción para introducir el precio, el actor introduce el precio deseado y acepta, el caso de uso termina.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	Las prestaciones de la institución han sido seleccionadas.
Postcondiciones	El precio de las prestaciones ha sido definido.
R.F. Asociados	RF16

Tabla 10: Caso de uso “Valuar prestaciones”.

CU Generar reporte de prestaciones	
Objetivo	Generar un reporte de las prestaciones.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor accede a la opción Generar reporte de las prestaciones, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes genera el reporte de información. El actor selecciona la opción de exportar el reporte, el sistema exporta los datos del reporte, el caso de uso termina.
Complejidad	Media.
Prioridad	Secundario.

Capítulo 2: Características del Sistema

Precondiciones	No existe.
Postcondiciones	El reporte fue generado.
R.F. Asociados	RF30

Tabla 11: Caso de uso "Generar reporte de prestaciones".

CU Generar reporte de facturas	
Objetivo	Generar un reporte de facturas.
Actores	Encargado de facturación.
Resumen	El caso de uso inicia cuando el actor accede a la opción Generar reporte de facturas, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes genera el reporte de información. El actor selecciona la opción de exportar el reporte, el sistema exporta los datos del reporte, el caso de uso termina.
Complejidad	Media.
Prioridad	Secundario.
Precondiciones	No existe.
Postcondiciones	El reporte fue generado.
R.F. Asociados	RF28

Tabla 12: Caso de uso "Generar reporte de facturas".

El desarrollo de este capítulo permitió abordar a fondo los procesos de facturación, demostrándose la complejidad de los mismos y la necesidad de realizar una propuesta de sistema, que minimice las dificultades que estos implican para los especialistas. Además, se logró alcanzar un mejor entendimiento de cómo sería el sistema y las restricciones que deben existir para satisfacer las especificaciones de los clientes. Se detallaron todos los requerimientos funcionales y no funcionales del sistema, se identificaron los actores que intervienen en el mismo; así como todos los casos de uso, los cuales fueron descritos minuciosamente y reflejan las funcionalidades recogidas en los requerimientos. El desarrollo de estos flujos de trabajo y los artefactos obtenidos, permiten comenzar con el análisis y diseño del sistema propuesto.

Capítulo 3: Análisis y Diseño del Sistema

El presente capítulo posee como objetivo representar una concepción más explícita del sistema que se propone realizar. Describiendo a partir del análisis y diseño, qué va a contener y gestionar el sistema y cómo este va a realizar la gestión a partir de las herramientas propuestas. En el capítulo además, se realizará una propuesta de la base de datos para el sistema, así como también un diseño para la interfaz de usuario del software.

3.1. Descripción de la arquitectura. Fundamentación

El buen funcionamiento de un software está determinado en gran medida por la organización y estructuración de los componentes del mismo y de las relaciones entre ellos. Realizar un sistema con un diseño y un orden establecido es factible para el desarrollo en sí y para alcanzar un grado de calidad. Es en este punto donde la elección de una arquitectura adecuada para el software que se está realizando, se convierte en una tarea sumamente importante.

Existen muchos patrones de arquitectura ya predefinidos y que pueden ser útiles en diferentes sistemas, de acuerdo con sus características. Uno de ellos es el patrón Modelo Vista Controlador (MVC), el cual separa los datos una la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (vista, controlador y modelo). Este patrón tiene como objetivo principal mejorar la reusabilidad y que las modificaciones en la vista impacten en menor medida en la lógica de negocio o de datos.

La propuesta del módulo de Facturación, utiliza este patrón de arquitectura donde la capa de presentación está formada por los formularios contenidos en las páginas XHTML, que mediante controles JSF y RichFaces gestionan los datos con los cuales el usuario realiza determinadas operaciones. La capa de negocio está integrada por las clases controladoras quienes son las responsables de recibir los eventos de entrada desde la vista y definen el estado de los datos y las entidades que manejan. La capa de acceso a datos está constituida por componentes Hibernate, mediante los cuales se realiza el mapeo de tablas de la base de datos a clases Java y provee facilidades de consultas de datos y de recuperación.

La comunicación entre los elementos de estas capas está regida por el framework Seam ya que este integra la capa de presentación (JSF) con la capa de negocios y persistencia. Seam permite además, que mediante anotaciones propias de este, las páginas de interfaz de usuario referencien las

funcionalidades definidas en las clases controladoras, de modo que estas, puedan usar los componentes de acceso a datos y otros de la capa del negocio.

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

Para la automatización de los procesos de facturación en las instituciones hospitalarias podría proponerse un sistema que gestione la información de forma autónoma e independiente. Sin embargo, un sistema de facturación debe obtener datos relevantes de todas las áreas del hospital, las cuales le proveen la información de las prestaciones ofrecidas al paciente. De modo que una conexión entre áreas sería más efectiva.

Es por ello que se propone realizar un módulo de Facturación, el cual será una ramificación del Sistema de Información Hospitalaria alas HIS, software encargado de gestionar toda la información que se maneja en las instituciones hospitalarias. Este no sólo será una parte del HIS, sino que además, utilizará componentes y módulos ya existentes en el mismo, para evitar la repetición de código.

El módulo cumplirá los requerimientos no funcionales de seguridad, a través de funcionalidades ya implementadas en el HIS, el cuál gestiona los niveles de usuarios, privilegios, así como las acciones de estos, en el módulo de Configuración. Para la gestión de prestaciones de la institución serán utilizadas las tablas que contienen información de los servicios de las áreas Bloque Quirúrgico, Anatomía Patológica, Consulta Externa, Farmacia y Laboratorio. Además, serán utilizadas las tablas de tratamientos y componentes sanguíneos de Banco de Sangre, entre otras.

3.3. Modelo de análisis

El modelo de análisis es el primer acercamiento al sistema. En este se obtiene una visión de “qué” va a hacer el software detalladamente, pero sin centrarse en el “cómo”. El objetivo principal de este modelo es traducir los requisitos en una especificación que describe qué va implementar el sistema, por lo que centra su atención en los requerimientos funcionales.

3.3.1. Diagramas de clases de análisis

Diagrama de clases del análisis del CU “Crear factura”

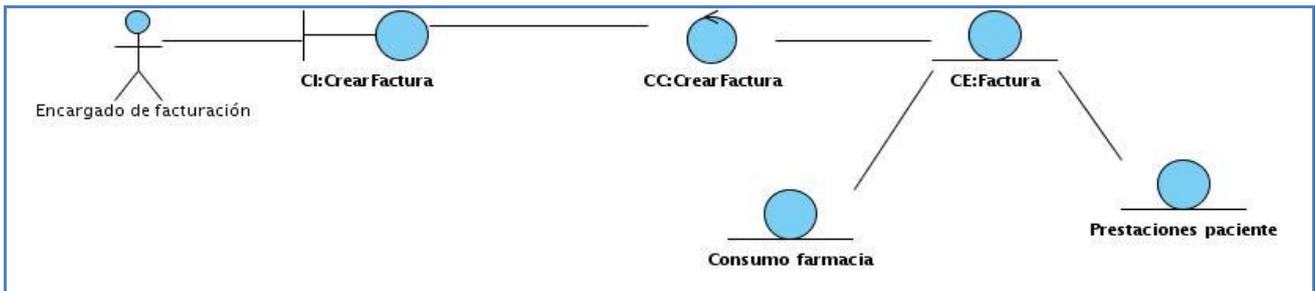


Ilustración 4: Diagrama de clases del análisis del CU “Crear factura”.

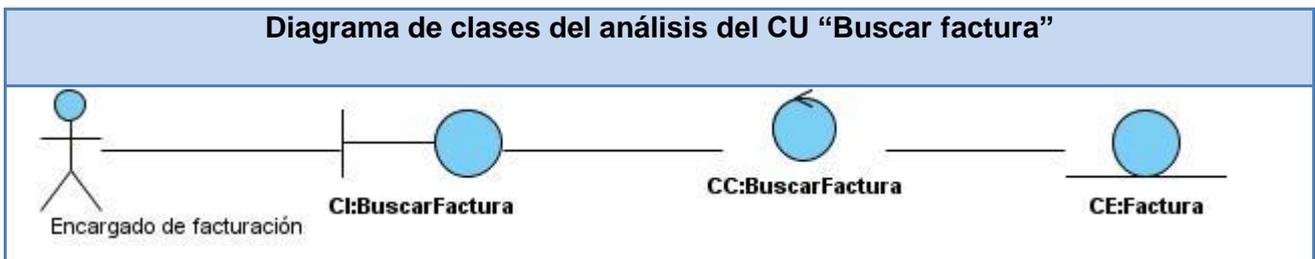


Ilustración 5: Diagrama de clases del análisis del CU “Buscar factura”.

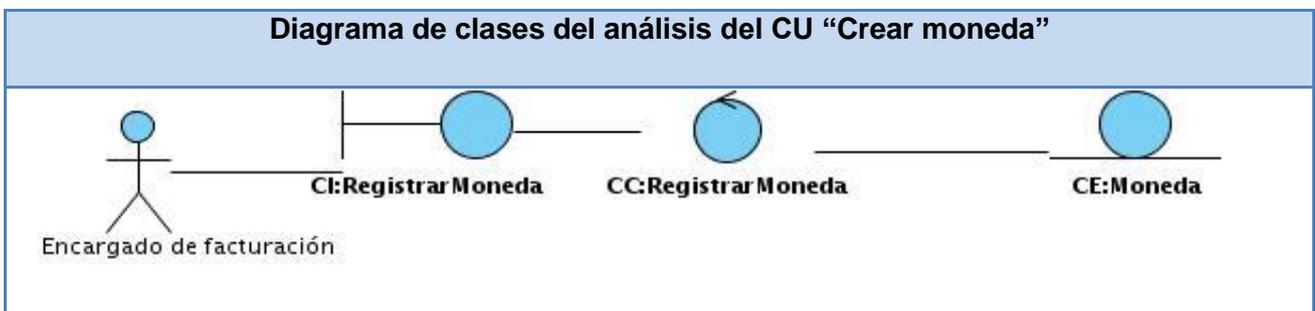


Ilustración 6: Diagrama de clases del análisis del CU “Crear moneda”.

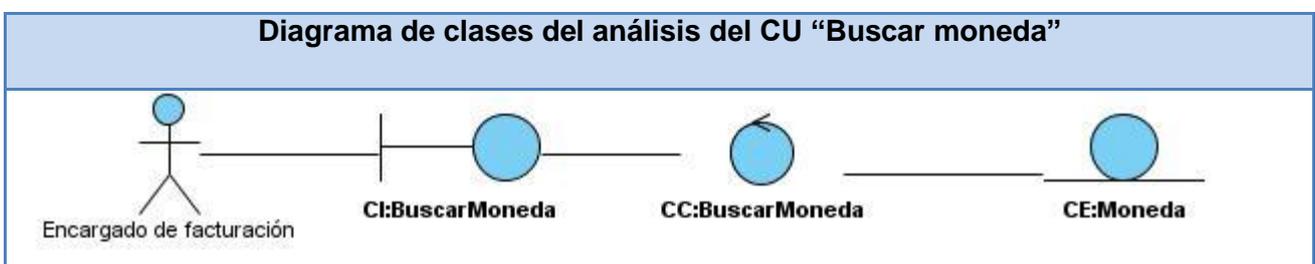


Ilustración 7: Diagrama de clases del análisis del CU “Buscar moneda”.

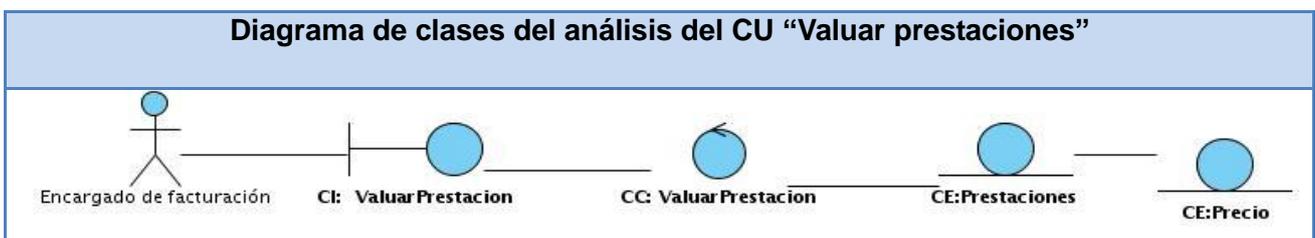


Ilustración 8: DCA Diagrama de clases del análisis del CU “Valuar prestaciones”.

3.4. Modelo de diseño

El modelo del diseño es una representación más profunda del sistema, donde se especifica “cómo” se va a implementar el sistema. El mismo define todas las clases del diseño y debe cumplir con todas las restricciones del lenguaje de programación a utilizar, así como de la arquitectura usada en el sistema. El modelo del diseño centra su atención además, en los requerimientos no funcionales del software.

3.4.1. Fundamentación del uso de patrones

Los patrones del diseño son una descripción de clases y objetos comunicándose entre sí, adaptados para resolver un problema de diseño general en un contexto particular. Estos permiten reutilizar las soluciones de determinados problemas creadas por programadores, de modo que garantizan la socialización del conocimiento y el mejoramiento del software.

Entre los patrones de diseño se encuentran los patrones GRASP, los cuales asignan responsabilidades a las clases representantes del software. En el módulo de Facturación fueron utilizados algunos patrones que se encuentran en este grupo; el patrón Experto y el Creador pues se le asignaron las responsabilidades a las clases que tenían mayor conocimiento de la información a tratar.

Además, el diseño permite que las distintas clases del sistema interactúen entre sí, sin afectar esto la reutilización de las mismas o el correcto funcionamiento de las clases por separado, con el objetivo de cumplir, de esta forma, con los patrones de Bajo Acoplamiento y de Alta Cohesión.

3.4.2. Estructura del diseño

Para la elaboración del modelo de diseño, se define una estructura de paquetes que permite dividir el sistema en fragmentos manejables para su futura implementación.

Los paquetes son graficados mostrando la relación que guardan entre sí. Estos utilizan el paquete repositorio de clases para su funcionamiento. Un paquete referente a procesos, está conformado por subpaquetes que responden a las realizaciones de casos de uso. El paquete repositorio contiene tres subpaquetes, uno para las vistas, otro para las entidades y por último el de las sesiones. El subpaquete de las vistas incluye todas las vistas existentes y utilizadas en el proceso actual.

En el subpaquete de entidades, se encuentran las clases autogeneradas definidas en el diseño de acuerdo con las tecnologías que serán usadas en la implementación. Las clases autogeneradas, como su nombre lo indica, se autogeneran desde la base de datos utilizando el ORM Hibernate.

Las clases personalizadas son aquellas que se modifican, por lo que pueden heredar de las entidades autogeneradas. El subpaquete de sesiones, está conformado por las clases controladoras autogeneradas por el entorno de desarrollo, además, de las clases controladoras personalizadas y las controladoras del proceso.

A continuación se muestra el diagrama de paquetes del sistema:

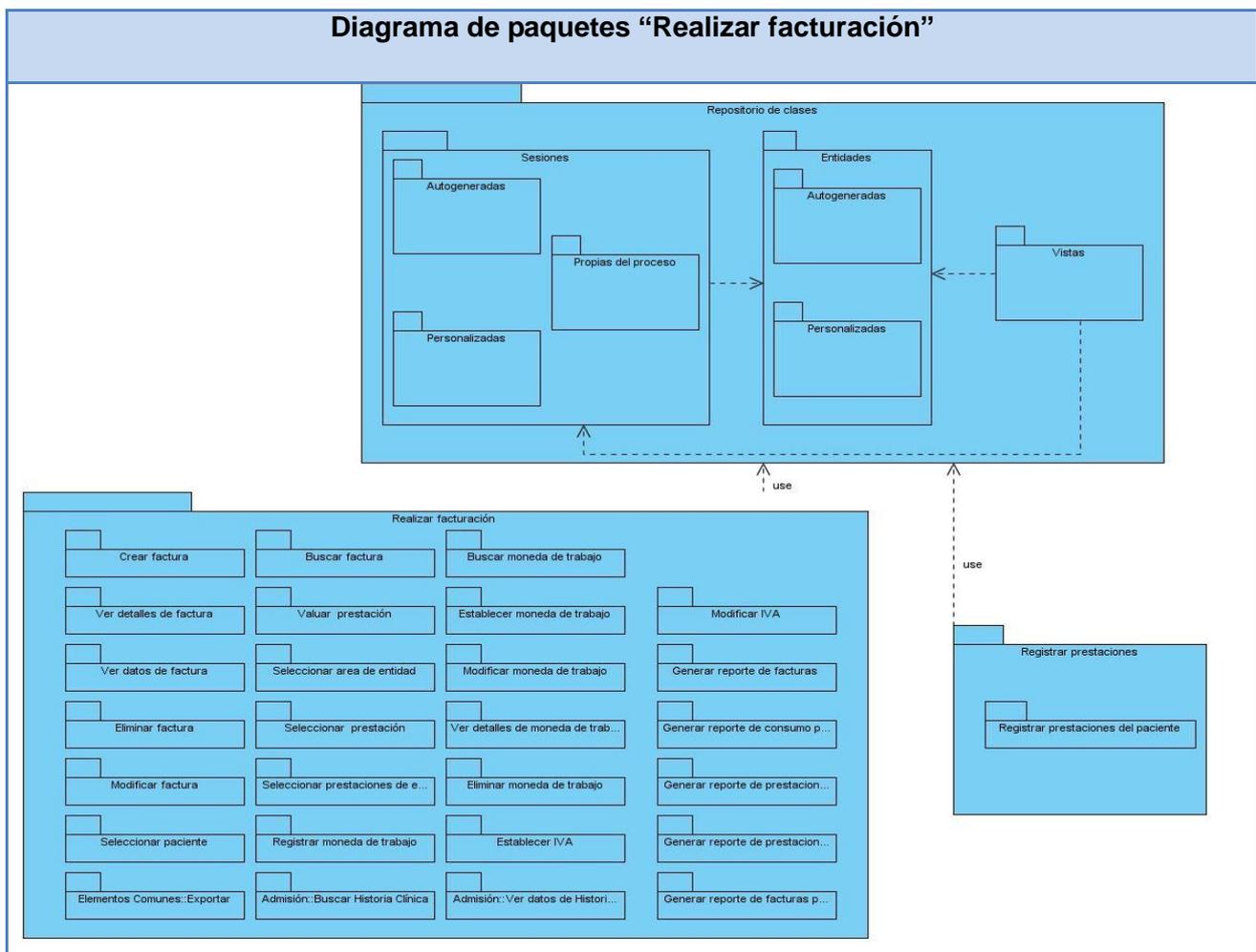


Ilustración 9: Diagrama de paquetes general "Realizar facturación".

3.4.3. Diagramas de clases del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, de los componentes que se encargarán del funcionamiento y la relación entre uno y otro. (Ver Anexo 1)

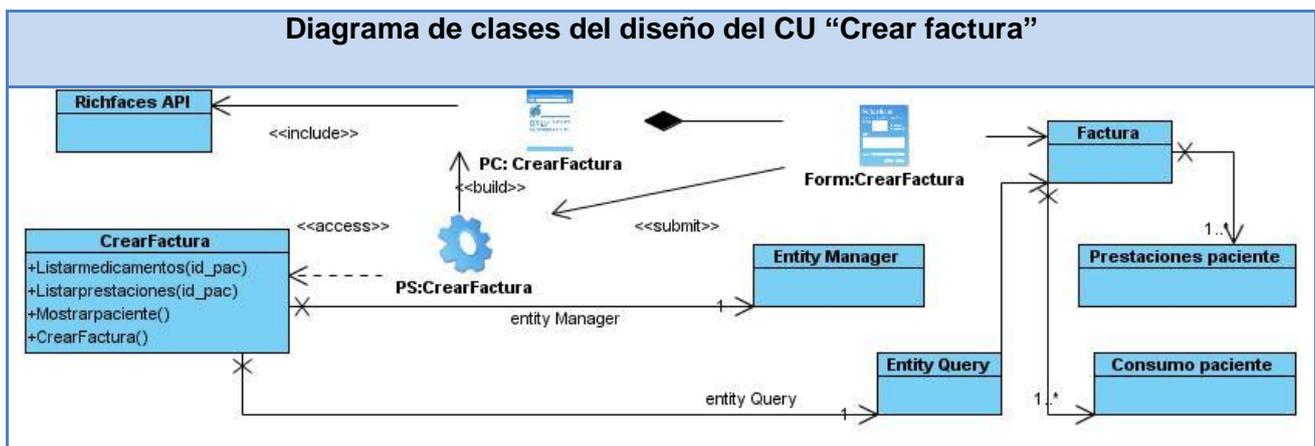


Ilustración 10: Diagrama de clases del diseño del CU "Crear factura".

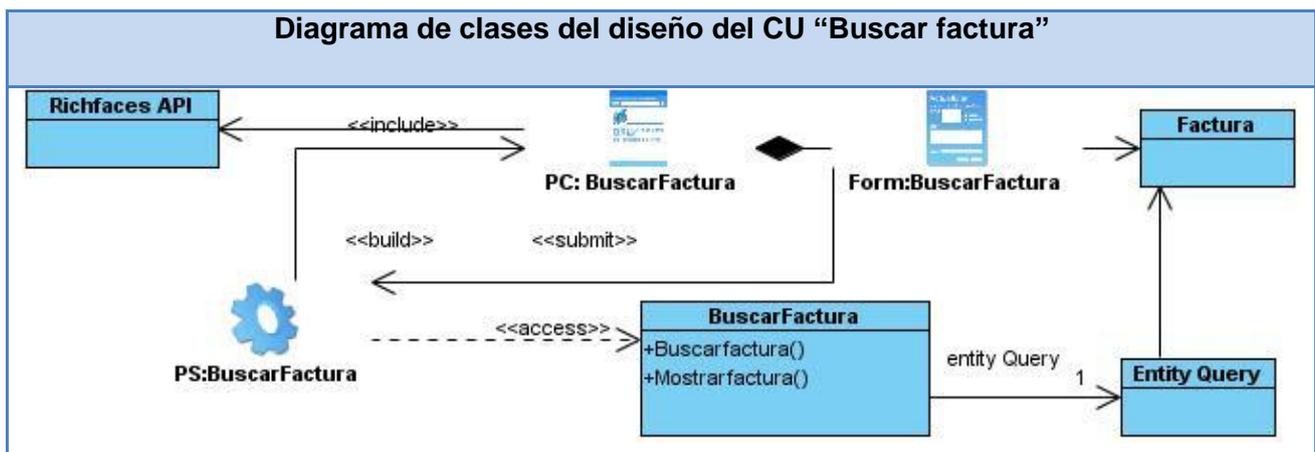


Ilustración 11: Diagrama de clases del diseño del CU "Buscar factura".



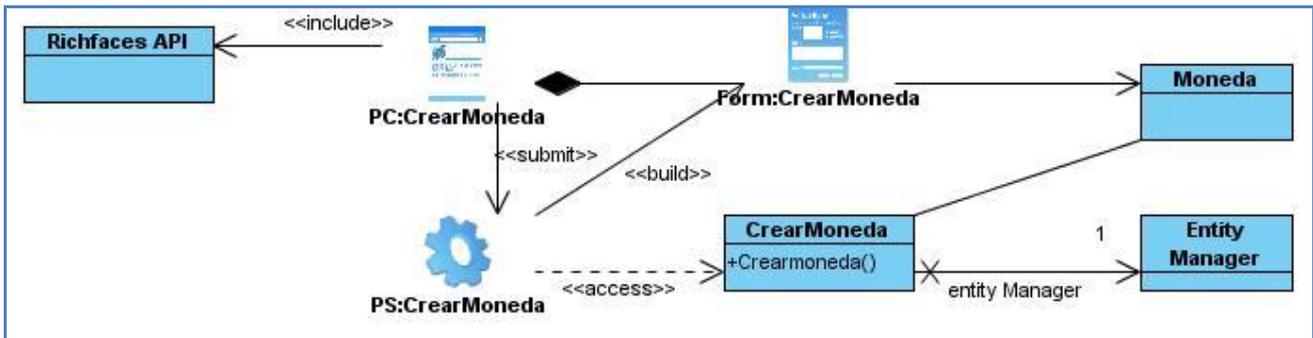


Ilustración 12: Diagrama de clases del diseño del CU “Crear moneda”.

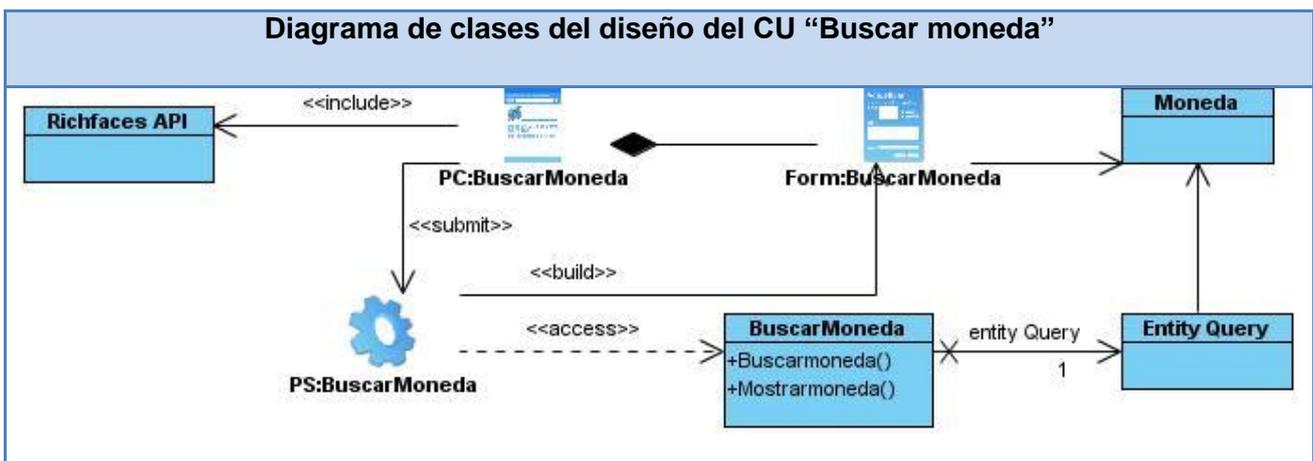


Ilustración 13: Diagrama de clases del diseño del CU “Buscar moneda”.

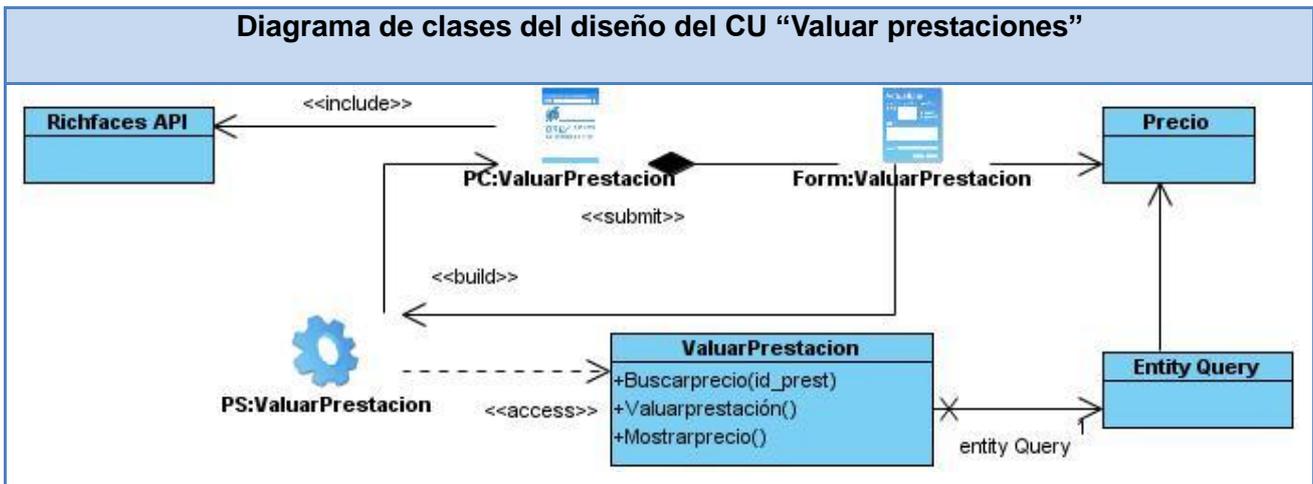


Ilustración 14: Diagrama de clases del diseño del CU “Valuar prestaciones”.

3.4.4. Descripción de las clases

Nombre: CrearFactura
Controladora

Atributo		Tipo
Id_pac		Integer
Id_fact		Integer
Para cada responsabilidad:		
Nombre:	Listarmedicamentos(id_pac)	
Propósito:	Esta operación se encarga de buscar todos los medicamentos que consumió el paciente.	
Nombre:	Listarprestaciones(id_pac)	
Propósito:	Esta operación se encarga de buscar todas las prestaciones ofrecidas al paciente.	
Nombre:	Mostrarpaciente()	
Propósito:	Muestra los datos del paciente y las prestaciones y medicamentos del mismo.	
Nombre:	CrearFactura()	
Propósito:	Crea la factura luego de insertados los datos.	

Tabla 14: Descripción de clase “CrearFactura”.

Nombre: BuscarFactura		
Controladora		
Atributo		Tipo
Id_fact		Integer
Para cada responsabilidad:		
Nombre:	Buscarfactura()	
Propósito:	Esta operación se encarga de buscar todas las facturas que cumplan con los criterios especificados.	
Nombre:	Mostrarfactura()	
Propósito:	Esta operación se encarga de mostrar todas las facturas encontradas en la búsqueda.	

Tabla 15: Descripción de clase “BuscarFactura”.

Nombre: CrearMoneda		
Controladora		
Atributo		Tipo
Id_moneda		Integer
Para cada responsabilidad:		

Nombre:	Crearmoneda()
Propósito:	Esta operación se encarga de adicionar un nuevo tipo de moneda al sistema.

Tabla 16: Descripción de clase “CrearMoneda”.

Nombre: BuscarMoneda	
Controladora	
Atributo	Tipo
Id_moneda	Integer
Para cada responsabilidad:	
Nombre:	Buscarmoneda()
Propósito:	Esta operación se encarga de buscar las monedas que cumplan con los criterios de búsqueda.
Nombre:	Mostrarmoneda()
Propósito:	Esta operación se encarga de mostrar las monedas encontradas en la búsqueda.

Tabla 17: Descripción de clase “BuscarMoneda”.

Nombre: ValuarPrestaciones	
Controladora	
Atributo	Tipo
Id_prest	Integer
Para cada responsabilidad:	
Nombre:	Buscarprecio(id_prest)
Propósito:	Esta operación se encarga de buscar el precio de la prestación.
Nombre:	Valuarprestacion()
Propósito:	Esta operación se encarga de introducir el precio de la prestación.
Nombre:	Mostrarrprecio()
Propósito:	Esta operación se encarga de mostrar el precio de la prestación.

Tabla 18: Descripción de clase “ValuarPrestaciones”.

3.4.5. Diagramas de interacción

Se trata de un término genérico que se aplica a varios tipos de diagramas que hacen hincapié en las interacciones entre objetos. En este caso, se le denomina así a los diagramas de secuencia y a los

diagramas de colaboración. Estos son diagramas isomorfos y generalmente se utiliza el diagrama de secuencia para realizar el diseño del sistema. (Ver Anexo 2)

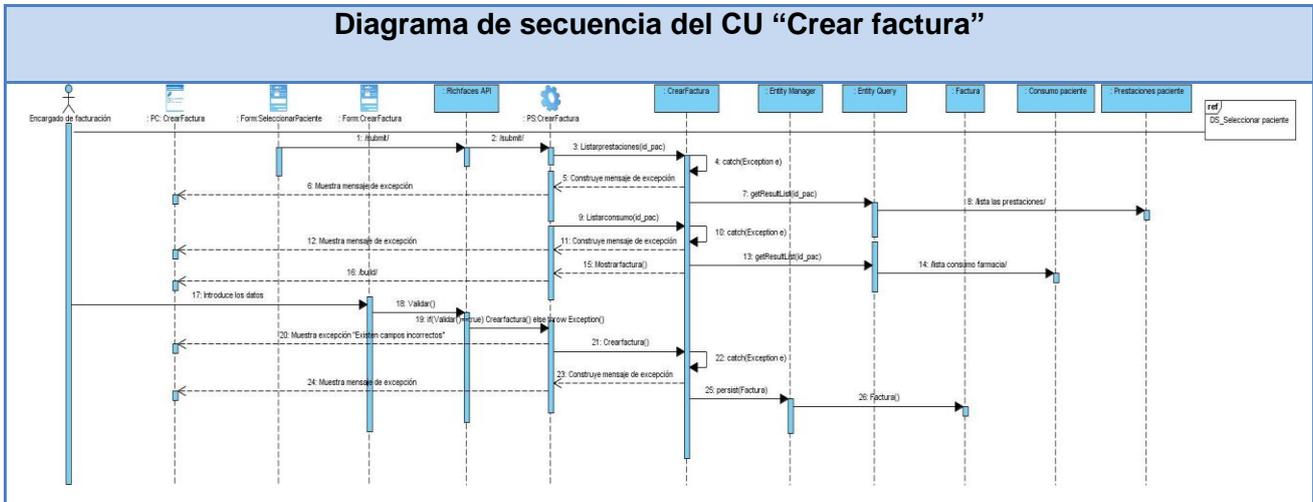


Ilustración 15: Diagrama de secuencia del CU "Crear factura".

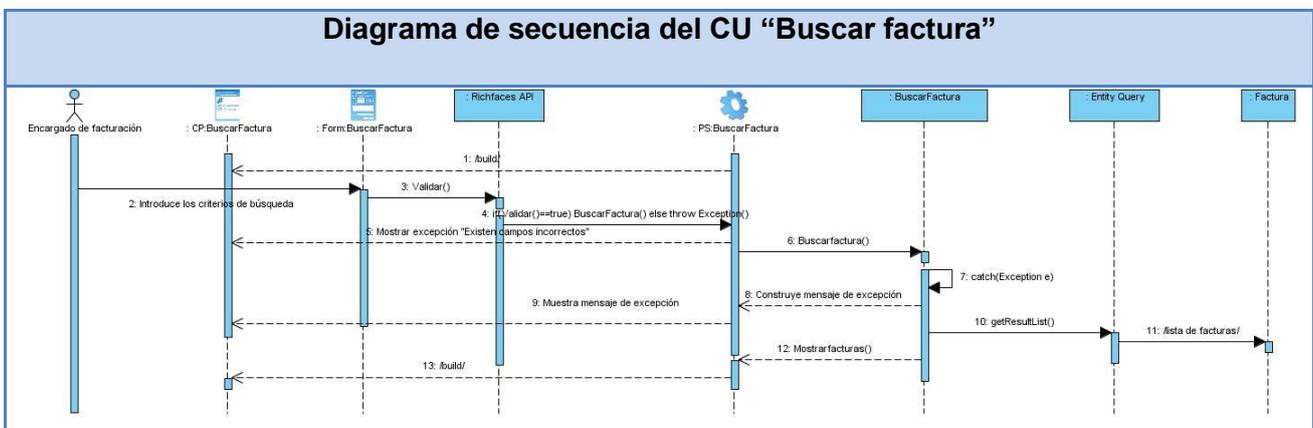


Ilustración 16: Diagrama de secuencia del CU "Buscar factura".



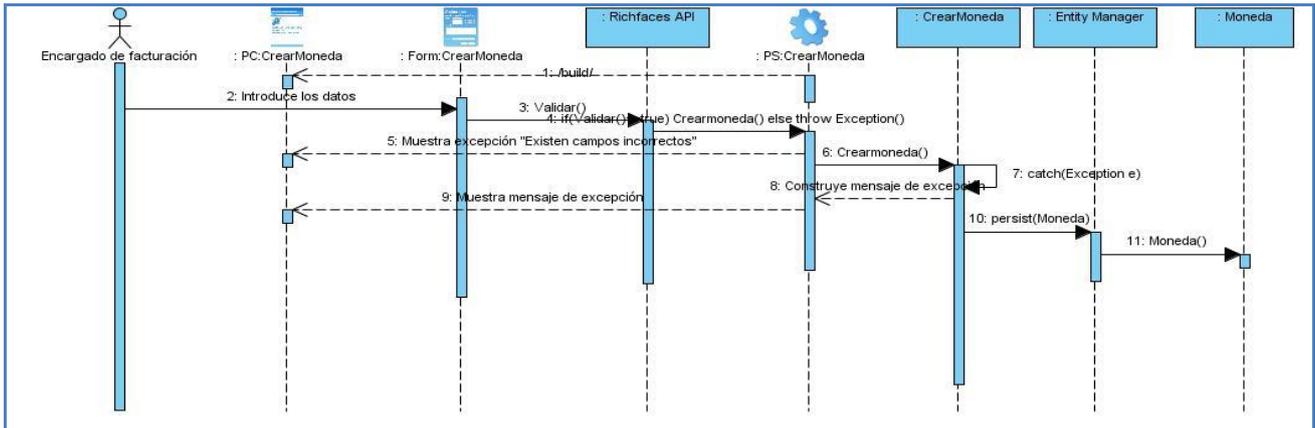


Ilustración 17: Diagrama de secuencia del CU “Crear moneda”.

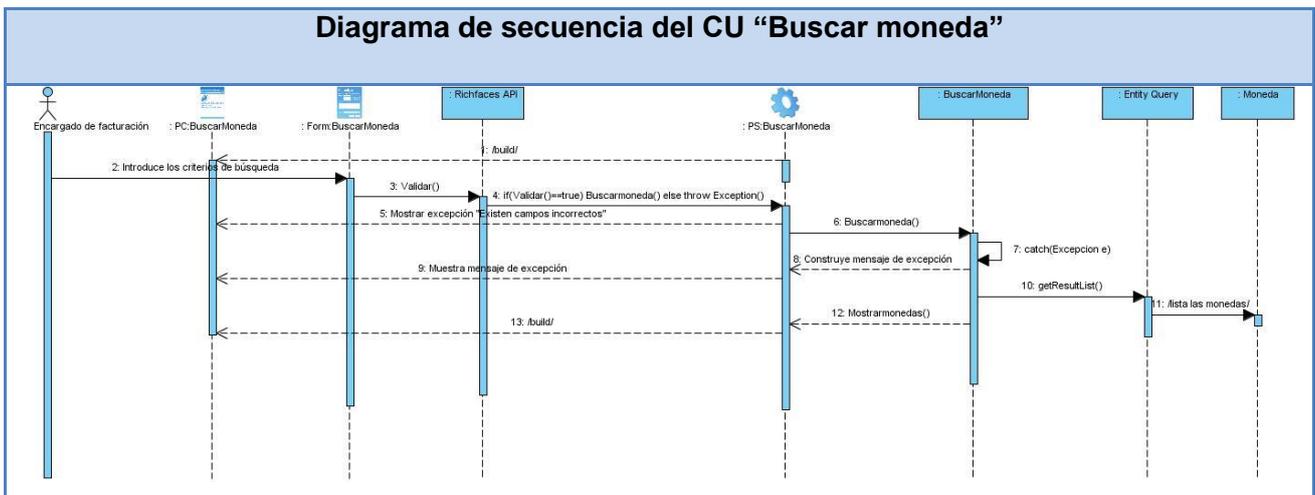


Ilustración 18: Diagrama de secuencia del CU “Buscar moneda”.

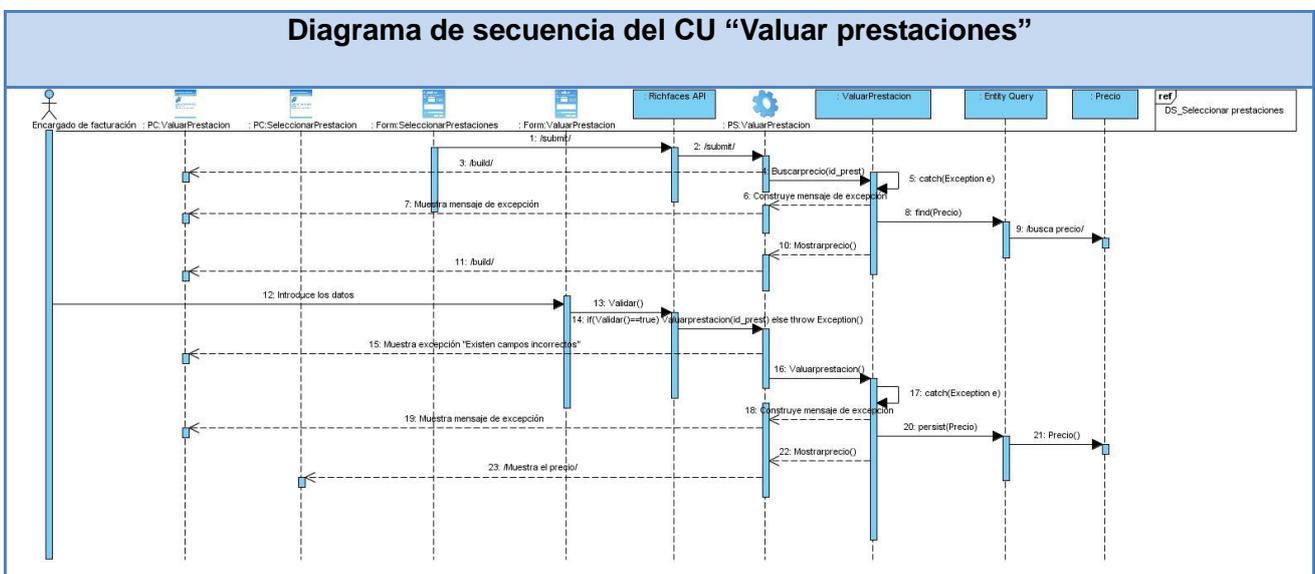


Ilustración 19: Diagrama de secuencia del CU “Valuar prestaciones”.

3.5.1. 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
id_factura	Integer	Identificador de la factura que se crea.
Id_user	Integer	Identificador del usuario al cual pertenece la factura.
Id_hoja frontal	Integer	Identificador de la historia clínica del paciente.
Id_moneda	Integer	Identificador de la moneda de trabajo establecida.
Eliminado	Boolean	Determina si la información ha sido eliminada.
Emitida	Boolean	Determina si la información ha sido emitida.
Fecha creación	Date	Informa la fecha de creación de la factura.
Fecha emisión	Date	Informa la fecha de emisión de la factura.
Id_receptor	Integer	Identificador de la persona que recibe la factura.

Tabla 19: Descripción de tabla “Factura”.

Prestaciones		
Descripción	Contiene la información de las distintas prestaciones que se ofrecen en el centro hospitalario a partir de las áreas existentes.	
Atributo	Tipo	Descripción
id_prestacion	Integer	Identificador de la prestación.
id_funcionalidad	Integer	Identificador de la funcionalidad de la prestación.
Nombre	Varchar	Nombre de la prestación.

Tabla 20: Descripción de tabla “Prestaciones”.

Moneda		
Descripción	Contiene la información de las distintas monedas de trabajo con las que se puede facturar en la institución.	
Atributo	Tipo	Descripción

id_moneda	Integer	Identificador de la moneda.
Símbolo	Char	Símbolo identificativo de la moneda.
Código	Char	Código identificativo de la moneda.
Nombre	Varchar	Nombre de la moneda.
Eliminado	Boolean	Determina si la información ha sido eliminada.

Tabla 21: Descripción de tabla “Moneda”.

Prestaciones paciente		
Descripción	Contiene la información referente a los gastos del paciente en las distintas áreas de la institución.	
Atributo	Tipo	Descripción
id_hoja frontal	Integer	Identificador de la historia clínica del paciente.
id_prestacion	Integer	Identificador de las prestaciones consumidas.
id_entidad	Integer	Identificador de la entidad que brinda la atención.
Fecha	Date	Fecha en que se brindó la prestación.

Tabla 22: Descripción de tabla “Prestaciones paciente”.

Consumo farmacia		
Descripción	Contiene la información referente a los gastos del paciente en el área de farmacia.	
Atributo	Tipo	Descripción
id_hoja frontal	Integer	Identificador de la historia clínica del paciente.
id_factura	Integer	Identificador de la factura.
Medicamento	Varchar	Nombre del medicamento consumido.
Cantidad	Decimal	Cantidad de medicamento consumido.
Precio	Decimal	Precio del medicamento consumido.

Tabla 23: Descripción de tabla “Consumo farmacia”.

Precios	
Descripción	Contiene la información referente a los gastos del paciente en las distintas áreas de la institución.

Capítulo 3: Análisis y Diseño del Sistema

Atributo	Tipo	Descripción
Precio	Decimal	Precio estándar de la prestación.
id_prestacion	Integer	Identificador de la prestación.
id_entidad	Integer	Identificador de la entidad que brinda la atención.
fecha_in	Date	Fecha en que se activó el precio de la prestación.
fecha_fin	Date	Fecha en que se desactivó el precio de la prestación.
precio_activo	Decimal	Precio activo de la prestación.
Eliminado	Boolean	Determina si la información ha sido eliminada.

Tabla 24: Descripción de tabla “Precios”.

A partir del desarrollo de este capítulo se llevó a cabo el flujo de trabajo Análisis y Diseño, el cual generó artefactos fundamentales para la elaboración de la solución propuesta. Durante esta etapa fueron definidas las clases del diseño y la relación e interacción entre estas; detallándose así, qué va a hacer el sistema y cómo va a lograr su correcto funcionamiento. Además, fue creado el modelo de datos, el cual describió todas las entidades persistentes en el sistema. La realización de estos artefactos propició el desarrollo del prototipo no funcional del sistema, quedando de este modo, asentadas todas las bases para la implementación del módulo Facturación del Sistema de Información Hospitalaria alas HIS.

Conclusiones

Los procesos de facturación, son complejos y requieren de precisión, incluso el estudio de estos, es una labor ardua que exige mucho esfuerzo. Durante la presente investigación, se lograron abordar de forma abarcadora, los procesos que rodean al área de facturación, lo cual propició que se le diera cumplimiento al objetivo general propuesto.

- El estudio del arte demostró que los sistemas existentes analizados no cumplían con las necesidades actuales de algunos usuarios, debido a sus características fundamentales; por lo que surgió la necesidad de crear una propuesta de sistema que satisfaga los nuevos intereses.
- El uso de las tecnologías web y el software libre para la elaboración del sistema, posibilita explotar los beneficios de la red y minimiza los costos de hardware en las instituciones hospitalarias.
- La facturación hospitalaria en Cuba, es un área aún en estudio y que necesita de gran explotación dentro de la informática, de modo que se visualizó la necesidad de profundizar en la misma con el apoyo de especialistas en el tema.
- El diseño y elaboración del módulo Facturación del Sistema de Información Hospitalaria alas HIS, facilitará la implementación del sistema, para poder así ofrecer una solución a los requerimientos definidos.

Recomendaciones

Luego de un detallado estudio de los procesos de facturación se logró realizar un prototipo no funcional del sistema que es un paso de avance para la consiguiente implementación del sistema. Toda la investigación, así como la ingeniería del sistema efectuada, proporcionaron un profundo conocimiento del negocio. Aún así, existen elementos que, aunque no fueron satisfechos en la investigación, no dejan de ser importantes para esta, por ello se proponen las siguientes recomendaciones:

- Estudiar con mayor profundidad los distintos procesos relacionados con la facturación en las instituciones hospitalarias, con apoyo de especialistas en economía y contabilidad.
- Investigar los métodos de gestión de costos en las instituciones hospitalarias para la posterior implementación de algoritmos que permitan realizar esta tarea con mayor precisión y factibilidad.
- Investigar los procesos de contrataciones realizados entre las empresas de seguros sociales y las instituciones hospitalarias, para asegurar la atención médica al paciente.
- Lograr que toda la infraestructura del HIS provea la información necesaria para facilitar la gestión de la facturación hospitalaria.

Referencias Bibliográficas

1. **Espasa, Arthur Andersen. 1997.** *Diccionario Espasa economía y negocios*. Madrid : s.n., 1997.
2. **Garrigues, A. 1987.** *Derecho mercantil*. Bogotá : s.n., 1987.
3. **2008.** Ser autónomo. [En línea] 2008. <http://www.serautonomo.net/tipos-de-factura.html>.
4. **2009.** Desoft. [En línea] 27 de noviembre de 2009. <http://www.desoft.cu/Productos1/BKMIS/tabid/474/Default.aspx>.
5. **2009.** Desoft. [En línea] 30 de noviembre de 2009. <http://www.desoft.cu/Portals/0/DESOFTE%20para%20la%20Sociedad%20Cubana.pdf>.
6. **2009.** Desoft. [En línea] 27 de noviembre de 2009. <http://www.desoft.cu/Productos1/Vindsor/tabid/437/Default.aspx>.
7. **2009.** Desoft. [En línea] 20 de noviembre de 2009. <http://www.desoft.cu/Portals/0/DESOFTE%20para%20la%20Sociedad%20Cubana.pdf>.
8. Adictos al trabajo. [En línea] <http://adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
9. **2009.** Seamframework. [En línea] 2009. <http://seamframework.org/>.
10. **2008.** Jboss. [En línea] 2008. http://downloads.jboss.com/drools/docs/5.0.1.26597.FINAL/drools-introduction/html_single/index.html.
11. **2009.** Kabuntu-es. [En línea] 28 de enero de 2009. <http://www.kubuntu-es.org/wiki/desarrollo-programacion/programas-desarrollo-libres>.
12. **2004.** Sistema gestor de base de datos. [En línea] 2004. http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
13. Web del programador. [En línea] <http://www.lawebdelprogramador.com/cursos/enlace.php?idp=5015&id=72 &texto=PostgreSQL>.

14. **Lockhart, Thomas.** *Manual del usuario de PostgreSQL.*
15. **2006.** Freedownloadmanager. [En línea] 2006. http://www.freedownloadmanager.org/es/downloads/facturaci%C3%B3n_de_opd_gratis/.
16. **2003.** Sergio Villagra. [En línea] 2003. [http://www.sergiovillagra.com/Contenidos/Recursos/WP03 Una Introduccion a CMMI.pdf](http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf).
17. **2003.** Ingenieros del software. [En línea] 2003 <http://www.ingenierossoftware.com/calidad/cmm-cmmi-nivel-2.php>.
18. **2009.** Rational. [En línea] 2009. <http://www.rational.com.ar/herramientas/rup.html>.
19. **2009.** IBM. [En línea] 2009. <http://www-01.ibm.com/software/awdtools/rup/>.
20. **2009.** Bizagi. [En línea] 2009. <http://www.bizagi.com/esp/descargas/BPMNbyExample.pdf>.
21. **2009.** BPMN. [En línea] 2009. <http://www.bpmn.org>.
22. **Fowler, Martin y Scott, Kendall. 2009.** *UML gota a gota.* 2009.
23. **Zaninotto, François y Potencier, Fabien. 2008.** Guía definitiva de Symfony. 2008.

Bibliografía

1. **Articulosinformativos.** [En línea] 2006. http://www.articulosinformativos.com.mx/Servicio_de_Facturacion_Medicinal-a854150.html#8062733.
2. **Bizagi.** [En línea] 2009. <http://www.bizagi.com/esp/descargas/BPMNbyExample.pdf>.
3. **BPMN.** [En línea] 2009. <http://www.bpmn.org>.
4. **Desoft.** [En línea] noviembre 27, 2009. <http://www.desoft.cu/Productos1/Vindsor/tabid/437/Default.aspx>.
5. **Desoft.** [En línea] noviembre 30, 2009. <http://www.desoft.cu/Portals/0/DESOFTE%20para%20la%20Sociedad%20Cubana.pdf>.
6. **Desoft.** [En línea] noviembre 27, 2009. <http://www.desoft.cu/Productos1/BKMIS/tabid/474/Default.aspx>.
7. **Freedownloadmanager.** [En línea] 2006. http://www.freedownloadmanager.org/es/downloads/facturaci%C3%B3n_de_opd_gratis/.
8. **Guía rápida de facturación.** [En línea] 2007.
9. **IBM.** [En línea] 2009. <http://www-01.ibm.com/software/awdtools/rup/>.
10. **Jboss.** [En línea] 2008. http://downloads.jboss.com/drools/docs/5.0.1.26597.FINAL/drools-introduction/html_single/index.html.
11. **Kubuntu-es.** [En línea] 2009. <http://www.kubuntu-es.org/wiki/desarrollo-programacion/programas-desarrollo-libres>.
12. **Net clínicas.** [En línea] 2009. <http://www.netclinicas.com/>.
13. **Procesos de facturación.** Córdova. s.l. : Michigan Ingeniería Informática S.A.
14. **Q-soft.** [En línea] 2008. <http://www.q-soft.net/DefaultSalus.aspx?gclid=CL7guOyGqZ4CFQiA5Qodw1SQrw>.
15. **Rational.** [En línea] 2009. <http://www.rational.com.ar/herramientas/rup.html>.
16. **Ser autónomo.** [En línea] 2008. <http://www.serautonomo.net/tipos-de-factura.html>.

17. **Softwaremas.** [En línea] <http://www.softwaremas.com/Facturacion/facturacion.htm>.
18. **Seamframework.** [En línea] 2009. <http://seamframework.org/>.
19. **Arbeloa, Carlos Serón. 2008.** Terra. [En línea] 2008. <http://www.terra.es/personal/cseron/tesis/informed.htm>.
20. **Autores, Colectivo de.** *Manual de factura electrónica avanzada.* Madrid : s.n. 84-611-4740-5.
21. **Autores, Colectivo de.** *Tesis C.M..* Madrid : s.n. 2009.
22. **Avramenko, Konstantin. 2008.** E-articles. [En línea] 02 2008. <http://e-articles.info/t/i/3769//es/>.
23. **Castro, Carmen L. 2008.** Mailxmail. [En línea] junio 17, 2008. <http://www.mailxmail.com/curso-administracion-oficina-medica/que-funcion-desempena-persona-encargada-facturacion-medica>.
24. **Castro, Raúl. 2007.** *Discurso pronunciado el 26 de julio.* Santiago de Cuba : 2007.
25. **Espasa, Arthur Andersen. 1997.** *Diccionario Espasa economía y negocios.* Madrid : s.n., 1997.
26. **Fernández, Erick R. Maldonado. 2009.** *Nuevos procesos de facturación y cobro a planes médicos.* Puesto Rico : Publicaciones Puertorriqueñas, 2009. 1-935145-26-6.
27. **Fowler, Martin y Scott, Kendall. 2009.** *UML Gota a Gota.* 2009.
28. **Garrigues, A. 1987.** *Derecho mercantil.* Bogotá : s.n., 1987.
29. **Lockhart, Thomas.** *Manual del usuario de PostgreSQL.*
30. **Pittiani, Mario and Alfonso, Eduardo Rodríguez. 2005.** *Hacia la definición de procesos de negocio seguros.* Castilla : s.n., 2005.
31. **Ramía, Germán Dueñas. 2009.** *Metodologías de cálculo de costos.* Madrid : s.n., 2009.
32. **Salus, Colectivo de.** Q-soft. *Dossier de descripción.* [En línea] <http://www.q-soft.net/>.
33. **Vignal, Juan. 2010.** *Conferencia de facturación.* 2010.
34. **Zaninotto, François and Potencier, Fabien. 2008.** *Guía definitiva de Symfony.* 2008.

Glosario de Términos

Factura: 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.

Factura electrónica: 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.

Facturación: Es una transferencia de crédito comercial de un titular a un tercero denominado "factor" que se encarga de su cobranza y que garantiza el buen fin, incluso en caso de impagos del deudor. La facturación se aplica principalmente para gestionar un control sobre todos los movimientos financieros que posee una empresa o negocio.

Facturación electrónica: Nuevo método de facturación que consiste en realizar los procesos de gestión de facturas utilizando dispositivos electrónicos. Este método consta de un documento factura que le es enviada al cliente por parte del vendedor de bienes o servicios, a través de un medio de comunicación a distancia y tiene como función, documentar la transacción realizada.

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. El mismo permite obtener un reporte del balance financiero de la institución, así como la notificación de los cobros pendientes.

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. Los impuestos son 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 al acto de consumir bienes y servicios tanto de alimentos, bebidas, indumentaria, artículos de tocador y perfumería, combustibles, servicios públicos, seguros, etc.

Anexos

1. Anexo: Diagramas de clases del diseño

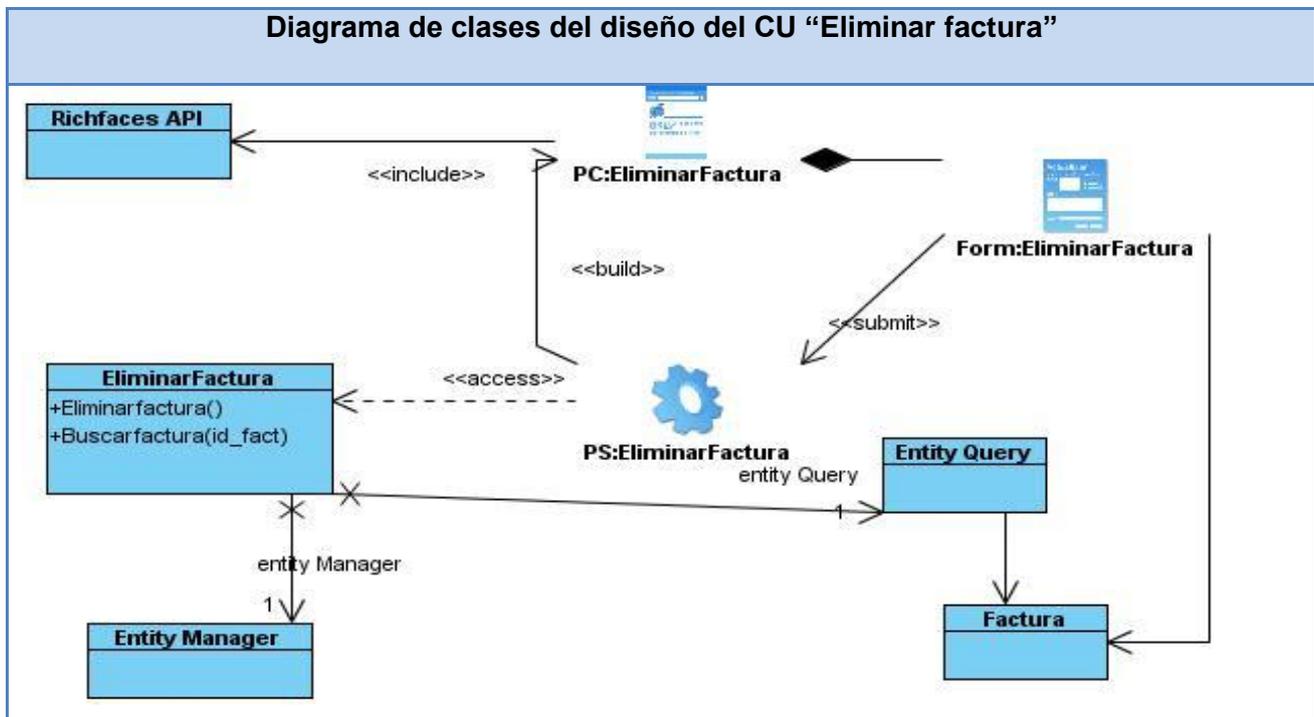


Ilustración 21: Diagrama de clases del diseño del CU "Eliminar factura"

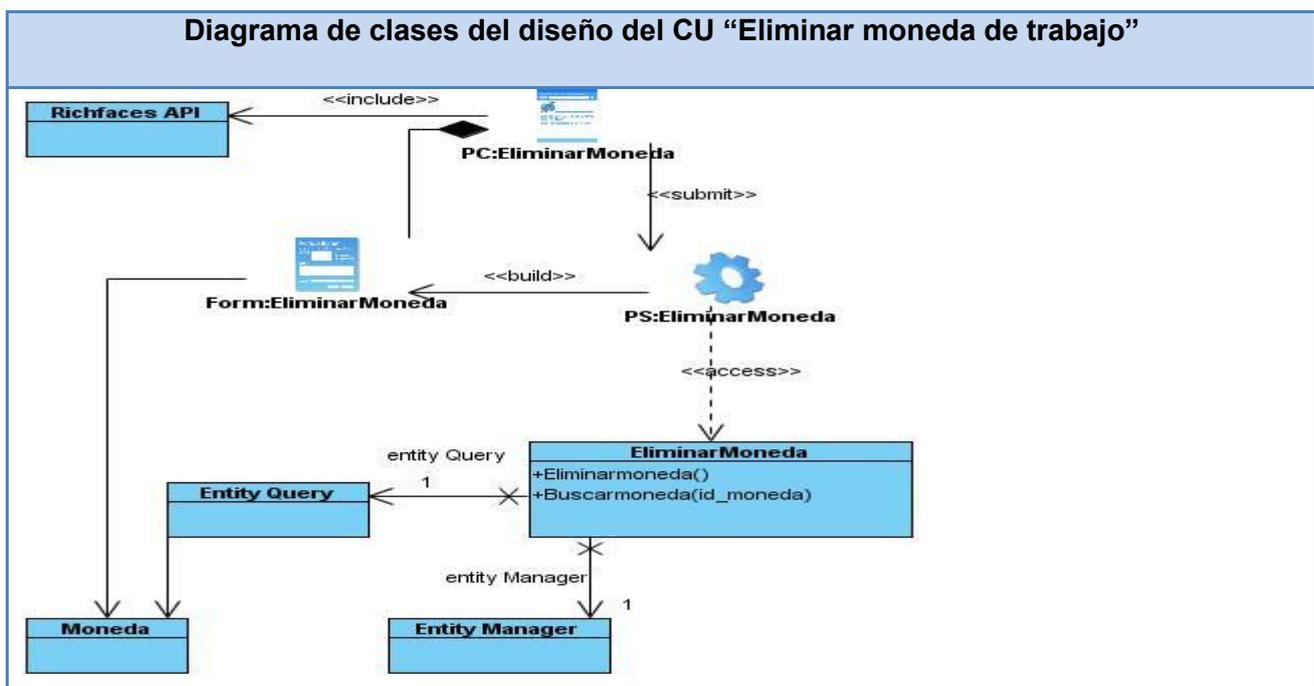


Ilustración 22: Diagrama de clases del diseño del CU "Eliminar moneda de trabajo"

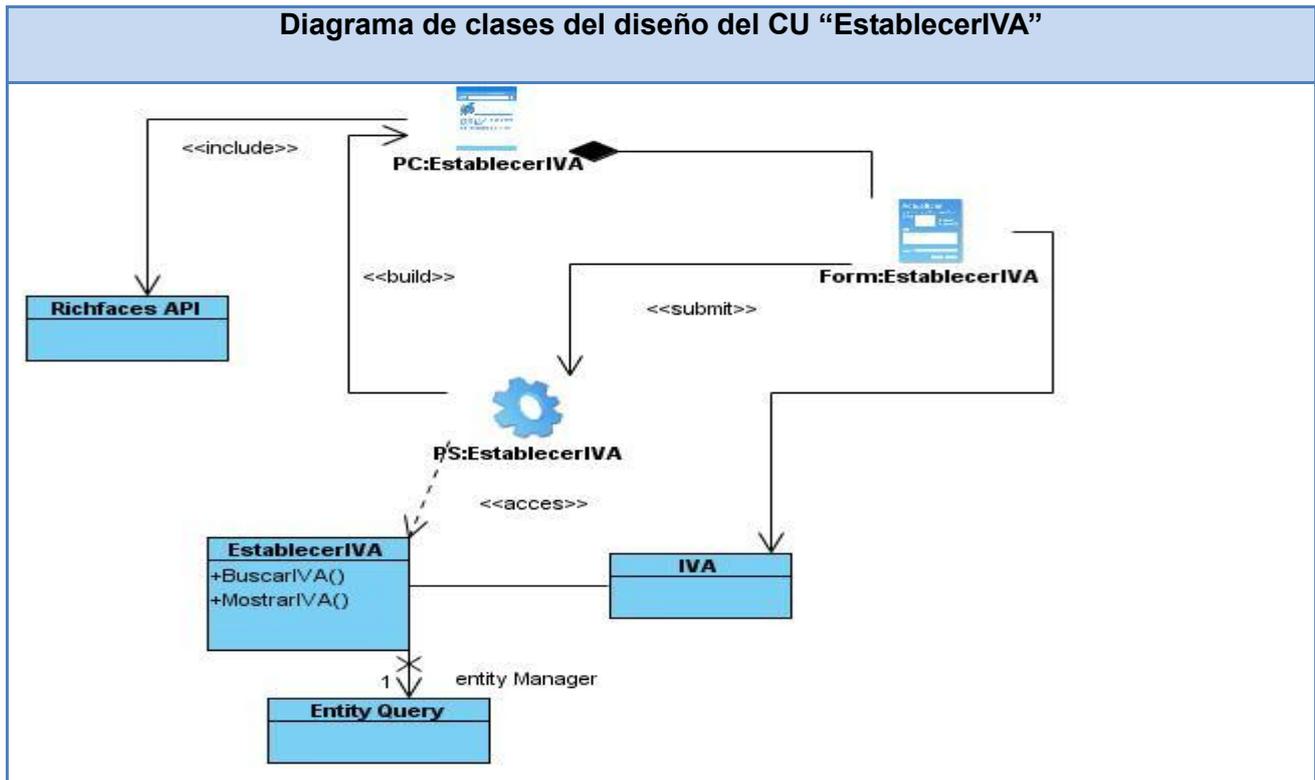


Ilustración 23: Diagrama de clases del diseño del CU "EstablecerIVA"

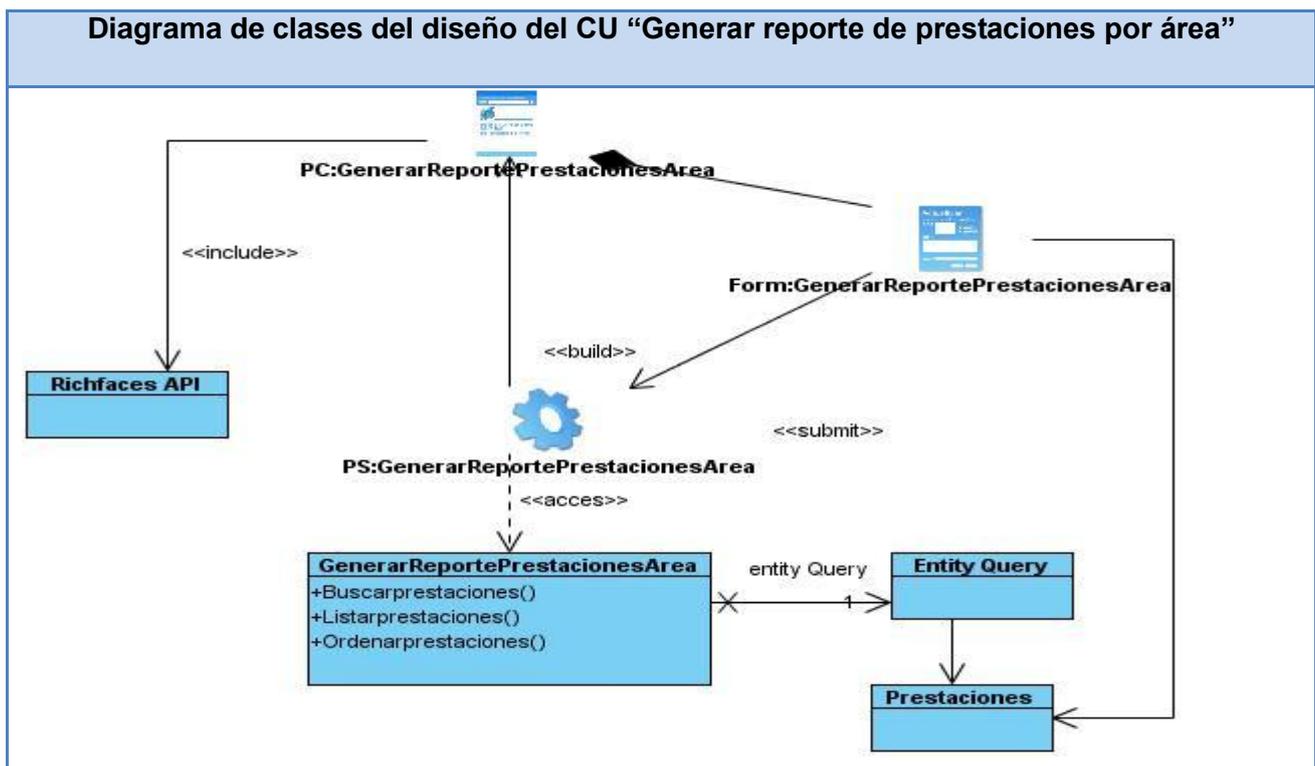


Ilustración 24: Diagrama de clases del diseño del CU "Generar reporte de prestaciones por área"

2. Anexo: Diagramas de secuencia del diseño

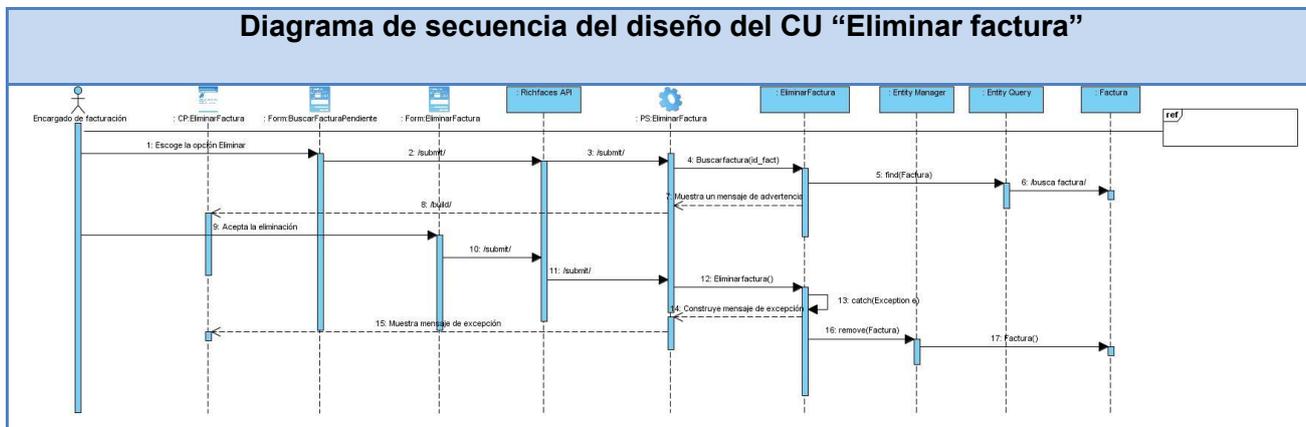


Ilustración 25: Diagrama de secuencia del diseño del CU “Eliminar factura”.

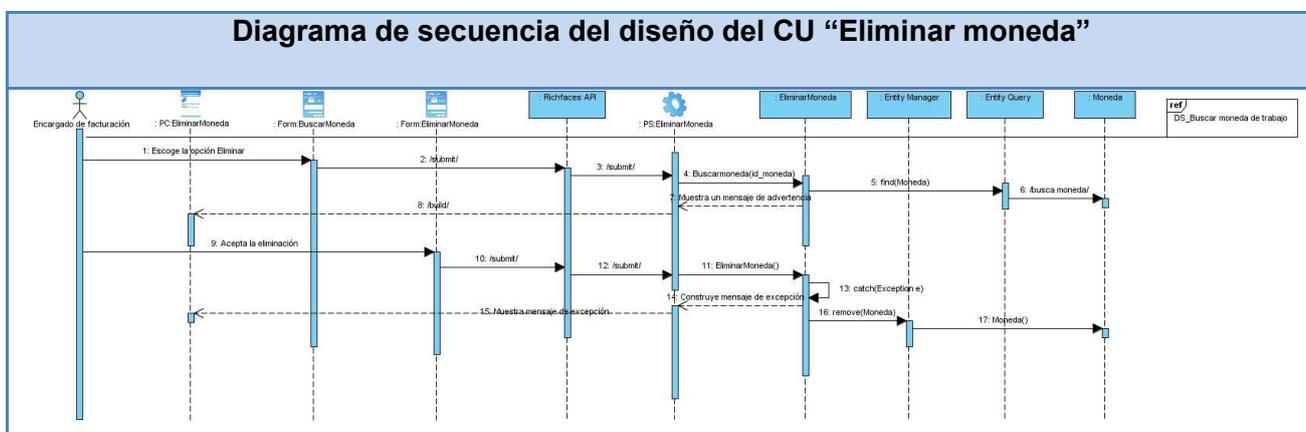


Ilustración 26: Diagrama de secuencia del diseño del CU “Eliminar moneda”.

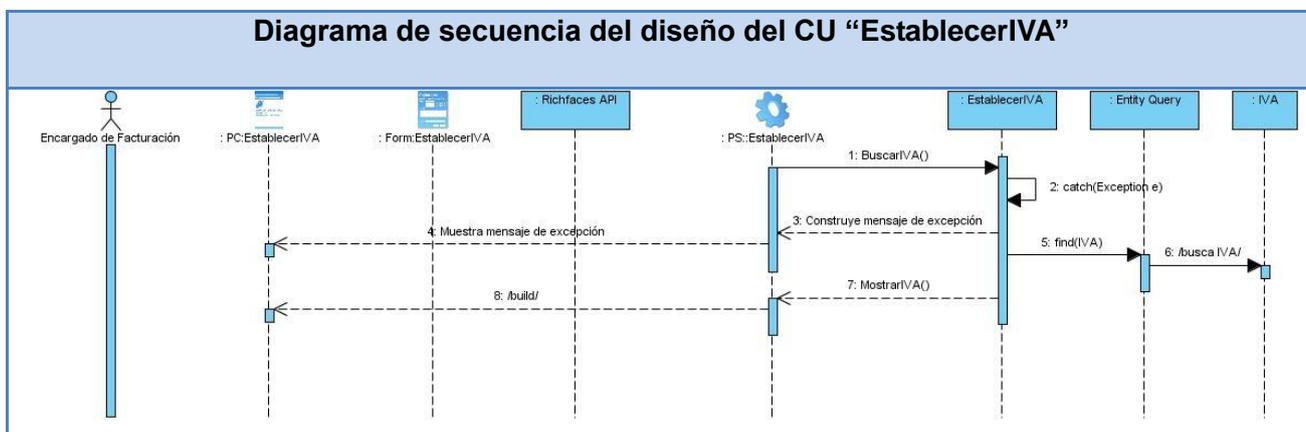


Ilustración 27: Diagrama de secuencia del diseño del CU “Eliminar factura”.

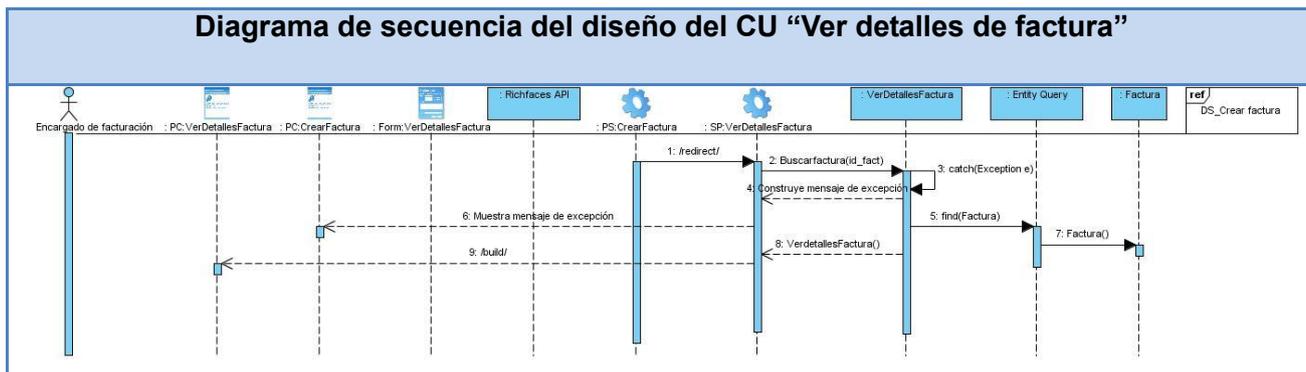


Ilustración 28: Diagrama de secuencia del diseño del CU "Ver detalles de factura".

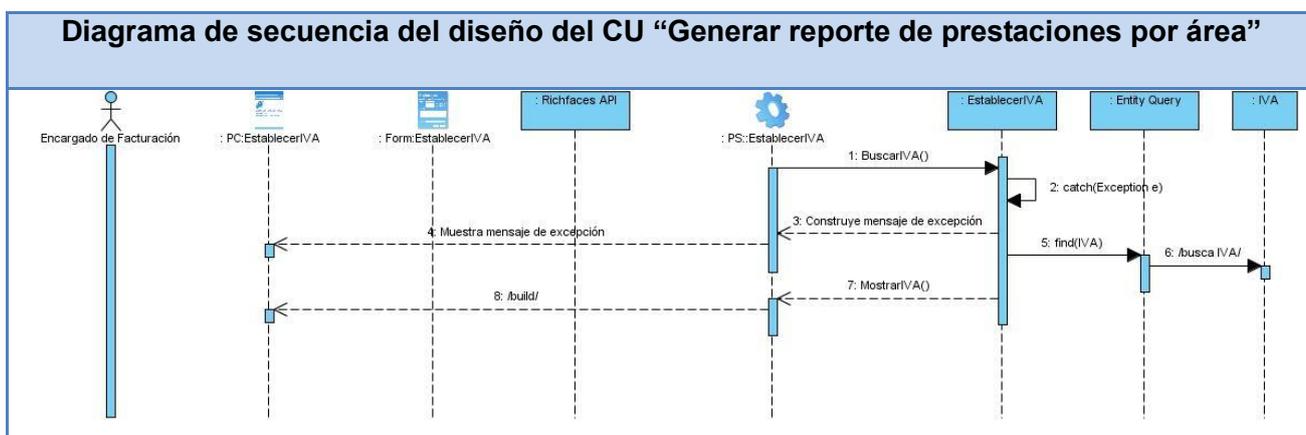


Ilustración 28: Diagrama de secuencia del diseño del CU "Generar reporte de prestaciones por área".

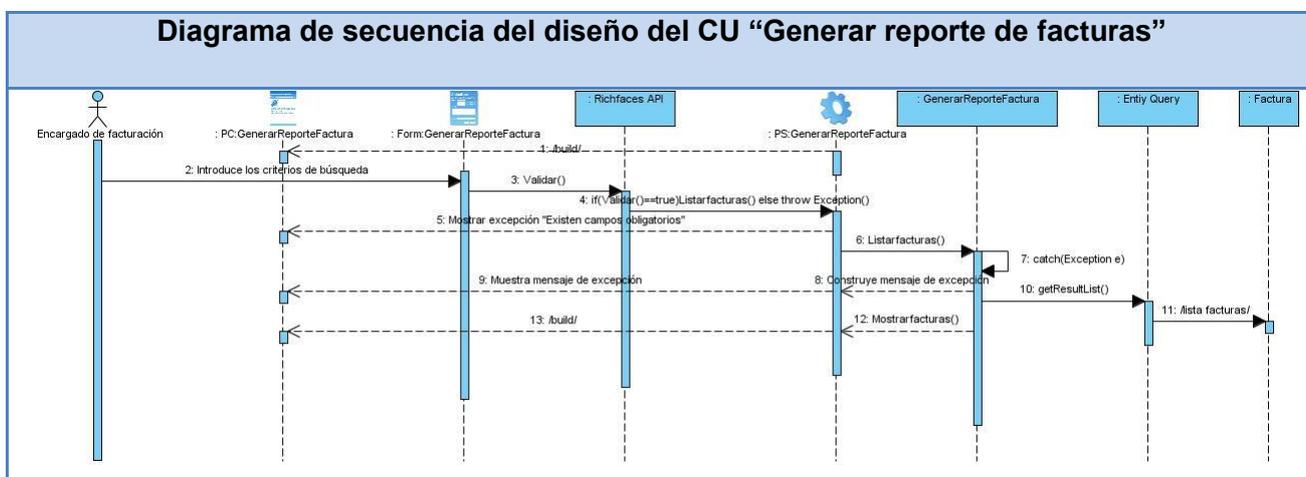


Ilustración 29: Diagrama de secuencia del diseño del CU "Generar reporte de facturas".