

Universidad de las Ciencias Informáticas

Facultad 2



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

Submódulo para contribuir a la gestión de la información de la Unidad de Sangrado en el Centro Nacional de Cirugía de Mínimo Acceso.

Autores: Annies Laydys Izaguirre Salazar

Iraldo Arévalo Delfín

Tutores: Ing. Amaya Alvarez Lorenzo

Ing. Yuliet Oliva Romero

La Habana, junio de 2015

“Año 57 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 23 días del mes de junio del año 2015.

Autores:

Annie Laydis Izaguirre Salazar

Iraldo Arévalo Delfín

Tutores:

Ing. Amaya Alvarez Lorenzo

Ing. Yuliet Oliva Romero

DATOS DE CONTACTO

Tutores:

Ing. Amaya Alvarez Lorenzo. Ingeniera en Ciencias Informáticas, graduada en la Universidad de las Ciencias Informáticas en el año 2012. Especialista, miembro del Departamento de Aplicaciones del Centro de Informática Médica (CESIM). Actualmente se encuentra vinculada al desarrollo del Sistema de Información Hospitalaria del CESIM.

Correo electrónico: aalorenzo@uci.cu

Ing. Yuliet Oliva Romero. Ingeniera en Ciencias Informáticas, graduada en la Universidad de las Ciencias Informáticas en el año 2013. Miembro del Departamento de Aplicaciones del Centro de Informática Médica (CESIM). Actualmente se encuentra vinculada al desarrollo del Sistema de Información Hospitalaria del CESIM.

Correo electrónico: yoromero@uci.cu

RESUMEN

El Centro de Informática Médica desarrolla el Sistema de Información Hospitalaria (HIS). El mismo está compuesto por módulos vinculados a cada una de las áreas de las instituciones hospitalarias. El módulo Emergencias del HIS brinda la posibilidad de gestionar el flujo de información en los procesos de recepción de pacientes que necesitan atención urgente, así como la atención de los mismos dentro de la institución, sin embargo no satisface las necesidades requeridas en los flujos de trabajo de la Unidad de Sangrado del Centro Nacional de Cirugía de Mínimo Acceso.

Con el objetivo de dar solución a la gestión de información de la Unidad de Sangrado, se realizó el análisis de los procesos de negocio asociados a dicha área, se evaluaron los sistemas existentes con objetivos similares, así como las tendencias tecnológicas para su implementación. La información obtenida del análisis realizado, determinó la necesidad de implementar un submódulo para el módulo Emergencias del HIS. A partir de estas decisiones se planificaron y ejecutaron las tareas del proceso de desarrollo del submódulo, las cuales de forma resumida fueron la gestión de requisitos, el modelado del negocio, el diseño y la implementación.

Como resultado de estas tareas se obtuvieron los artefactos derivados de cada flujo de trabajo, obteniendo un producto acabado y funcional, que incorpora funcionalidades no presentes en otros sistemas. Se podrá acceder a la información histórica de los pacientes para facilitar los diagnósticos, tratamientos y seguimientos, así como el desarrollo de investigaciones médicas mediante los reportes de información.

Palabras clave: emergencias, información, paciente, unidad de sangrado.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS CONCEPTOS ASOCIADOS A LAS UNIDADES DE SANGRADO DE LAS INSTITUCIONES HOSPITALARIAS.....	5
1.1. CONCEPTOS BÁSICOS RELACIONADOS CON EL DOMINIO DEL PROBLEMA	5
1.2. SOLUCIONES EXISTENTES A NIVEL INTERNACIONAL Y NACIONAL	6
1.3. TENDENCIAS Y TECNOLOGÍAS A CONSIDERAR.....	10
1.3.1. <i>Arquitectura cliente-servidor</i>	10
1.3.2. <i>Tecnologías utilizadas en el proceso de desarrollo</i>	12
1.4. METODOLOGÍAS DE DESARROLLO DE SOFTWARE	17
1.5. HERRAMIENTAS	18
CAPÍTULO 2. CARACTERÍSTICAS DEL SUBMÓDULO UNIDAD DE SANGRADO	21
2.1. FLUJO ACTUAL DE LOS PROCESOS INVOLUCRADOS EN EL CAMPO DE ACCIÓN	21
2.2. OBJETO DE INFORMATIZACIÓN	22
2.3. MODELADO DEL NEGOCIO	23
2.4. REQUISITOS FUNCIONALES DEL SISTEMA.....	26
2.5. REQUISITOS NO FUNCIONALES DEL SISTEMA	27
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SUBMÓDULO UNIDAD DE SANGRADO.....	36
3.1. DESCRIPCIÓN DE LA ARQUITECTURA, FUNDAMENTACIÓN.....	36
3.2. ESTRATEGIAS DE INTEGRACIÓN	37
3.3. MODELO DE DISEÑO	38
CAPÍTULO 4. IMPLEMENTACIÓN DEL SUBMÓDULO UNIDAD DE SANGRADO	45
4.1. MODELO DE DATOS	45
4.1.1. <i>Descripción de las tablas de la base de datos</i>	47
4.2. MODELO DE IMPLEMENTACIÓN	49
4.2.1. <i>Diagramas de componentes</i>	50
4.2.2. <i>Diagrama de despliegue</i>	51
4.3. TRATAMIENTO DE ERRORES	52
4.4. SISTEMA DE SEGURIDAD	53

4.5. ESTRATEGIAS DE CODIFICACIÓN	53
CONCLUSIONES	55
RECOMENDACIONES.....	56
REFERENCIAS BIBLIOGRÁFICAS.....	57
BIBLIOGRAFÍA	62
GLOSARIO DE TÉRMINOS.....	65
ANEXOS	66
ANEXO 1: DIAGRAMA DE PROCESOS DE NEGOCIO	66
ANEXO 2: DIAGRAMAS DE CASOS DE USO.....	67
ANEXO 3: DIAGRAMA DE CLASES DEL DISEÑO.....	69
ANEXO 4: FUNCIONALIDADES INCORPORADAS DEL MÓDULO EMERGENCIAS	72

ÍNDICE TABLAS

TABLA 1.1 COMPARACIÓN DE LOS SISTEMAS EXISTENTES QUE GESTIONAN LA INFORMACIÓN DEL ÁREA DE EMERGENCIAS, UNIDAD DE SANGRADO	9
TABLA 2.1 TRABAJADORES DEL NEGOCIO	26
TABLA 2.3 ACTORES DEL SISTEMA.....	30
TABLA 2.4 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: RECIBIR PACIENTE.....	34
TABLA 2.5 DESCRIPCIÓN TEXTUAL DEL CASO DE USO: ATENDER PACIENTE	34
TABLA 3.1 DESCRIPCIÓN DE LA CLASE CONTROLADORA: RECIBIR PACIENTE CONTROLADOR_SANGRADO	43
TABLA 4.1 DESCRIPCIÓN DE LOS ATRIBUTOS COMUNES DENTRO DE TODAS LAS ENTIDADES...	47
TABLA 4.2 DESCRIPCIÓN DE LA TABLA: EVAL_PRE_ENDOSC	48
TABLA 4.3 DESCRIPCIÓN DE LA TABLA: EXAMEN_FISICO	48
TABLA 1 FUNCIONALIDADES DE EMERGENCIAS	72

ÍNDICE FIGURAS

FIGURA 2.1: DIAGRAMA DEL PROCESO DE NEGOCIO: RECIBIR PACIENTE.....	25
FIGURA 2.2 DIAGRAMA DE ACTORES.....	31
FIGURA 2.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA: RECIBIR PACIENTE.....	32
FIGURA 2.4 DIAGRAMA DE CASOS DE USO DEL SISTEMA: ATENDER PACIENTE.....	33
FIGURA 3.1 DIAGRAMA DE INTEGRACIÓN.....	38
FIGURA 3.2 DIAGRAMA DE PAQUETES.....	40
FIGURA 3.3 DIAGRAMA DE CLASES DEL DISEÑO: RECIBIR PACIENTE.....	41
FIGURA 4.1 MODELO DE DATOS.....	46
FIGURA 4.2 SUBSISTEMAS DE IMPLEMENTACIÓN POR CAPAS.....	51
FIGURA 4.3 DIAGRAMA DE DESPLIEGUE.....	52
FIGURA 1 PROCESO ATENDER PACIENTE.....	66
FIGURA 2: DIAGRAMA DE CASOS DE USO DEL SISTEMA: ATENDER PACIENTE.....	67
FIGURA 3: DIAGRAMA DE CASOS DE USO DEL SISTEMA: GENERAR REPORTE.....	68
FIGURA 4: DIAGRAMA DE CLASES DEL DISEÑO: ACTUALIZAR HOJA DE EMERGENCIA DE SANGRADO	69
FIGURA 5: DIAGRAMA DE CLASES DEL DISEÑO: CREAR EVALUACIÓN TRANS-ENDOSCÓPICA....	70
FIGURA 6: DIAGRAMA DE CLASES DEL DISEÑO: CREAR EVALUACIÓN PRE-ENDOSCÓPICA.....	71

INTRODUCCIÓN

En la actualidad la información se ha convertido en el eje promotor de cambios sociales, económicos y culturales. El auge de las telecomunicaciones ha producido una transformación de las Tecnologías de la Información y la Comunicación (TIC), cuyo impacto ha alcanzado a todos los sectores de la economía y de la sociedad. (1)

Más allá de la descripción de las propias tecnologías y de las numerosas prestaciones que hacen posible su utilización en cualquier actividad, negocio o tarea que se realiza, uno de los aspectos más importantes de las TIC son sus aplicaciones concretas y específicas para cada sector social. Entre los sectores beneficiados con el uso de las TIC, se encuentra el de salud. La cantidad de información que se maneja en el mismo, debe estar correctamente clasificada y accesible en cualquier momento, posibilitando que el servicio que se brinde sea óptimo en la medida de lo posible. Este sector influye considerablemente en el progreso de la sociedad, constituyendo un elemento de vital importancia para elevar la calidad de vida de las personas.

Con el objetivo de contribuir a mejorar la atención del paciente y los servicios que se brindan en los centros de salud se crearon los Sistemas de Información Hospitalaria conocidos por sus siglas en inglés como HIS (Hospital Information System). La función de un HIS es la de apoyar las actividades en los niveles operativos, tácticos y estratégicos dentro de un hospital. (2) Como resultado, permiten la recolección, almacenamiento, procesamiento, recuperación y comunicación de la información relacionada con la atención al paciente y la labor administrativa de todas las actividades del hospital o centro de atención médica. Contribuyen a la toma de decisiones administrativas, clínico-epidemiológicas, estratégicas y operativas.

En un inicio los sistemas que existían en los hospitales se dedicaban a tareas administrativas, sistemas para admisión y altas de pacientes, toma de órdenes, revisión de resultados, control de inventarios, entre otras. Luego, su uso se extendió al área clínica, lo que facilitó acceso a la Historia Clínica, y a la documentación bibliográfica de consulta. (3) Actualmente un sistema informático que gestione la información generada en un hospital, debe contener: el registro del paciente con expediente digital, control de laboratorios y quirófanos desde una sala de mando, generación de recetas, órdenes o formularios de trabajo y generación de reportes estadísticos.

Los sistemas más avanzados en su desarrollo no solo permiten la gestión de la información, estos brindan a los pacientes una estancia más agradable en el hospital y proveen las herramientas necesarias para que el

personal sanitario pueda compartir con el afectado, imágenes, resultados de pruebas, vídeos y demás materiales a través de los diferentes dispositivos electrónicos como tablets y smartphones, ayudándole a conocer y comprender su estado de salud. De esta forma se genera una mayor colaboración entre pacientes y personal médico. (4)

La Universidad de las Ciencias Informáticas (UCI), es un centro de altos estudios que vincula en la producción de software a estudiantes y profesores. La UCI se encuentra efectuando negociaciones con diferentes instituciones hospitalarias con el fin de implantar un HIS desarrollado en el Centro de Informática Médica (CESIM). El sistema está compuesto por varios módulos que interactúan en las diferentes áreas de un hospital como son: Admisión, Citas, Consulta Externa, Bloque Quirúrgico, Banco de Sangre, Laboratorio, Enfermería, Emergencias, Farmacia, y otros.

El Centro Nacional de Cirugía de Mínimo Acceso (CNCMA) es uno de los centros de salud interesados en poner en funcionamiento el HIS. El CNCMA es un instituto de referencia nacional dedicado a la terapéutica endoscópica y a la cirugía mínimamente invasiva. Constituye el Centro de Educación de Postgrado del Instituto Superior de Ciencias Médicas, dedicado a la capacitación, entrenamiento y certificación de los médicos especialistas nacionales y extranjeros en el dominio de las técnicas básicas y de avanzadas de la cirugía de mínimo acceso, es una Unidad de Ciencia e Innovación. (5)

El CNCMA tiene características muy propias como es el caso de la existencia de una Unidad de Sangrado. Esta área funciona las 24 horas del día como una unidad de urgencias, pero trabaja de forma más selectiva y realizando diferentes estudios y/o tratamientos a los pacientes que se reciben. El módulo Emergencias del HIS brinda la posibilidad de gestionar el flujo de información en los procesos de recepción de pacientes que necesitan atención urgente, así como la ubicación del mismo dentro de la institución para ser atendido, sin embargo no satisface las necesidades requeridas en los flujos de trabajo de la Unidad de Sangrado.

En el sistema no es posible registrar los datos asociados a la aprobación o no de la remisión de los pacientes referidos al CNCMA debido a un sangrado gastrointestinal, ni de la recepción de los mismos en la unidad. Presenta limitaciones para clasificar el paciente, de acuerdo al tipo de tratamiento que deba recibir. Los datos asociados a estudios y tratamientos terapéuticos realizados y la conducta a seguir no se contemplan durante la atención al paciente. El sistema no genera la información asociada a la cantidad de pacientes que fueron atendidos, al total de pacientes que fueron egresados ni al total de pacientes según conducta a seguir. Parte de la información que se recoge acerca del estado del paciente se realiza de forma manual, provocando

inconvenientes, como la pérdida de información relacionada con la actividades que se llevan a cabo en la institución y la omisión de datos que pueden ser necesarios a la hora de emitir diagnósticos. La información se encuentra dispersa, haciendo el trabajo del médico difícil y lento, dificultando la toma de decisiones en determinados momentos.

Por todo lo anteriormente expuesto se plantea como **problema a resolver**: ¿cómo contribuir a la gestión de la información relacionada con los procesos de la Unidad de Sangrado del Centro Nacional de Cirugía de Mínimo Acceso?

El **objeto de estudio** lo constituye el proceso de gestión de información en las áreas de emergencias de las instituciones hospitalarias. El mismo se enmarca en el **campo de acción** proceso de gestión de información de la Unidad de Sangrado en el Centro Nacional de Cirugía de Mínimo Acceso.

Para resolver el problema planteado, se define como **objetivo general**: desarrollar un submódulo que contribuya a la gestión de la información de la Unidad de Sangrado en el Centro Nacional de Cirugía de Mínimo Acceso.

Para dar cumplimiento al objetivo general se definen las siguientes **tareas de la investigación**:

1. Descripción de los procesos de negocio asociados a la Unidad de Sangrado en el Centro Nacional de Cirugía de Mínimo Acceso.
2. Identificación de los sistemas existentes que abarquen la información sobre las Unidades de Sangrado de instituciones hospitalarias, para establecer similitudes con la investigación en curso.
3. Asimilación de la arquitectura y pautas definidas para el desarrollo de los artefactos, el diseño y las aplicaciones del Sistema de Información Hospitalaria del Centro de Informática Médica.
4. Generación, basado en el Proceso Unificado de Desarrollo, de los artefactos resultantes del Modelado de Negocio, la Gestión de Requisitos, el Diseño y la Implementación.
5. Implementación de los procesos identificados en la Unidad de Sangrado del Centro Nacional de Cirugía de Mínimo Acceso.

Para el desarrollo de este trabajo se utilizaron los siguientes **métodos de investigación**:

Métodos Teóricos

- **Análisis histórico-lógico**: se utilizó para hacer la valoración de sistemas que gestionan la información en las áreas de emergencias de las instituciones hospitalarias, a nivel nacional e internacional. Permitió

el estudio y análisis del sistema ProGastro, utilizado para el registro de la información de los pacientes atendidos en la Unidad de Sangrado del Centro Nacional de Cirugía de Mínimo Acceso.

- **Inductivo-deductivo:** se empleó con la finalidad de interpretar la información obtenida durante el estudio de los procesos de gestión de información relacionada con la Unidad de Sangrado, complementándose mutuamente en el proceso de desarrollo del conocimiento científico.
- **Modelación:** se utilizó para modelar a través de diagramas, los procesos que intervienen en la Unidad de Sangrado.

Al finalizar la investigación se espera obtener los siguientes beneficios:

- Registro único e integrado con la información de los pacientes que se atienden en la Unidad de Sangrado del Centro Nacional de Cirugía de Mínimo Acceso.
- Alta disponibilidad de los registros del paciente.

El documento está estructurado en cuatro capítulos:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LOS CONCEPTOS ASOCIADOS A LAS UNIDADES DE SANGRADO DE INSTITUCIONES HOSPITALARIAS: se realiza un estudio preliminar de los sistemas existentes que gestionan la información de la Unidad de Sangrado en las instituciones hospitalarias. A partir de la arquitectura definida del Sistema de Información Hospitalaria, se fundamentan las tecnologías, metodologías y herramientas de desarrollo a utilizar.

CAPÍTULO 2: CARACTERÍSTICAS DEL SUBMÓDULO UNIDAD DE SANGRADO: se realiza una descripción de las funcionalidades de la solución propuesta a la problemática planteada. Contiene un marco conceptual asociado a la información que será procesada por el sistema y las funcionalidades, requerimientos deseados y el objeto de informatización.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SUBMÓDULO UNIDAD DE SANGRADO: se describe la arquitectura y los patrones de diseño utilizados durante el desarrollo del módulo Unidad de Sangrado. Se realiza el diseño mediante los diagramas de clases del diseño y la descripción de las clases que lo conforman.

CAPÍTULO 4: IMPLEMENTACIÓN SUBMÓDULO UNIDAD DE SANGRADO: se describe la implementación de las clases y subsistemas en términos de componentes de la solución propuesta. Se expone la propuesta de solución para lograr una gestión más eficiente de los procesos hospitalarios asociados al área en cuestión.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS CONCEPTOS ASOCIADOS A LAS UNIDADES DE SANGRADO DE LAS INSTITUCIONES HOSPITALARIAS

En el presente capítulo se describen los principales conceptos asociados a las Unidades de Sangrado de las instituciones hospitalarias, y el análisis realizado sobre algunos de los sistemas que gestionan la información referente a las emergencias de sangrado. También se fundamentan las herramientas y tecnologías a utilizar en el desarrollo de la propuesta de solución.

1.1. Conceptos básicos relacionados con el dominio del problema

En el área de *Emergencias* se reciben a los pacientes que necesitan atención médica de forma inmediata. También es preciso aclarar que bajo este mismo concepto se emplea ocasionalmente el término de Urgencias, respetando la bibliografía consultada. La Unidad de Sangrado es un área de Emergencias donde se les brinda atención particular a pacientes que presentan hemorragia digestiva alta, utilizando todos los medios disponibles en el arsenal terapéutico para el tratamiento de esta afección, considerando los medios más modernos de tratamiento. (6)

La hemorragia digestiva alta (HDA) se define como el sangrado originado en el tracto digestivo superior, proximal al ángulo de Treitz¹, y constituye la urgencia gastroenterológica más importante. Los avances en el diagnóstico y tratamiento médico endoscópico y quirúrgico han mejorado el pronóstico y la evolución clínica de los pacientes con hemorragia digestiva alta. La mortalidad ha permanecido estable en la última década entre el 7-10%, debido al incremento de las enfermedades concomitantes y la edad de la población. (7)

La recepción en la Unidad de Sangrado de los pacientes que presentan sangrado digestivo alto está determinada por diferentes criterios clínicos y analíticos que son medidos por la escala de Blatchford², escala de riesgo para predecir la necesidad de tratamiento. De tal manera los pacientes con el puntaje mayor que

¹ Conexión entre el duodeno y el yeyuno, es la parte más fijada del sistema digestivo. Sobre ese punto se desarrollara todo lo que es la movilidad y motilidad de la osteopatiavisceral.

² Predice la necesidad de tratamiento o intervención en el paciente con hemorragia, utilizando la urea sérica, hemoglobina, presión arterial, pulso, presencia de melena, síncope, enfermedad hepática o falla cardíaca.

seis tienen alto riesgo, tributario de tratamiento endoscópico de urgencia. En los pacientes con el puntaje menor e igual que seis tienen riesgo bajo, puede realizarse la endoscopia de urgencia, con baja probabilidad de realizar alguna de las variantes de tratamiento endoscópico. (8)

Independientemente del tratamiento endoscópico a realizar se informará a los departamentos de Laboratorio de Urgencia, Banco de sangre, Unidad de Terapia, y Cirugía el traslado del paciente. Esto se realiza con el objetivo de agilizar el proceso de recepción.

1.2. Soluciones existentes a nivel internacional y nacional

A continuación se presentan las principales características de los sistemas que se consideraron más significativos para el objeto de estudio de la investigación.

Sistema de Información para la Gerencia Hospitalaria (Sigho)

Aplicación web basada en la Norma Oficial Mexicana NOM-168-SSA1-1998 referente al resguardo y uso del expediente clínico electrónico, para facilitar las actividades de gerencia dentro del hospital. Desarrollada con las mejoras realizadas en los servicios de Internet Information Server 6.0 (IIS 6.0), Microsoft ASP.NET y Microsoft .NET Framework. Este software se apoya de estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos, tales como el CIE-10 y CIE-9MC. Como gestor de base de datos utiliza SQL Server 2000. El sistema operativo que utiliza es Windows y en una versión Server 2000 o superior. (9)

Este sistema está compuesto por 14 módulos (2 administrativos y 12 relacionados con la atención al paciente). Permite realizar registros individuales alrededor de la Historia Clínica Electrónica. Entre los módulos que lo componen está Hospitalización, que gestiona la información que se genera en la atención de las áreas de Urgencia, Enfermería e Ingreso.

Sigho brinda como funcionalidades:

- Listar los pacientes por cada uno de los servicios y áreas,
- Registrar la Historia Clínica del paciente;
- Elaborar notas médicas tales como Ingreso, Evolución, Egreso, Traslado, Interconsulta;
- Registrar Indicaciones médicas (dietas, medicamentos, soluciones, cuidados y control de líquidos);
- Realizar el seguimiento de las indicaciones por parte de enfermería;

- Registrar los diagnósticos mediante una codificación; notificar al médico los diagnósticos asignados cuando no corresponden de acuerdo al sexo y edad del paciente
- Notificar a Epidemiología la existencia de diagnósticos de interés.

Con la implementación de SIGHO se mejora el tiempo y calidad de atención del paciente, evitando realizar el registro de información en diferentes y diversos formatos con la finalidad de convertirse en un sistema único.

GalenHospital

En Cuba, se han desarrollado también sistemas de gestión hospitalarios, tal es el caso de la empresa SOFTEL que brinda soluciones para la salud. Uno de los productos desarrollados por SOFTEL es Galen Hospital, sistema que está orientado hacia la informatización de la gestión de pacientes como elemento básico de control para mejorar la atención médica. (10)

Uno de los módulos del Galen Hospital es el de Urgencias. El módulo de Urgencias registra la recepción y atención de los pacientes cuyo estado de salud es grave; mantiene el control de los atendidos y el de los servicios y medicamentos usados; indica cambios en la atención; emite reportes variados y consulta el registro general de pacientes. Este módulo consta de tres programas: Recepción de Cuerpo de Guardia, Hoja de Cargo de Cuerpo de Guardia, Hoja de Cargo Médica de Cuerpo de Guardia.

En la recepción de Cuerpo de Guardia se realiza el registro de los datos primarios de la consulta del paciente que acude con alguna dolencia o padecimiento. Además se registra la información sobre la hora de atención y el médico que atenderá al paciente. Permite listarse los datos de los pacientes que han sido atendidos en el día, donde se muestran los campos: nombre del paciente, hora de llegada y tiempo de espera. Este programa brinda otras funcionalidades como: establecer los médicos de guardia, agregar consulta, buscar un paciente, pasar a consulta, buscar consultas, eliminar consultas y generar reporte de casos atendidos.

Este sistema utiliza como gestor de bases de datos SQL Server 2000 y sus módulos corren sobre los sistemas operativos Windows 2000 y Windows XP, por lo que los principales inconvenientes de este sistema es que es una aplicación que se ejecuta en el escritorio y además depende de tecnologías que son propiedades de compañías privadas.

MediSys®EM

El producto MediSys®EM también forma parte de un HIS. Está financiado por la Dirección General de Salud y Consumo de la Comisión Europea y desarrollado por la Joint Research Centre. Es un sistema informático

diseñado y construido para agilizar las actividades realizadas en el área de Emergencias de un establecimiento de salud. Posibilita registrar la información general del paciente y su clasificación en la hoja de emergencia. (11)

Este sistema permite obtener un reporte de las consultas efectuadas en el servicio, así como las que fueron realizadas por cada médico. Con MediSys®EM los médicos, el personal de enfermería y administrativo pueden acceder y actualizar la información del paciente, y enviar las órdenes médicas. Es una solución web que utiliza como gestor de base de datos para plataforma cliente/servidor SQL Server 6.5. (12)

X-HIS

Solución desarrollada por la empresa CSC *Enterprise Corp.*, que gestiona la información clínica y administrativa en un entorno abierto, y permite la integración con otros sistemas de información. Está conformado por los módulos: Recursos Humanos, Farmacia, Admisión, Consultas Externas, Bloque Quirúrgico, Anatomía Patológica, Urgencias, entre otros.

A través del módulo Urgencias los usuarios del sistema podrán realizar el registro, control y tratamiento de un paciente en Urgencias, desde que es ingresado hasta que es dado de alta, enlazándose además con los módulos de Admisión y Consultas Externas. Así quedan registrados todos los datos clínicos y administrativos del paciente, además de generar toda la información administrativa como: reportes judiciales, informes clínicos, autorizaciones, entre otros. (13)

Está desarrollado de acuerdo a los siguientes estándares: CMBD, Codificación ICD -9-CM/ICD 10, SNOMED, HL7, NOC, NANDA, entre otros. Es multiplataforma (Unix, Windows, Linux), tiene soporte para arquitectura cliente/servidor en tres capas. Posee independencia de bases de datos: MS SQL SERVER, ORACLE, INFORMIX, ADAPTIVE SERVER, SYBASE. (14)

ProGastro

Es un sistema automatizado desarrollado en Cuba en el año 2007, el cual fue puesto a prueba inicialmente en el Instituto de Gastroenterología de Cuba (IGE), con el registro de todos los procedimientos que el centro y el sistema brinda. Desarrollado en el lenguaje de programación Delphi, empleando estructura de base de datos SQLite3 con soporte para la encriptación, el cual puede ser instalado o portable, en sistemas operativos Windows® (15)

Para la estructuración de los datos a registrar, se consideraron los parámetros generales del paciente como individuo, y todos aquellos relacionados con cada proceder en específico. Se tuvieron en cuenta los registros de los últimos 5 años del IGE y la experiencia de profesores del Servicio de Endoscopia Digestiva del propio centro. Una vez puesto en marcha durante casi 6 meses, se comenzó a implantar en otros servicios de La Habana hasta la fecha.

Diseñado para el registro de forma fácil y rápida la descripción íntegra, esencial y específica de las Endoscopias Digestivas Superiores, Ileocolonoscopias, Enteroscopias, Colangiopancreatografía Retrógrada Endoscópica (CPRE) y Laparoscopias, de cualquier servicio de Gastroenterología. El objetivo fundamental de dicho registro es el propio paciente, a partir del cual se computan todos y cada uno de los procedimientos que se le practiquen evolutivamente. De esta forma, se crea realmente una historia clínica digital virtual (objetivo primordial del Ministerio de Salud Pública en Cuba), accesible en cualquier momento y por cualquier servicio.

En resumen, como parte de la investigación sobre los antecedentes de los sistemas que gestionan la información relacionada con la Unidad de sangrado, se obtuvieron los resultados que se muestran en la Tabla 1.1.

Tabla 1.1 Comparación de los sistemas existentes que gestionan la información del área de emergencias, Unidad de Sangrado

	Tipo de software	Licencia	Bases de datos	Plataforma	Estándares
SIGHO	Web	Propietario	SQL Server 2000	Windows	CIE-10 CIE 9CM
GalenHospital	Desktop	Propietario	SQL Server 2000	Windows	CIE-10
MediSys®EM	Web	Propietario	SQL Server 6.5	Windows	CIE-10
X-HIS	Desktop	Propietario	MS,SQL Server, ORACLE, INFORMIX, ADAPTIVE SERVER, SYBASE	Multi-plataforma	CMBD, ICD-9- CM/ICD 10, SNOMED, HL7, NOC, NANDA
ProGastro	Desktop	Propietario	SQLite3	Windows	CIE-10 CIE 9CM

Al realizar un estudio de los diferentes productos existentes en el mercado, que dan solución total o parcial al problema relacionado con la gestión de la información de la Unidad de Sangrado como un área de emergencias, se arribó a las conclusiones siguientes:

- Estas soluciones en su mayoría presentan licencias de software propietario, por tanto se debe invertir tanto en la compra del producto, como en el pago de licencias de sistemas gestores de base de datos, servidores y sistema operativo.
- En su mayoría son aplicaciones de escritorio, lo que crea ciertos inconvenientes como la necesidad de instalación, y dificulta el proceso de soporte.
- Las soluciones existentes no cumplen con las exigencias de trabajo que se realiza en la Unidad de Sangrado; no se integran con el resto de las áreas hospitalarias que prestan servicios complementarios para la atención del paciente.

Como resultado de este análisis se determina la necesidad de desarrollar una aplicación, que abarque los procesos realizados en la Unidad de Sangrado y que gestione la información asociada a la atención al paciente. Se demuestra que es necesario incluir en la informatización de los procesos asociados a la Unidad de Sangrado, actividades como la coordinación de la remisión del paciente al Centro Nacional de Cirugía de Mínimo Acceso, y los diferentes tratamientos que se le realizan a los pacientes con hemorragia digestiva alta. Además generar los reportes asociados a estas actividades, almacenando por cada una de ellas información detallada que contribuya a diagnósticos y estudios posteriores. La gestión de la información asociada a la atención al paciente debe realizarse de forma amigable, fácil y segura.

1.3. Tendencias y tecnologías a considerar

El desempeño de un proyecto está antecedido por la investigación de las tecnologías de avanzada que se utilizan, así como las ventajas y desventajas que trae su aplicación. Para realizar la presente investigación se asimilaron las tecnologías en el desarrollo del Sistema de Información Hospitalaria del Centro de Informática Médica.

1.3.1. Arquitectura cliente-servidor

Para lograr que el sistema sea accesible desde las más disímiles ubicaciones se propone el desarrollo de una aplicación web. Las aplicaciones web se basan en la arquitectura cliente-servidor. Este modelo consiste

básicamente en que un programa (el cliente) realiza peticiones a otro programa (el servidor) que le da respuesta. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo y los servidores de correo. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. (12)

Este tipo de aplicación posibilita tener los datos almacenados centralmente en bases de datos, accesibles desde un navegador web o una terminal móvil, gracias a que la lógica se ejecuta en el servidor y el diseño del interfaz es transferido a dichas terminales. Los requerimientos de hardware para estas aplicaciones solo se limitan a contar con un servidor web potente y una conectividad permanente y relativamente rápida. La instalación y despliegue de las mismas es sencilla, pues solo necesita montarse el servidor de aplicaciones y este será accedido por los clientes, independientemente de la plataforma, la arquitectura de máquina y la localización física que tengan dichas computadoras. El mantenimiento de estas aplicaciones es relativamente sencillo, pues los cambios son publicados en el servidor y los clientes acceden a estos, por tanto no hay necesidad de instalar las actualizaciones requeridas en cada estación de trabajo.

Patrones de Arquitectura y Diseño

Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. (16)

Modelo Vista Controlador (MVC)

Modelo Vista Controlador (por sus siglas en inglés MVC), es un patrón de diseño de arquitectura de software que separa la parte lógica de una aplicación de su presentación. Básicamente sirve para separar el lenguaje de programación del código HTML lo máximo posible y para poder reutilizar componentes fácilmente. Utilizar este patrón brinda la facilidad de agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de otras capas y facilita el mantenimiento en caso de errores. (17)

Modelo: representa las estructuras de datos. Las clases del modelo contendrán funciones para consultar, insertar y actualizar información de la base de datos.

Vista: es la información presentada al usuario y permite la interacción de este con la aplicación. Una vista puede ser una página web o una parte de ella.

Controlador: actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página web. Recibe los eventos de entradas desde la vista, procesa estos y devuelve la respuesta correspondiente, consultando si es necesario a las clases del modelo.

Arquitectura en capas

La arquitectura en capas es un estilo de diseño cuyo objetivo primordial es la separación y agrupamiento de los componentes del software atendiendo a la función que cumplen en el mismo. Para realizar el agrupamiento se tiene en cuenta las funcionalidades relacionadas con el usuario del sistema, así como la información que este gestiona y las operaciones que realiza sobre la misma en dependencia de la complejidad necesaria para el sistema. Esta división muchas veces se hace en tres capas: la capa de presentación, la capa de negocio y la capa de datos.

1.3.2. Tecnologías utilizadas en el proceso de desarrollo

En este epígrafe se exponen las tecnologías y herramientas utilizadas en el proceso de desarrollo. Estas aparecerán relacionadas a continuación según su ubicación en las capas de presentación, negocio y acceso a datos, separadas las que no estén ubicadas en ninguna de estas capas, así como una relación de las herramientas propuestas.

Java

Con el objetivo de desarrollar un sistema libre de costos de patentes asociadas a software como sistemas operativos y bases de datos se utiliza Java. Este es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones web con arquitectura del tipo cliente-servidor. (18)

Java Platform Enterprise Edition (JavaEE 5)

Java Platform Enterprise Edition o Java versión 5 es una plataforma de programación (parte de la plataforma Java) para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N niveles distribuida. Se basa ampliamente en componentes de software modulares y se ejecuta sobre un servidor de aplicaciones. (19) Las tecnologías utilizadas durante el desarrollo en las diferentes capas del sistema están soportadas sobre esta plataforma.

Java Runtime Environment (JRE 6)

JRE es el acrónimo de Java Runtime Environment (entorno en tiempo de ejecución Java) y se corresponde con un conjunto de utilidades que permite la ejecución de programas Java sobre todas las plataformas soportadas. La máquina virtual de java (JVM) es una instancia de JRE en tiempo de ejecución. Este interpreta el código Java y está compuesto además por las librerías de clases estándar que implementan el API de Java. Ambas JVM y API deben ser consistentes entre sí, de ahí que sean distribuidas de modo conjunto. (20)

JBoss Application Server

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4 e incluye servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta EJB 3.0, lo que hace el desarrollo de las aplicaciones mucho más simple. Además, al ser desarrollado con tecnología Java, es multiplataforma. (21)

Capa de presentación

La capa de presentación es la que presenta el sistema al usuario, le comunica y captura la información la información que este introduce en un mínimo de procesos. Esta capa se comunica únicamente con la capa de negocio. (22)

Java Server Faces (JSF)

JSF es un marco de trabajo (framework) para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JSP como la tecnología que permite hacer el despliegue de las páginas, facilita y agiliza el diseño de interfaces de usuario, pues implementa una serie de componentes, estado de los mismos, eventos del lado de servidor, entre otras ventajas. (23)

RichFaces

RichFaces: es un framework de código abierto que añade capacidad Ajax dentro de aplicaciones JSF existentes sin recurrir a JavaScript. RichFaces incluye ciclo de vida, validaciones, conversiones y la gestión de recursos estáticos y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax que puede ser fácilmente incorporado dentro de las aplicaciones JSF. (24) Esta librería de componentes visuales es ampliamente utilizada durante el desarrollo de las interfaces de usuario.

Ajax4JSF

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan del lado del cliente, es decir, en el navegador de los usuarios, mientras se mantiene en segundo plano la comunicación asíncrona con el servidor. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas totalmente, lo que significa aumentar la interactividad, velocidad y usabilidad de las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en lenguaje de marcado extensible (XML). Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM). (25)

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. (26)

Facelets

Facelets es un framework para plantillas centrado en la tecnología JSF, lo cual permite que JSP y JSF puedan funcionar conjuntamente en una misma aplicación web, ya que estos no se complementan naturalmente. JSP procesa los elementos de la página de arriba a abajo, mientras que JSF dicta su propio rendering pues su ciclo de vida está dividido en fases marcadas. Facelets llena este vacío entre JSP y JSF, siendo una tecnología centrada en crear árboles de componentes y estar relacionado con el complejo ciclo de vida JSF. (27) Facelets permite la separación de los componentes de interfaces de usuario de la solución desarrollada en diferentes archivos.

XHTML

XHTML, acrónimo inglés de Extensible Hypertext Markup Language (Lenguaje Extensible de Marcado de Hipertexto), es el lenguaje de marcado pensado para sustituir al lenguaje HTML (HyperText Markup Language) como estándar para las páginas web. Su objetivo es avanzar en el proyecto del World Wide Web Consortium (W3C) de lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas. (28) XHTML combina los lenguajes XML (eXtensible Markup Language) y HTML en un mismo formato. Es el lenguaje de etiquetas utilizado en JSF.

Cascading Style Sheets (CSS)

Las hojas de estilo en cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Lo que se persigue con el desarrollo de CSS es separar la estructura de un documento de su presentación. (29)

Capa de negocio

La capa de negocio es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para almacenar o recuperar los mismos. (12)

JBoss Seam

JBoss Seam es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB), funcionando, según versa su significado en español, como una “costura” entre estos componentes. Seam también se integra perfectamente con otros frameworks como: RichFaces, ICE Faces, MyFaces, Hibernate y Spring. (30)

Capa de acceso a datos

La capa de acceso a datos contiene las clases que interactúan con la base de datos. Estas clases altamente especializadas, utilizan los procedimientos almacenados para realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio. (12)

Hibernate

Es una herramienta de mapeo objeto-relacional para la plataforma Java, que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación. Utiliza para esto archivos declarativos (XML) que permiten establecer estas relaciones. Es una tecnología de software libre distribuida bajo los términos de la licencia GNU LGPL.

Como todas las herramientas de su tipo, 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). Le permite a la aplicación manipular los datos de la base de datos operando sobre objetos, es decir, con todas las características de la programación orientada a objetos. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. (31)

Esta herramienta genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias. Logra mantener la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución. Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente conocida como "Criteria". (32)

Enterprise Java Beans (EJB3)

Enterprise JavaBeans (EJB), es una de las API que forman parte del estándar de construcción de aplicaciones empresariales Java Enterprise Edition (JEE). Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB. Los EJB proporcionan un modelo distribuido y estándar de componentes que se ejecutan en el servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.), para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. (33) La herramienta de mapeo objeto-relacional Hibernate implementa las especificaciones de EJB3.

Java Persistence API (JPA)

Java Persistence API, más conocida por su sigla JPA, es la API de persistencia desarrollada para la plataforma Java EE y está incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue su diseño es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, y permitir usar objetos regulares conocidos como POJOs (Plain Old Java Object). (34) La herramienta de mapeo objeto-relacional Hibernate hace uso de esta API para la persistencia de datos.

1.4. Metodologías de desarrollo de software

Proceso Unificado de Desarrollo (RUP)

RUP es el resultado de varios años de trabajo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. En RUP se han agrupado las actividades en grupos lógicos en los que se definen nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. El ciclo de vida de RUP se caracteriza por ser dirigido por caso de uso, centrado en la arquitectura, iterativo e incremental. (35)

Lenguaje Unificado de Modelado (UML)

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Permite la modelación de sistemas con tecnología orientada a

objetos. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso utilizar. Este lenguaje de modelado formal permite tener un mayor rigor en la especificación, realizar una verificación y validación del modelo desarrollado, automatizar determinados procesos y generar código a partir de los modelos y a la inversa. Esto último permite que el modelo y el código estén actualizados. (36)

Notación para Gestión de Procesos de Negocio

BPMN (Business Process Management Notation) es un nuevo estándar de modelado de procesos de negocio donde se presentan gráficamente las diferentes etapas de su proceso. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

El objetivo principal de esta notación es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. A través del modelado de las actividades y los procesos puede lograrse un mejor entendimiento del negocio. Muchas veces esto brinda un mejor enfoque, lo que permite mejorarlos. (37)

1.5. Herramientas

Eclipse

Eclipse es un Entorno de Desarrollo Integrado (del inglés IDE), de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-Liviano" basadas en navegadores. Esta plataforma típicamente ha sido usada para desarrollar entornos de desarrollo integrados como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte del Eclipse. (38)

Este IDE emplea módulos (en inglés plugins) para proporcionar toda su funcionalidad al frente de la plataforma de los llamados clientes enriquecidos. 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. (38)

JBoss Tools

Es un conjunto de plugins para Eclipse que permite el manejo de diferentes frameworks que facilitan el desarrollo de aplicaciones. Está constituido por varios módulos: RichFaces VE, Seam Tools, Hibernate Tools y JBoss AS Tools. (39) Estos módulos añaden a Eclipse las funcionalidades necesarias durante la fase de implementación.

Sistemas de Gestión de Base de Datos

Los Sistemas de Gestión de Base de Datos (DBMS por sus siglas en inglés); son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los SGBD es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante. Entre los gestores de base de datos se destacan MySQL, Microsoft SQL-Server, Oracle y PostgreSQL. (40)

PostgreSQL Server

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacional (Object-Relational Database Management System (ORDBMS) de código abierto, por lo que se puede disponer de su código fuente, modificarlo y redistribuirlo libremente. PostgreSQL garantiza concurrencia, para lo cual utiliza la tecnología de Control de Concurrencia Multi-Versión (Multiversion concurrency control (MVCC), con lo que se logra que ningún lector sea bloqueado por un escritor. Es extensible, soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Tiene una adecuada documentación, además de contar con una comunidad de usuarios y desarrolladores a los que acudir en caso de tener problemas. (41)

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 la construcción de aplicaciones de calidad, y con un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (42)

Conclusiones parciales

- El flujo real de trabajo que se realiza en la Unidad de Sangrado no es posible gestionarlo en los sistemas estudiados.
- Los sistemas analizados no se integran con el resto de las áreas de un centro hospitalario.
- Se identificaron como funcionalidades principales: la coordinación del paciente, el registro de los tratamientos a un paciente con hemorragia digestiva alta y los reportes vinculados a las actividades realizadas.

CAPÍTULO 2. CARACTERÍSTICAS DEL SUBMÓDULO UNIDAD DE SANGRADO

En este capítulo se describen los procesos actuales que se llevan a cabo en la Unidad de Sangrado. Para obtener una representación gráfica de estos, se muestra el modelo de negocio así como los actores involucrados. Se especifican los requisitos funcionales y no funcionales del sistema y los casos de uso del sistema a desarrollar como respuesta a la situación problemática actual.

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

La Unidad de Sangrado es un área de Emergencias, donde se les brinda atención particular a pacientes que presentan hemorragia digestiva alta. Para **recibir al paciente** en la Unidad de Sangrado tiene que haber una previa coordinación por parte de la institución remitente con el Centro Nacional de Cirugía de Mínimo Acceso. El paciente debe estar estabilizado y con la vena canalizada para poder ser trasladado. Si el paciente cumple con las características para que sea recibido, entonces se traslada.

Una vez el paciente se encuentra en el centro, se le realiza un chequeo y se le estabiliza la hemodinámica, todos estos datos se actualizan en la Historia Clínica del paciente. Después de chequear los parámetros se valora si el paciente debe ser atendido en el centro, si es así, se registran los datos del paciente como son: nombre, apellidos, carnet de identidad y el motivo de su consulta. En caso de que el paciente no esté estabilizado se regresa a la institución remitente.

La atención médica es el proceso más importante dentro de la Unidad de Sangrado dada su función de salvaguardar la vida del paciente. Después que el paciente es recibido en el centro, se pasa a la atención del mismo. Se le realiza un interrogatorio minucioso para saber las causas de su padecimiento, y se miden los signos vitales. También se les hace un reconocimiento mediante el examen físico que permite identificar los órganos o sistemas de órganos lesionados y describir los hallazgos positivos encontrados en ellos. Todos los resultados arrojados en el estudio se registran en la Historia Clínica del paciente.

Durante la **atención al paciente** el médico define si la endoscopia de urgencia que se va a realizar, es con tratamiento endoscópico o sin tratamiento endoscópico. En el primer caso, se le hace una evaluación de anestesia al paciente y se crea una Historia Clínica de anestesia, luego es trasladado a Cirugía donde se le realiza la intervención, y el médico crea un informe operatorio, el reporte endoscópico y se le dan indicaciones al paciente para su posterior recuperación.

Al egresar el paciente del postoperatorio, se sigue la conducta posterior del mismo, y si el criterio del médico es favorable se traslada al paciente a su hospital de origen. Si el paciente no se recupera satisfactoriamente y necesita de otras atenciones, se remite a la Unidad de Cuidados Intensivos, o a un Hospital Clínico Quirúrgico.

Si el tratamiento que se le realiza al paciente es una endoscopia de urgencia sin tratamiento endoscópico, una vez realizada la endoscopia, se crea un reporte de endoscopia. Luego el médico le indica al paciente lo que debe de hacer para su posterior recuperación, trasladándolo a la sala de cuidados intermedios. En esta sala el médico decide si está apto para regresar a su hospital de origen, a la Unidad de Cuidados Intensivos, o a un Hospital Clínico Quirúrgico.

2.2. Objeto de informatización

Una vez descrito y analizado el funcionamiento del área de la Unidad de Sangrado se definen los procesos a informatizar y se realiza la depuración de las actividades manuales, quedando aquellas que pasen a ser funcionalidades del sistema. Se informatizarán los procesos de esta área, a partir de la inclusión de un submódulo en el HIS, lo que facilitará el trabajo realizado en el hospital y por lo tanto la atención al paciente.

El primer objeto a informatizar es la Hoja de sangrado que será asociada a la Historia Clínica del paciente, como elemento contenedor de toda su atención médica en el transcurso de su vida. La Hoja de sangrado se crea para cada paciente que llega a la Unidad de Sangrado. Si se realiza una lista automática, se podrá obtener la relación de pacientes atendidos, mostrándose los campos más importantes como el nivel de gravedad del paciente, sus datos de identificación, motivo de la consulta, entre otros.

Se podrá realizar búsquedas sobre las Historias Clínicas, lo que proporciona un fácil y rápido acceso a la información médica del paciente. Mediante la informatización de la Historia Clínica se logra visualizar información para la atención urgente al paciente, ejemplo de ello lo constituye: los antecedentes personales, familiares, hábitos psicobiológicos y las inmunizaciones. Otros datos que ayudan a la decisión del diagnóstico son: el examen físico y el interrogatorio, que aportan la información de los medicamentos habituales que el paciente ingiere y la enfermedad que padece, así como los órganos o sistemas de órganos que tiene afectados.

Las solicitudes de los exámenes complementarios también serán informatizadas por lo que el médico podrá solicitar los exámenes complementarios de análisis de laboratorio y solicitudes de estudios radiológicos e

imagenológicos en tiempo real. También podrá solicitar las transfusiones de sangre en caso que el paciente lo necesite y realizará una descripción del transcurso del proceso, como historial de reacciones del paciente.

Luego que el paciente recibe el tratamiento, se crea un reporte endoscópico y un informe operatorio, donde se recogen los principales datos del paciente después de ser atendido. También se le realizan indicaciones médicas al paciente y si esta incluye algún medicamento se crea una receta. El sistema permitirá seleccionar los medicamentos a prescribir para cada uno de los tratamientos.

El sistema permitirá generar reportes de acuerdo a los diferentes criterios de selección. Los reportes abarcan toda la información referente a los procesos que se llevan a cabo en la Unidad de Sangrado, desde que el paciente es recibido en el centro, hasta su egreso, constituyendo un elemento clave para la toma de decisiones. Debe brindar la funcionalidad de referir a los pacientes a su hospital de origen si la conducta posterior del paciente es satisfactoria, o coordinar el traslado del paciente a un Hospital Clínico Quirúrgico, y en caso de que necesite intervención, trasladarlo al quirófano.

Clasificador Internacional de enfermedades (CIE)

Para el desarrollo del submódulo se hizo necesario la utilización del Codificador Internacional de Enfermedades (CIE) en su décima edición, unificando las terminologías asociadas a las enfermedades y procedimientos en la medicina. La finalidad del CIE es clasificar las enfermedades, las afecciones y las causas externas de enfermedades y traumatismos. El empleo de este codificador le permite al médico obtener una impresión diagnóstica de los casos atendidos, el registro sistemático, el análisis, la interpretación y comparación de los datos de mortalidad y morbilidad de los pacientes.

2.3. Modelado del negocio

La finalidad del modelo de negocio es describir cada proceso del negocio, especificando sus datos, actividades (tareas), roles (agentes) y reglas de negocio (pautas). Se utilizó el modelo de negocio para representar los procesos de gestión en la Unidad de Sangrado de la institución hospitalaria. Todos estos procesos contienen datos, que en forma organizada, son utilizadas por distintas tareas, además se encuentran regidos por un conjunto de pautas que las coordinan, o sea las reglas de negocio, que determinan la estructura de la información y las políticas de la organización.

Para realizar el modelado del negocio por procesos se siguió la notación BPMN. Esta notación permite visualizar las actividades que se realizan, y ayuda a lograr un mejor entendimiento del flujo de trabajo existente

en la Unidad de Sangrado, lo cual es importante a la hora de informatizar el negocio. Mediante los elementos gráficos que la notación BPMN define, se pueden construir los diagramas de procesos. Para un mejor entendimiento se describe seguidamente el aporte de los elementos gráficos empleados:

- Los eventos de inicio y fin, indican una acción lógica en el flujo del proceso. Las bifurcaciones, determinan la ramificación del flujo a través de decisiones inclusivas o exclusivas.
- Los artefactos brindan la información de los documentos o registros que se requieren y se producen en cada actividad. Estos se relacionan mediante líneas discontinuas con la actividad que lo emplea. En la Unidad de Sangrado uno de los documentos más utilizados es la Historia Clínica del paciente, que se actualiza y consulta en varias actividades del proceso por parte del especialista.
- Los diagramas están compuestos por calles, que representan a los actores y trabajadores del negocio; donde el mayor cúmulo de actividades está en las calles de los trabajadores. En estas se muestran las actividades que realizan en orden lógico, respondiendo al proceso de negocio descrito. Los trabajadores y actores del negocio en este proceso son: Paciente, Médico de la Unidad de Sangrado y el Anestesiólogo.

La Figura 2.1 muestra el diagrama del proceso Recibir paciente. En el [Anexo 1](#) se refleja el diagrama del proceso Atender paciente. Existe una calle por cada rol participante en el escenario, que incluye las actividades que realiza dicho rol. Dentro de las actividades representadas en el diagrama se encuentran:

- Realizar examen físico
- Definir tratamiento
- Realizar intervención
- Realizar endoscopia
- Seguir conducta posterior del paciente
- Valorar estado del paciente

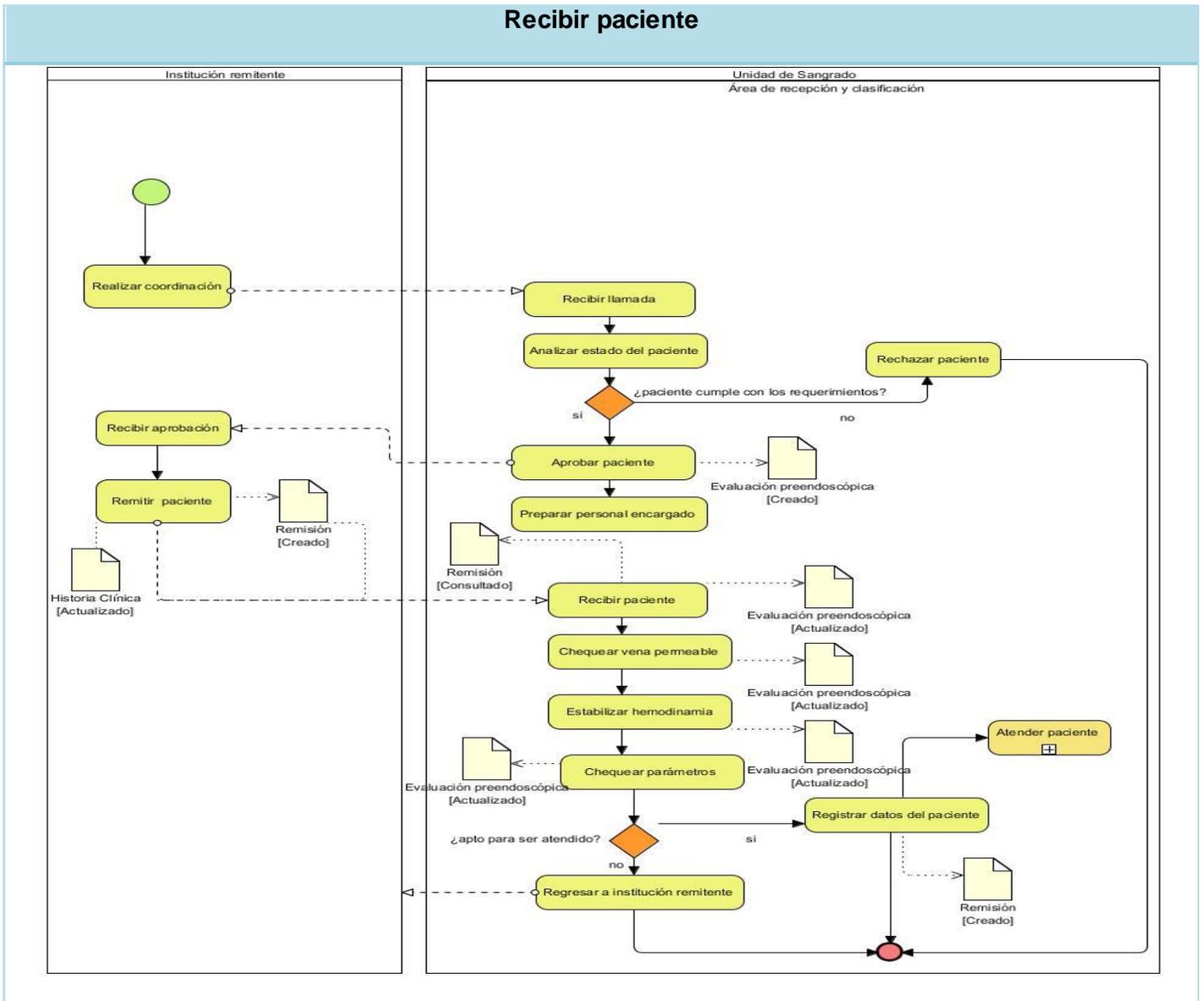


Figura 2.1: Diagrama del proceso de negocio: Recibir paciente (elaboración propia)

A continuación se describen los principales actores involucrados en el proceso Recibir paciente y Atender paciente.

Tabla 2.1 Trabajadores del negocio

Rol	Funciones
Paciente	Recibir atención médica.
Médico de la Unidad de Sangrado	Es el responsable de registrar los datos del paciente en el Registro de Pacientes Atendidos, se encarga de examinar, diagnosticar e indicar conducta a seguir a pacientes que solicitan asistencia médica.

2.4. Requisitos funcionales del sistema

A partir del análisis de los procesos de negocio y las actividades a informatizar, surgen las funcionalidades que el sistema debe incorporar para dar solución al problema identificado. Estas funcionalidades son especificadas mediante requisitos funcionales, o sea capacidades o condiciones que el sistema debe cumplir y que definen el comportamiento interno del software. Los requisitos funcionales se utilizan para describir los servicios que se espera que el sistema cumpla para satisfacer las necesidades del cliente. Además aporta una visión más detallada de lo que se va a implementar. A continuación se listan los requisitos funcionales del submódulo Unidad de Sangrado:

RF1 Crear evaluación pre-endoscópica.

RF2 Recibir paciente.

RF3 Atender paciente.

RF4 Crear hoja de emergencia de sangrado.

RF5 Actualizar hoja de emergencia de sangrado.

RF6 Crear evaluación trans-endoscópica.

RF7 Actualizar evaluación trans-endoscópica.

RF8 Generar reportes del total de pacientes atendidos.

RF9 Generar reportes del total de paciente según conducta a seguir.

RF10 Generar reportes de pacientes en observación según tiempo de estancia.

RF11 Generar reporte de pacientes egresados.

El sistema incorporará funcionalidades ya implementadas del módulo Emergencias, las cuales se describen en el [Anexo 4](#).

2.5. Requisitos no funcionales del sistema

Una vez determinado lo que el sistema debe hacer, se especifica cómo debe comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Los requisitos no funcionales responden a las condiciones que debe cumplir el sistema para satisfacer el contrato o especificación. Están regidos por las necesidades del usuario para poder resolver un problema o conseguir un beneficio determinado.

Los requisitos no funcionales son de gran utilidad para la aceptación del software, debido a que representan las ventajas más visibles al usuario y repercuten en el óptimo funcionamiento y mantenimiento del sistema. Para lograr que la solución satisfaga las necesidades del cliente y cuente con las propiedades fundamentales de un sistema como el tiempo de respuesta, la fiabilidad, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida, y la representación de datos que se utilizan en las interfaces, se realizó la especificación de los requisitos no funcionales.

A continuación se especifican los requisitos no funcionales que responden a cada una de las condiciones que deberá cumplir la solución propuesta.

Usabilidad

El submódulo debe cumplir con los principios fundamentales de usabilidad. Además deberá ser diseñado de manera que los usuarios adquieran las habilidades necesarias para lograr realizar las consultas y realizar un informe integral del comportamiento del paciente en un tiempo reducido. El sistema debiera ser admitido en un plazo de 30 días para el tipo de usuario al que está destinado.

Seguridad

El sistema debe implementar mecanismos de seguridad y control a nivel de usuario, que garanticen el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Se registrarán todas las acciones que se realizan por cada usuario en todo momento. Además proporcionará un registro de actividades

de cada usuario en el sistema. La información que se haya ingresado en el sistema nunca podrá ser eliminada físicamente de la base de datos.

Rendimiento

El sistema debe minimizar el volumen de datos en las peticiones y además optimizar el uso de recursos críticos como la memoria RAM. Se propone que se respeten las buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual de Java como la creación de objetos.

Soporte

Se debe permitir la administración de usuarios, el otorgamiento de privilegios y roles, la asignación de perfiles y la activación de permisos.

Software

El servidor debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma Java (JVM, JBoss AS y PostgreSQL). Los clientes deberán disponer de un navegador web que debe ser Firefox 2.0 o superior.

Hardware

Estaciones de trabajo

En la solución se incluyen estaciones de trabajo para las consultas del HIS, las que necesitan capacidad de hardware, soporte un sistema operativo, cuente con un navegador actualizado y siga los estándares web (se recomienda Internet Explorer 7 o superior o Firefox 2.x o superior).

Servidores

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y permanencia de la información y las aplicaciones bajo esquemas seguros y confiables. Los requerimientos mínimos con los que contará el submódulo son los siguientes:

Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo GNU/Linux.

Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo GNU/Linux.

Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 2 GB de memoria y 2x72GB de disco y sistema operativo GNU/Linux.

Restricciones de diseño

La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se debe manejar 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 de este.

Interfaz de usuario

Las ventanas del sistema que utilizará el usuario deberán contener bien estructurados los datos, además de permitir la interpretación correcta de la información. La interfaz deberá contar con accesos directos y menús desplegables que faciliten y aceleren su utilización. La entrada de datos incorrecta será detectada e informada al usuario.

Modelo de casos de uso del sistema

Los requisitos o capacidades internas del sistema a desarrollar han sido expresados como casos de uso, teniendo en cuenta que estos son adecuados para describir 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 (casos de uso), el entorno del sistema (actores), y las relaciones entre ellos. Aunque la parte más visible de dicho modelo son los diagramas de casos de uso, suele ir acompañado de una especificación textual de cada caso de uso. En la Tabla 2.3 se describen los actores del sistema identificados.

Tabla 2.3 Actores del sistema

Actor	Descripción
Médico de la Unidad de Sangrado	Este rol es el encargado de registrar la atención del paciente en el sistema. Aprueba la recepción del paciente, crea y actualiza la Hoja de emergencias de sangrado y la Evaluación trans-endoscópica donde se registran los principales datos asociados a la atención del paciente.
Personal de la Unidad de Sangrado	Es la persona autorizada para realizar la admisión del paciente y coordinar la remisión del mismo. Este rol puede desempeñarlo un enfermero(a) o un médico de la Unidad de Sangrado.

Los actores pueden representar roles jugados por usuarios humanos, hardware externo, u otros sujetos. Un actor no necesariamente representa una entidad física específica, sino simplemente una faceta particular (es decir, un "rol") de alguna actividad que es relevante a la especificación de sus casos de uso asociados. Así, una única instancia física puede jugar el rol de muchos actores diferentes y, así mismo, un actor dado puede ser interpretado por múltiples instancias diferentes. (43)

En la Figura 2.2 se muestra el diagrama de actores correspondiente.

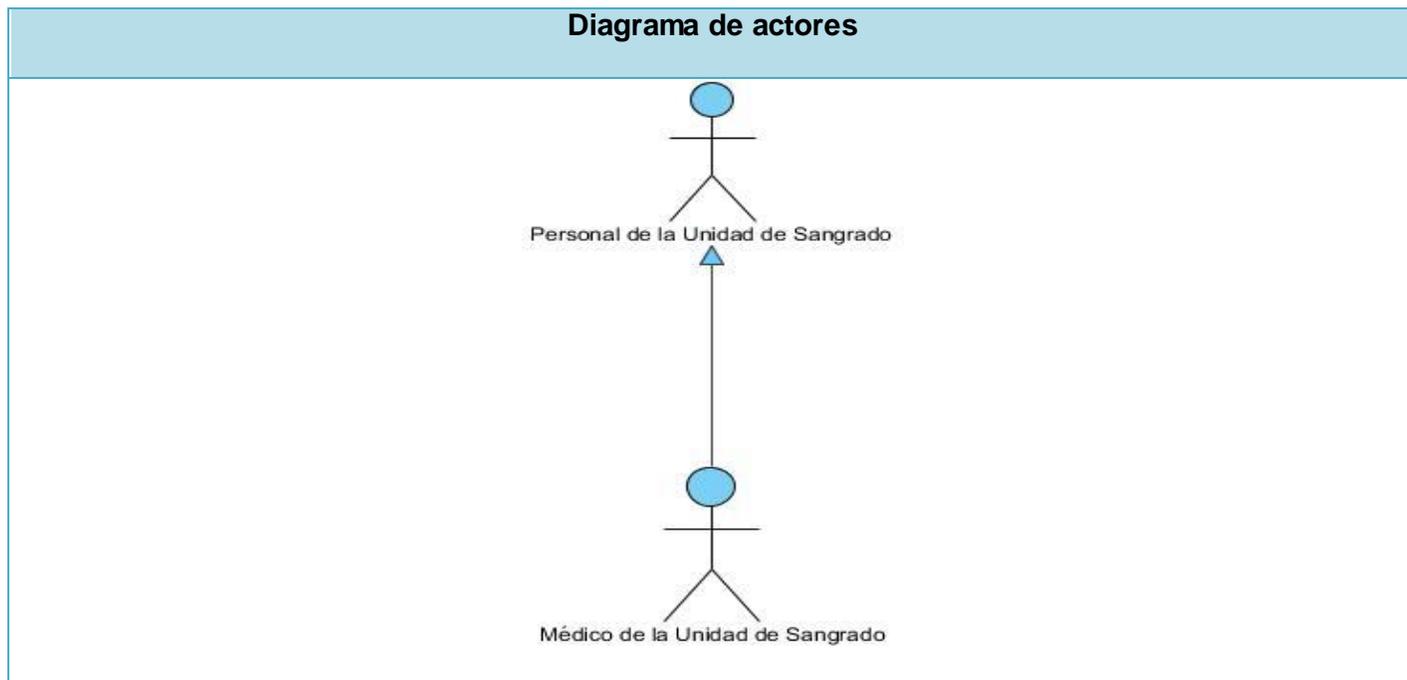


Figura 2.2 Diagrama de actores (elaboración propia)

El diagrama de casos de uso representa la forma en cómo un cliente (Actor) opera con el sistema en desarrollo. Además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). (44)

En la Figura 2.3 y 2.4 se muestran los diagramas de casos de uso correspondientes a los requisitos Recibir paciente y Atender paciente respectivamente. En el [Anexo 2](#) se muestran los diagramas de casos de uso de los requisitos Generar reportes y Atender paciente, este último contiene los casos de uso pertenecientes a otros módulos que son necesarios para complementar la atención del paciente.

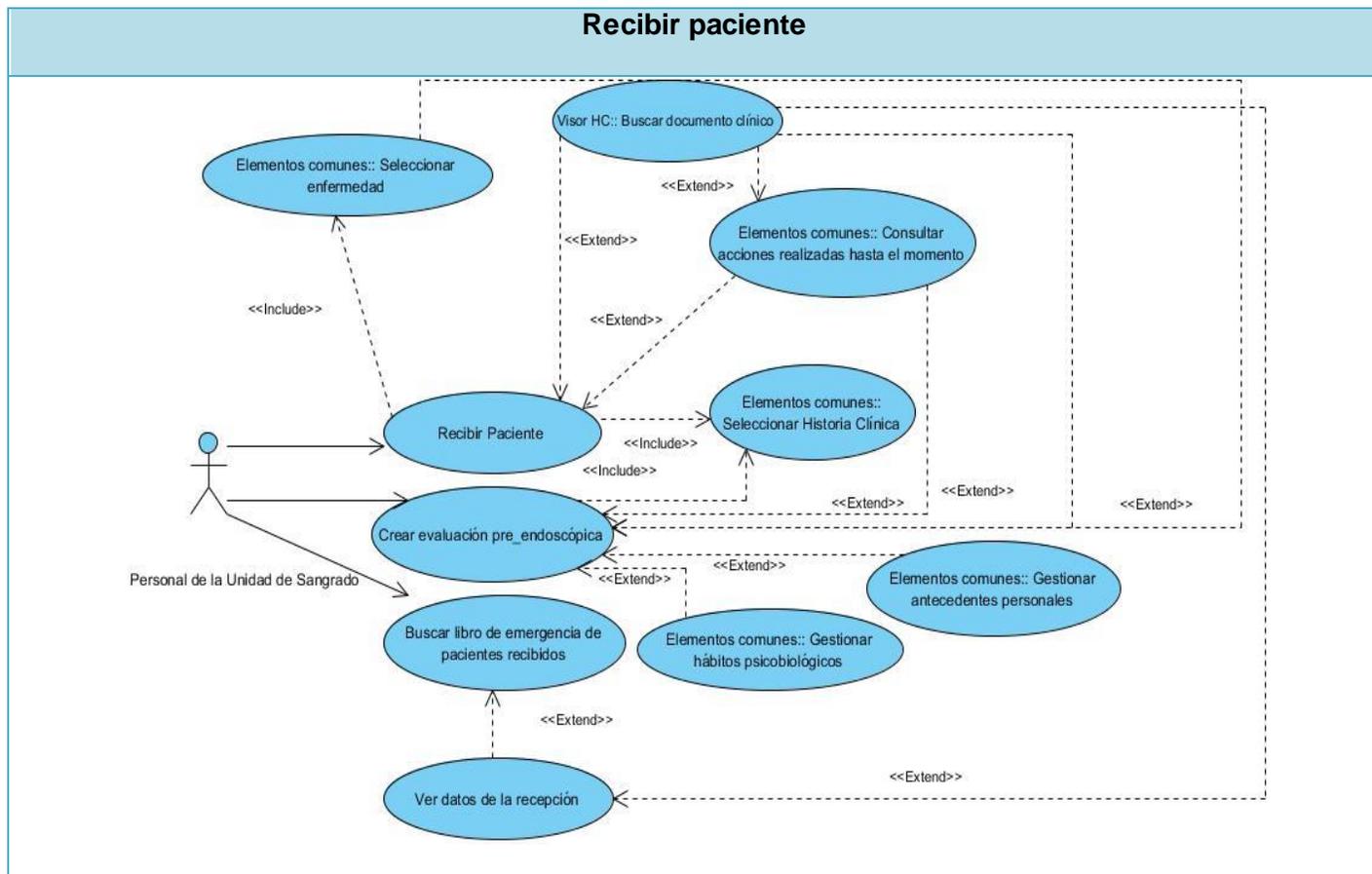


Figura 2.3 Diagrama de casos de uso del sistema: Recibir paciente (elaboración propia)

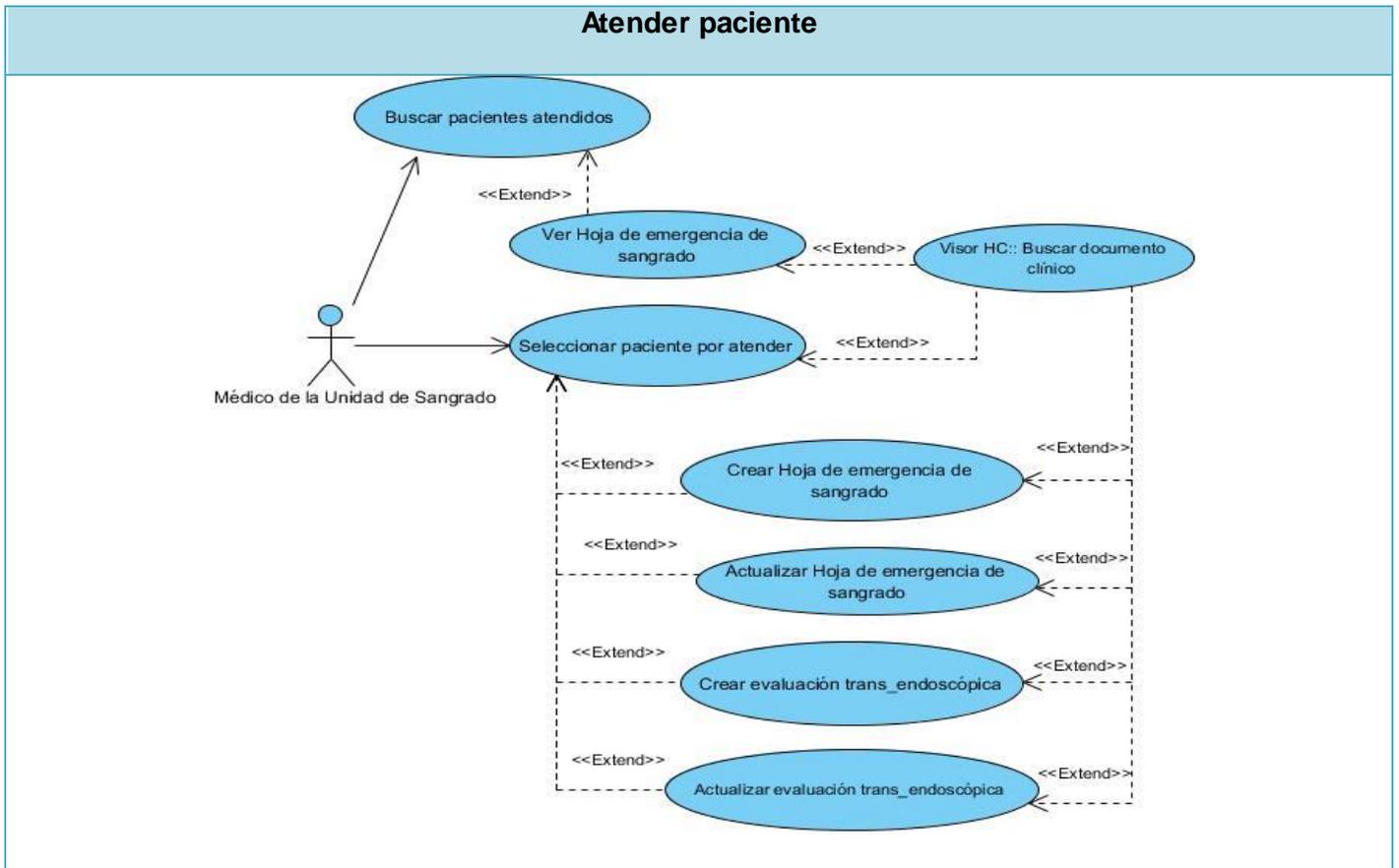


Figura 2.4 Diagrama de casos de uso del sistema: Atender paciente (elaboración propia)

A continuación se realiza la descripción textual de los casos de uso Recibir paciente y Atender paciente, Tablas 2.4 y 2.5, respectivamente.

Tabla 2.4 Descripción textual del caso de uso: Recibir paciente

Recibir paciente	
Actores:	Médico de la Unidad de Sangrado.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Recibir paciente, el sistema brinda la posibilidad de buscar la Historia Clínica del paciente, dado su cédula. Una vez encontrada el sistema permite introducir los datos para recibir al paciente. El sistema crea la Recepción del paciente y la asocia a la Historia Clínica, el caso de uso termina.
Precondiciones:	Ha sido creada la Historia Clínica Electrónica.
Postcondiciones:	Se creó la Recepción del paciente.
Referencias:	RF2

Tabla 2.5 Descripción textual del caso de uso: Atender paciente

Atender paciente	
Actores:	Médico de la Unidad de Sangrado
Resumen:	El caso de uso inicia cuando el actor accede a la opción Listar pacientes por atender, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el Paciente por atender deseado, el actor introduce los datos que considera como criterios para buscar el Paciente por atender, el sistema busca y muestra los Pacientes por atender que cumplen con los criterios de búsqueda, el actor selecciona Paciente por atender deseado, el sistema carga en la vista anterior los datos del Paciente por atender seleccionado y permite realizar varias opciones, el caso de uso termina.
Precondiciones:	El paciente fue recibido.
Postcondiciones:	Se seleccionó un paciente por atender.

Referencias:

RF3

Conclusiones parciales

- Se identificaron como procesos principales asociados al área de la Unidad de Sangrado, los procesos Recibir paciente y Atender paciente.
- El estudio del dominio de aplicación junto a las demandas específicas del cliente, permitió identificar 11 requisitos funcionales y 8 no funcionales, para garantizar el correcto funcionamiento del submódulo.
- Con el apoyo de diagramas y fichas se elaboró una representación detallada de cada proceso, lo que posibilitó un mejor entendimiento del problema y una rápida identificación de las funcionalidades.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SUBMÓDULO UNIDAD DE SANGRADO

En este capítulo se realiza una descripción del diseño del sistema. Se presentan los principales patrones de diseño utilizados y los diagramas de clases del diseño que forman parte del Modelo de diseño. Se brinda una descripción de las clases.

3.1. Descripción de la arquitectura, fundamentación

La arquitectura del software es el diseño de más alto nivel de la estructura de un sistema. Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. (45)

La solución propuesta presenta una arquitectura cliente-servidor, y para su implementación se utiliza el patrón de diseño MVC, expuesto en el Capítulo 1. Este separa los elementos de las capas presentación, negocio y acceso a datos para lograr que cada una se comunique con sus adyacentes, permitiendo que los cambios de una capa puedan realizarse sin afectar a todo el sistema. El uso del framework JBoss Seam permite contar con la implementación de dicho patrón, brindándole al submódulo la posibilidad de tener una separación clara entre cómo se muestra la información al usuario, cómo se manejan las acciones que el usuario desea hacer sobre el sistema y cómo se realizan estas acciones modificando y validando la información.

Los llamados patrones GRASP (patrones para asignar responsabilidades) tuvieron una importante utilidad en el diseño realizado. A continuación se describe la utilización de cada uno de ellos:

- Experto: se utiliza en la implementación de las clases *RecibirPacienteControlador_sangrado* pues la misma poseen la información necesaria para realizar el cálculo de la escala de Blatchford.
- Creador: es utilizado en la implementación de la clase *CrearEvalPreEndoscControlador_sangrado* pues contiene objetos de la Evaluación pre-endoscópica del paciente y registra estos objetos en la base de datos.
- Controlador: se utiliza en la implementación de las clases *CrearEvalTransEndoscControlador_sangrado* pues constituye el controlador de las acciones y eventos del sistema para el caso de uso Crear evaluación trans-endoscópica.

3.2. Estrategias de integración

La práctica de reutilización de código fuente trae ventajas para toda aplicación que se encuentre en desarrollo y permite principalmente reducir tiempo, minimizar las redundancias y aprovechar el trabajo anterior. La forma más eficiente de la reutilización de código es la creación de componentes reutilizables para evitar la duplicidad del mismo. (46)

Para el desarrollo del sistema HIS se utilizan varios componentes de forma común en todos los módulos, con el objetivo de lograr uniformidad en el desarrollo y mejorar la calidad del trabajo. Entre los componentes que se reutilizan están: la clase *ActiveModule* para conocer en qué módulo y entidad se encuentra el usuario que está utilizando el sistema, la clase *Bitácora* para el control de las trazas de todas las acciones que se realizan con la aplicación; la clase *Usuario* para saber qué usuario está trabajando con la aplicación en tiempo real.

Los procesos de la Unidad de Sangrado abarcan diferentes áreas dentro de la institución hospitalaria, por esto se hace necesario integrar las funcionalidades desarrolladas con otros módulos del HIS, englobando todo el proceso de atención al paciente. El submódulo establece dependencias tanto unidireccionales como bidireccionales con algunos de los módulos que componen el sistema HIS.

La relación que existe entre el submódulo Unidad de Sangrado y Laboratorio es bidireccional, desde la Unidad de Sangrado se emiten las solicitudes de análisis de laboratorio. Una vez que estas se procesan en el laboratorio, desde la Unidad de Sangrado se pueden consultar los resultados de los análisis. Con el módulo de Banco de sangre la relación es similar a con Laboratorio, desde la Unidad de Sangrado se puede emitir una solicitud de examen, siempre y cuando en el centro de salud o entidad de salud se cuente con un Banco de sangre. Una vez que el Banco de sangre procese la solicitud, desde la Unidad de Sangrado se puede consultar el resultado del mismo.

En el caso de Admisión la relación si es bidireccional, en este se crean las Historias Clínicas a los pacientes que aún no han sido registrados en el sistema, y se visualizan en el submódulo Unidad de Sangrado. Una vez atendido el paciente, se emite la orden de admisión en caso de ingreso. La relación entre la Unidad de Sangrado y Farmacia es unidireccional, desde Unidad de Sangrado se emiten las indicaciones médicas con presentaciones de medicamento, las cuales tienen asociadas recetas que llegan al módulo de Farmacia, que es quien se encarga de despacharlos. La relación con el módulo de Bloque Quirúrgico también es unidireccional, desde la Unidad de Sangrado se crea una la solicitud de intervención quirúrgica que es

procesada en el módulo Bloque Quirúrgico. Con Enfermería la relación es unidireccional, una vez que se emiten indicaciones médicas con tratamientos médicos y presentaciones de medicamento para cumplimiento de la enfermera, estas llegan al módulo de Enfermería que se encarga de darles cumplimiento. En la Figura 3.1 se muestran las entradas y salidas del submódulo desarrollado con los demás módulos del HIS.

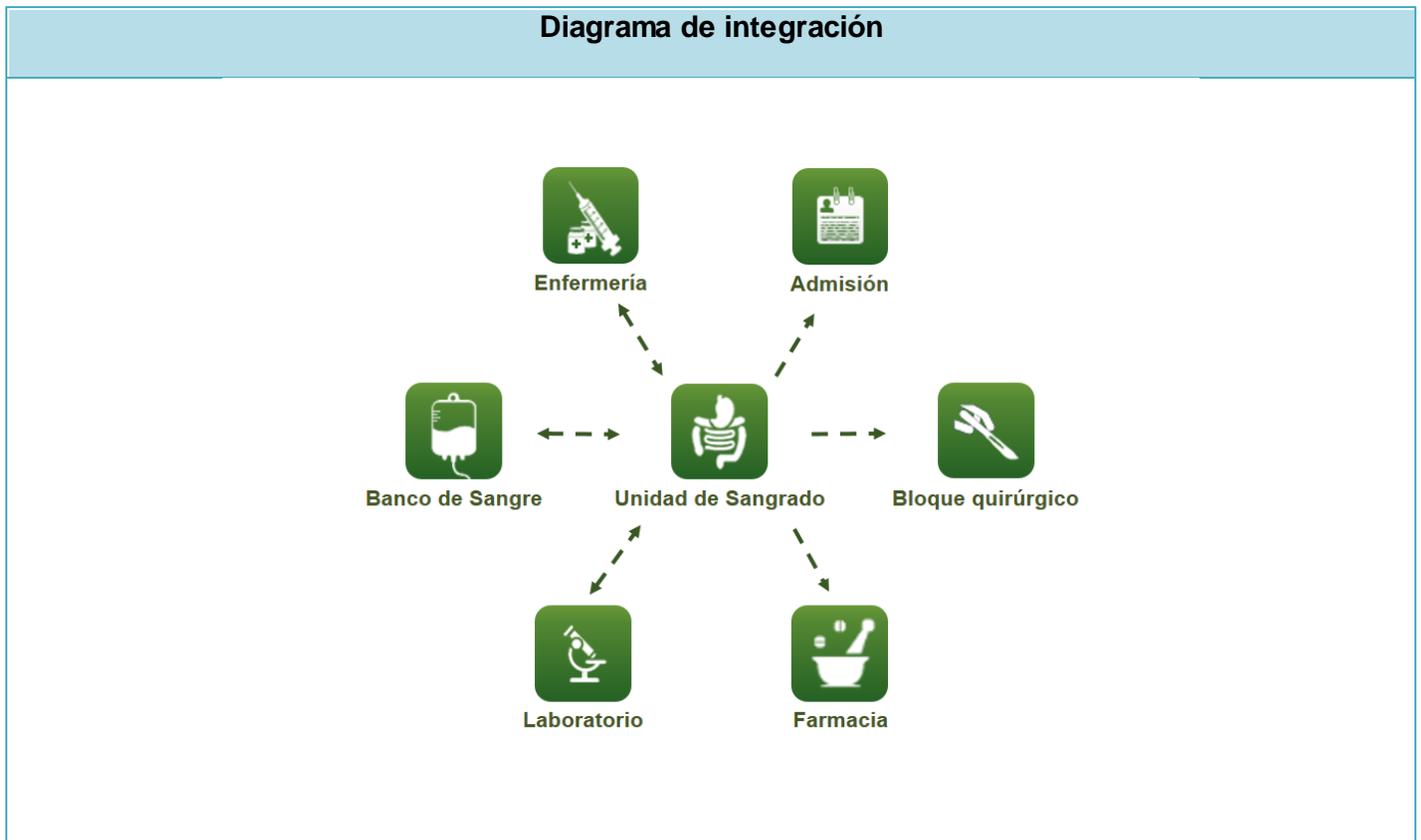


Figura 3.1 Diagrama de integración (elaboración propia)

3.3. Modelo de diseño

Mediante el Modelo de diseño se hace un refinamiento del proceso de análisis anteriormente realizado. Para ello se tienen en cuenta los requisitos no funcionales del sistema ya que el principal propósito del modelado del diseño es crear un plano del modelo de implementación. También se define la arquitectura del sistema. Los casos de uso son realizados por las clases del diseño y sus objetos, a partir de los cuales se forma el diagrama de clases del diseño. Los términos empleados en el Modelo de diseño son: clase de diseño, Diagramas de clases de diseño, Diagramas de interacción y Diagrama de secuencia. (47)

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. Se emplea el criterio de empaquetamiento por proceso, siguiendo la estructura de procesos definidos en el sistema.

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, donde cada una de ellas contiene un Diagrama de clases del diseño y los respectivos Diagramas de secuencia.

El paquete *repositorio* contiene dos subpaquetes, uno para las entidades y otro para las sesiones. En el *subpaquete de entidades* se encuentran las clases autogeneradas definidas en el diseño de acuerdo a 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 (permite la generación de objetos java desde la base de datos). 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 en la Figura 3.2 se muestra el Diagrama de paquetes correspondiente al submódulo Unidad de Sangrado.

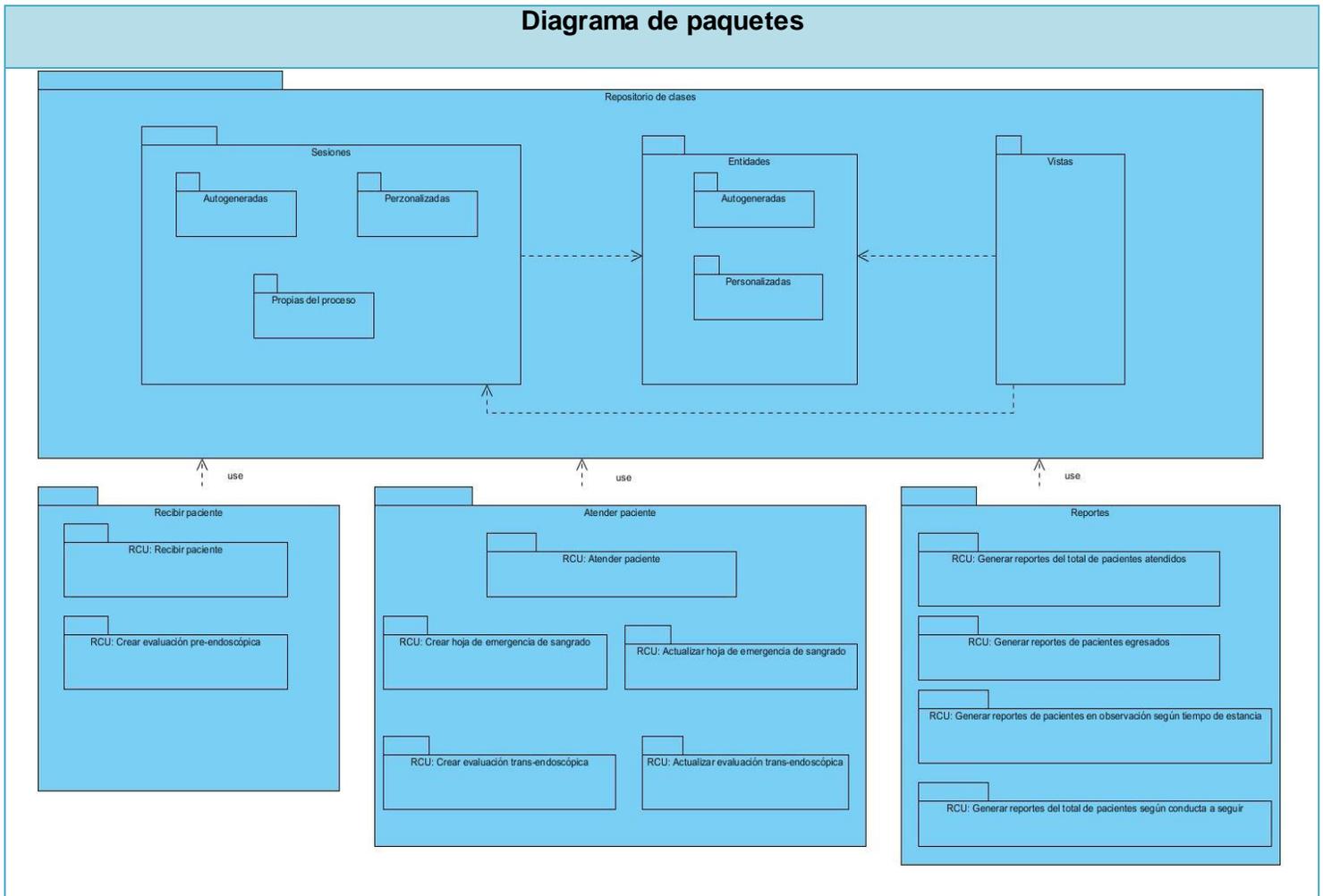


Figura 3.2 Diagrama de paquetes (elaboración propia)

Como se evidencia en el Diagrama de paquetes, el sistema está dividido en dos procesos: Recibir paciente y Atender paciente. En la Figura 3.3 se muestra el diagrama de clases del diseño: Recibir paciente. En el [Anexo 3](#) se reflejan los diagramas de clases del diseño correspondiente a los requisitos Crear evaluación pre-endoscópica, Actualizar hoja de emergencia de sangrado y Crear evaluación trans-endoscópica.

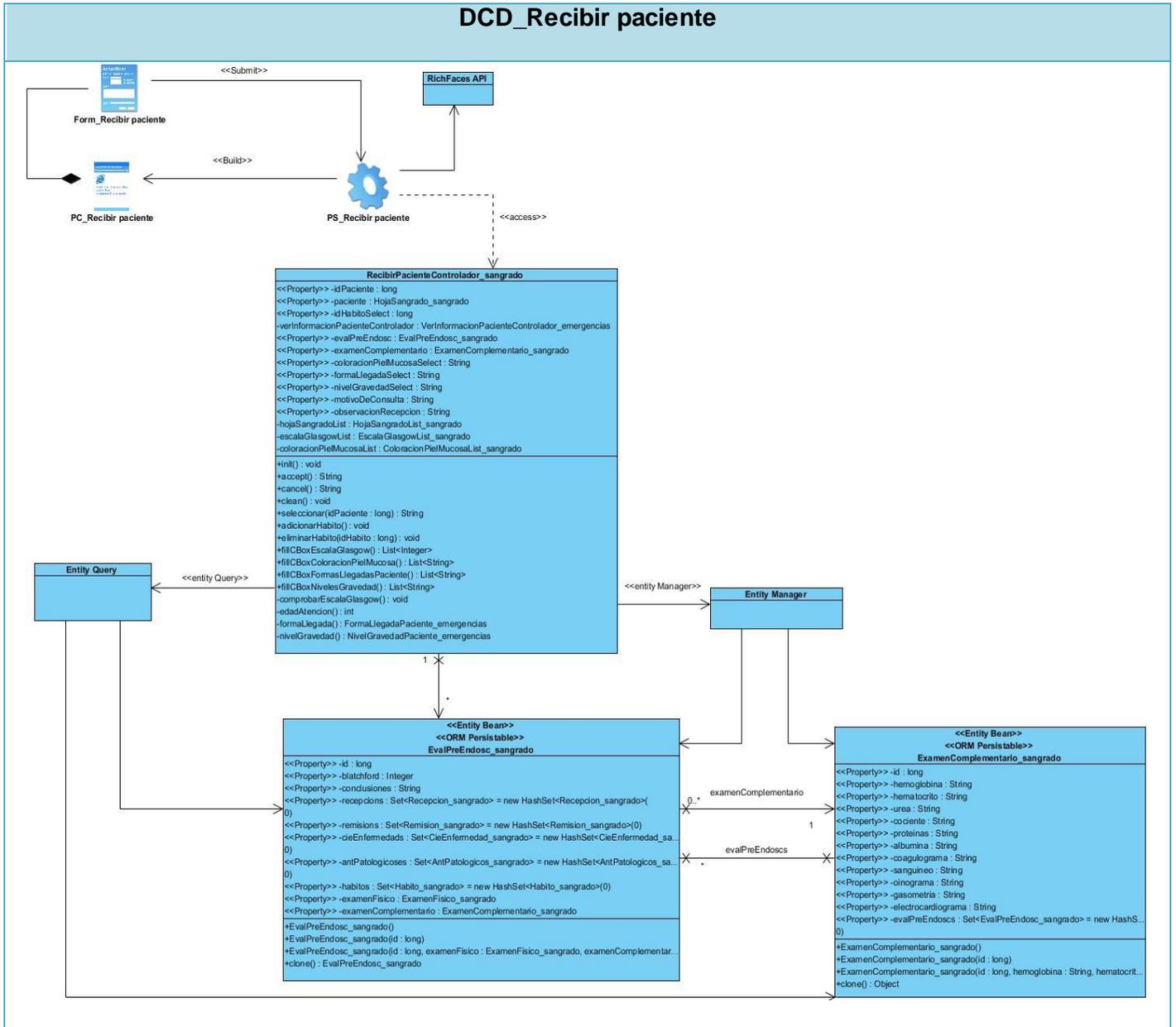


Figura 3.3 Diagrama de clases del diseño: Recibir paciente (elaboración propia)

Las clases del diseño están agrupadas en:

Páginas servidoras: están compuestas por componentes Facelets, RichFaces, JSF, Seam UI, así como código HTML. Todo este código será ejecutado en el servidor web, generando páginas clientes que pueden ser representadas por los navegadores web.

Páginas clientes: están compuestas por código HTML, CSS, JavaScript. Son interpretadas por los navegadores web presentándole al usuario la interfaz con la que puede interactuar con el sistema.

Formularios: un formulario HTML es una sección de un documento enmarcado entre tags `<form>` y que puede contener elementos especiales llamados controles (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, entre otros.), y rótulos (labels) en esos controles. Los usuarios normalmente "completan" un formulario modificando sus controles (introduciendo texto, seleccionando objetos de un menú, etc.), y lo envían al servidor donde estos son procesados. Es una manera de obtener en el servidor información entrada por el usuario en el cliente.

Controladoras: las clases controladoras implementan la lógica del negocio que se está informatizando, generalmente cada una de estas se encargan de la implementación de un caso de uso o un proceso en dependencia de la complejidad de los mismos.

Seguidamente se explican algunas de las clases identificadas para su implementación describiéndose las responsabilidades de las páginas servidoras que responden a la Lógica de Negocio. De esta manera se tendrá una comprensión mayor del funcionamiento que tendrá el sistema en desarrollo.

Tabla 3.1 Descripción de la clase controladora: RecibirPacienteControlador_sangrado

Nombre: RecibirPacienteControlador_sangrado.	
Tipo de clase: Controladora.	
Atributo	Tipo
idPaciente	long
Paciente	HojaSangrado_sangrado
idHabitoSelect	long
coloracionPielMucosa	String
nivelGravedad	String
motivoDeConsulta	String
observacionRecepcion	String
Para cada responsabilidad:	
Nombre	int()
Descripción	Inicializa los campos y atributos requeridos.
Nombre	accept()
Descripción	Guarda las acciones realizadas por el usuario.
Nombre	cancel()
Descripción	Cancela las acciones realizadas por el usuario.
Nombre	clean()
Descripción	Limpia los campos y atributos utilizados.
Nombre	seleccionar(idPaciente: long)
Descripción	Selecciona un paciente de acuerdo a su id.

Nombre	adicionarHabito()
Descripción	Adiciona un hábito del paciente.
Nombre	eliminarHabito(idHabito: long)
Descripción	Elimina un hábito del paciente.
Nombre	edadAtención()
Descripción	Calcula la edad actual del paciente.
Nombre	formaLlegada()
Descripción	Selecciona la forma de llegada del paciente.
Nombre	NivelGravedad()
Descripción	Selecciona el nivel de gravedad del paciente.

Conclusiones parciales

- El análisis realizado permitió identificar las clases fundamentales que deben ser definidas para que el sistema funcione satisfactoriamente.
- Se especificaron los atributos y métodos que deben tener las mismas para tener una idea clara de lo que se debe implementar.
- Se obtuvo la realización de los casos de uso por procesos, donde se elaboraron los diagramas de clases correspondientes.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SUBMÓDULO UNIDAD DE SANGRADO

En el presente capítulo se introduce el flujo de trabajo de implementación que es consecuencia del flujo de Diseño. Se muestra el Diagrama de despliegue como parte del Modelo de implementación, que indica la distribución física de la solución implementada. Además se proporciona una detallada explicación de dicha solución.

4.1. Modelo de datos

El Modelo de datos describe la representación lógica y física de la información persistente manejada por el sistema. Puede ser inicialmente creado a través de la ingeniería inversa de un almacenamiento de datos persistentes que ya exista o puede ser inicialmente creado a partir de un conjunto de clases del diseño (entidades) persistentes en el Modelo de diseño. Es usado para definir el mapeo entre las clases del diseño y las estructuras de datos. (48)

Las entidades son objetos de los que el sistema necesita guardar información, las cuales están constituidas por uno o más atributos. Los atributos son las características asociadas a una entidad. Estos pueden ser clasificados en obligatorios, opcionales, claves foráneas y claves primarias (estas se dividen en simples y compuestas). Las relaciones, por su parte, muestran la forma en que dos entidades se asocian. Se representan mediante una línea que une a las dos entidades implicadas y manifiestan dos características principales: la cardinalidad y la obligatoriedad. (12)

- **Cardinalidad:** se refiere al número de ocurrencias de una entidad con respecto a la otra. La entidad de la que sale la relación tendrá tantas ocurrencias como indique el número asociado a la entidad a donde llega la relación señalando con una flecha el sentido de entrada, de no mostrarse el número de la cardinalidad se asume que la ocurrencia es de solamente una vez.
- **Obligatoriedad:** determina que ante la existencia de una entidad puede haber ocurrencias de otras relacionadas con esta. Si la ocurrencia es obligatoria, se representa mediante una línea continua, en caso contrario, se realiza a través de una línea discontinua.

La descripción de los componentes del Modelo de datos permite entender el diagrama que a continuación se presenta:

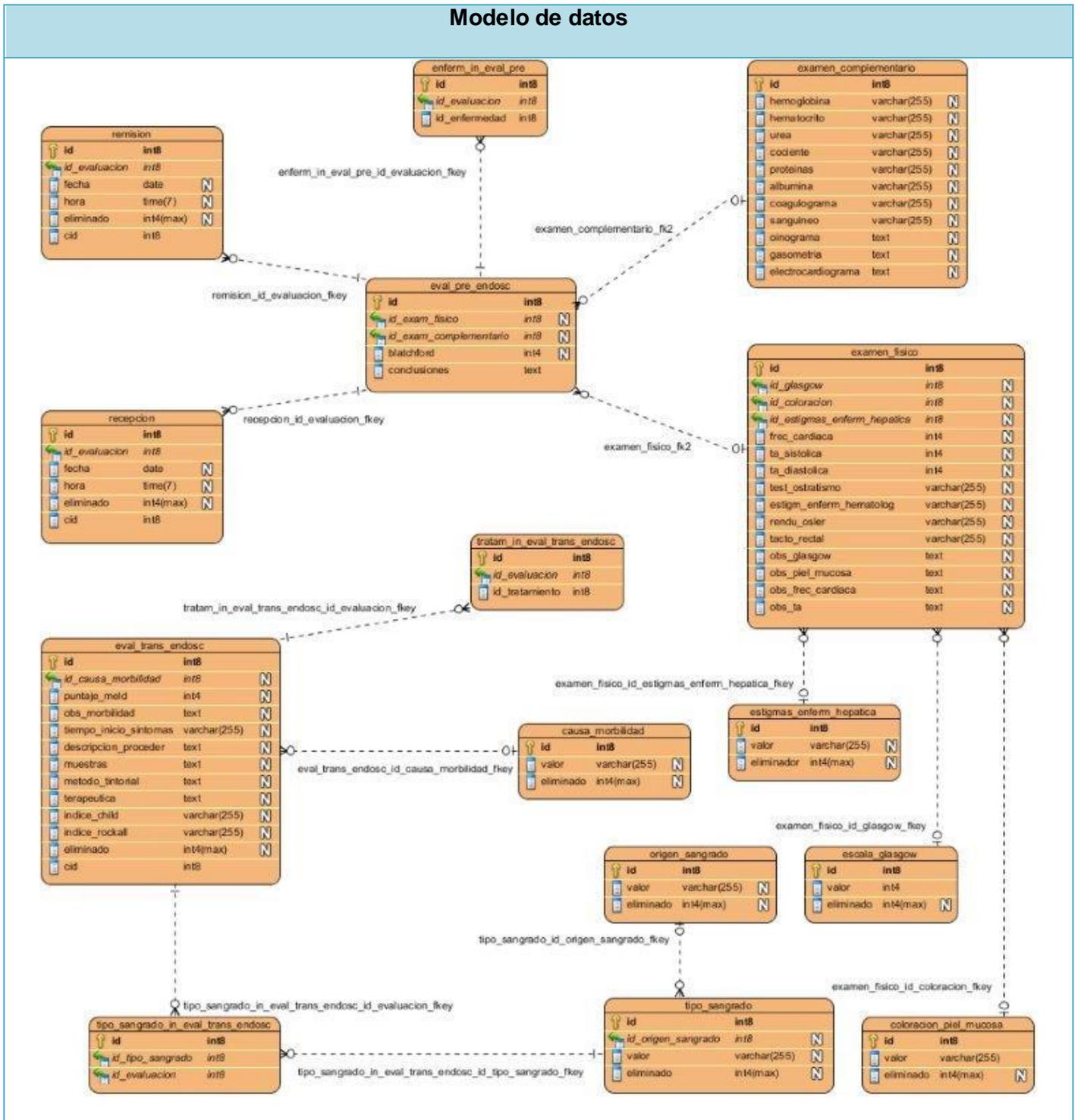


Figura 4.1 Modelo de datos (elaboración propia)

4.1.1. Descripción de las tablas de la base de datos

A continuación se describen las entidades más representativas del Modelo de datos de la solución propuesta. Los siguientes atributos son comunes a todas las entidades ya que fueron agregados con el objetivo de facilitar la implementación de algunas funcionalidades del sistema.

Tabla 4.1 Descripción de los atributos comunes dentro de todas las entidades

Atributos comunes en todas las entidades		
Atributos	Tipo	Descripción
id	bigint	Id necesario en cada entidad para las referencias en las relaciones entre tablas.
version	integer	Indica con qué versión de la entidad se está trabajando. Es usado para garantizar que se está trabajando con la versión de la entidad más actualizada que existe en la base de datos.
eliminado	boolean	Permite la eliminación lógica con que cuenta el sistema, cuando está en verdadero indica que la entidad está eliminada.
cid	long	Permite identificar quién realiza alguna acción sobre la entidad.

Tabla 4.2 Descripción de la tabla: eval_pre_endosc

eval_pre_endosc		
Atributo	Tipo	Descripción
id	bigint	Identificador
id_exam_fisico	bigint	Identificador del examen físico, llave foránea.
id_exam_complementario	bigint	Identificador del examen complementario, llave foránea.
blatchford	integer	Valor de la escala de Blachford.
conclusiones	text	Conclusiones

Tabla 4.3 Descripción de la tabla: examen_fisico

examen_fisico		
Atributo	Tipo	Descripción
id	bigint	Identificador
frec_cardiaca	integer	Frecuencia cardiaca
ta_sistolica	integer	Tensión arterial sistólica
ta_diastolica	integer	Tensión arterial diastólica
test_ostatismo	boolean	Si se le realizo test de Ortostatismo
estigm_enferm_hematolog	boolean	Si presenta estigmas de la enfermedad hematológica
rendu_osler	boolean	Si presenta angomas dérmicos en el caso de Rendu-Osler

tacto_rectal	boolean	Si presenta tacto rectal positivo de sangrado
obs_glasgow	text	Observaciones sobre el estado de conciencia
obs_piel_mucosa	text	Observaciones sobre la coloración de piel y mucosas
obs_frec_cardiaca	text	Observaciones sobre la frecuencia cardiaca
obs_ta	text	Observaciones sobre la tensión arterial
id_glasgow	bigint	Identificador de la escala de Glasgow, llave foránea.
id_coloracion	bigint	Identificador de la coloración de piel y mucosa, llave foránea.
id_estigmas_enferm_hepatica	bigint	Identificador de los estigmas de la enfermedad hepática, llave foránea.

4.2. Modelo de implementación

El Modelo de implementación describe cómo los elementos del diseño se implementan en componentes. Entre los componentes se puede encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe entre los paquetes y clases del Modelo de diseño y los subsistemas y componentes físicos. El propósito del Modelo de implementación es definir la organización del código, planificar las integraciones de sistema necesarias en cada iteración e implementar las clases y subsistemas encontrados durante el diseño. (12)

4.2.1. Diagramas de componentes

Los Diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización como el código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente. Estos tienen relaciones de traza con los elementos del Modelo de diseño. (49)

De los estereotipos estándar que se aplican a los componentes según el lenguaje de modelado UML se emplean: library y file, los cuales representan, una biblioteca de objetos estática o dinámica y un documento que contiene código fuente, respectivamente.

Las relaciones de dependencia se utilizan en los Diagramas de componentes para indicar que un componente se refiere a los servicios ofrecidos por otro componente. Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación. Estos paquetes son estereotipados como <<subsistemas>>.

Cada subsistema puede contener componentes y otros subsistemas. La descomposición en subsistemas no es necesariamente una descomposición funcional. La relación entre paquetes y clases en el nivel lógico es el que existe entre subsistemas y componentes en el nivel físico. (49)

A continuación se expone el Diagrama de componente asociado a varios de los subsistemas de implementación identificados. El mismo es similar al del módulo Emergencias del Sistema de Información Hospitalaria. Los componentes que interactúan en el submódulo Unidad de Sangrado son los mismos que los del módulo Emergencias, por lo que se hizo necesario hacer referencia al mismo diagrama. Siguiendo la arquitectura en capas, la estructuración en subsistemas de implementación es la siguiente:

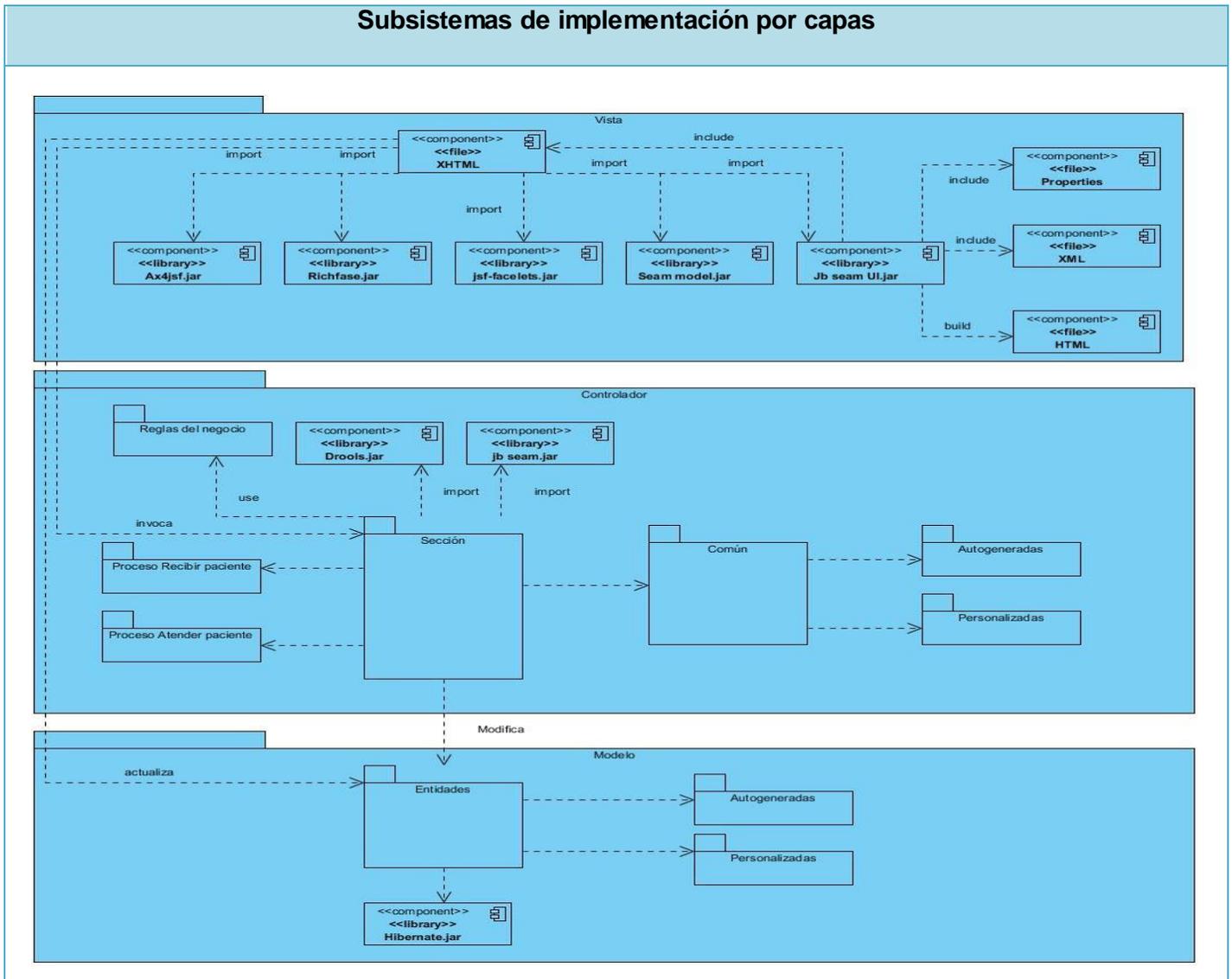


Figura 4.2 Subsistemas de implementación por capas (elaboración propia)

4.2.2. Diagrama de despliegue

El Diagrama de despliegue muestra las relaciones físicas entre los componentes de hardware y software en el sistema final. Es un grafo de nodos unidos por conexiones de comunicación. Estos muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. (50)

La arquitectura en tiempo de ejecución de la solución propuesta se modela haciendo uso de tres elementos de hardware: computadora cliente, servidor de aplicaciones y servidor de base de datos. Por otra parte la comunicación entre los nodos se rige por los protocolos HTTP para la agrupación entre la computadora cliente y el servidor de aplicaciones y TCP/IP para la conexión entre los servidores.

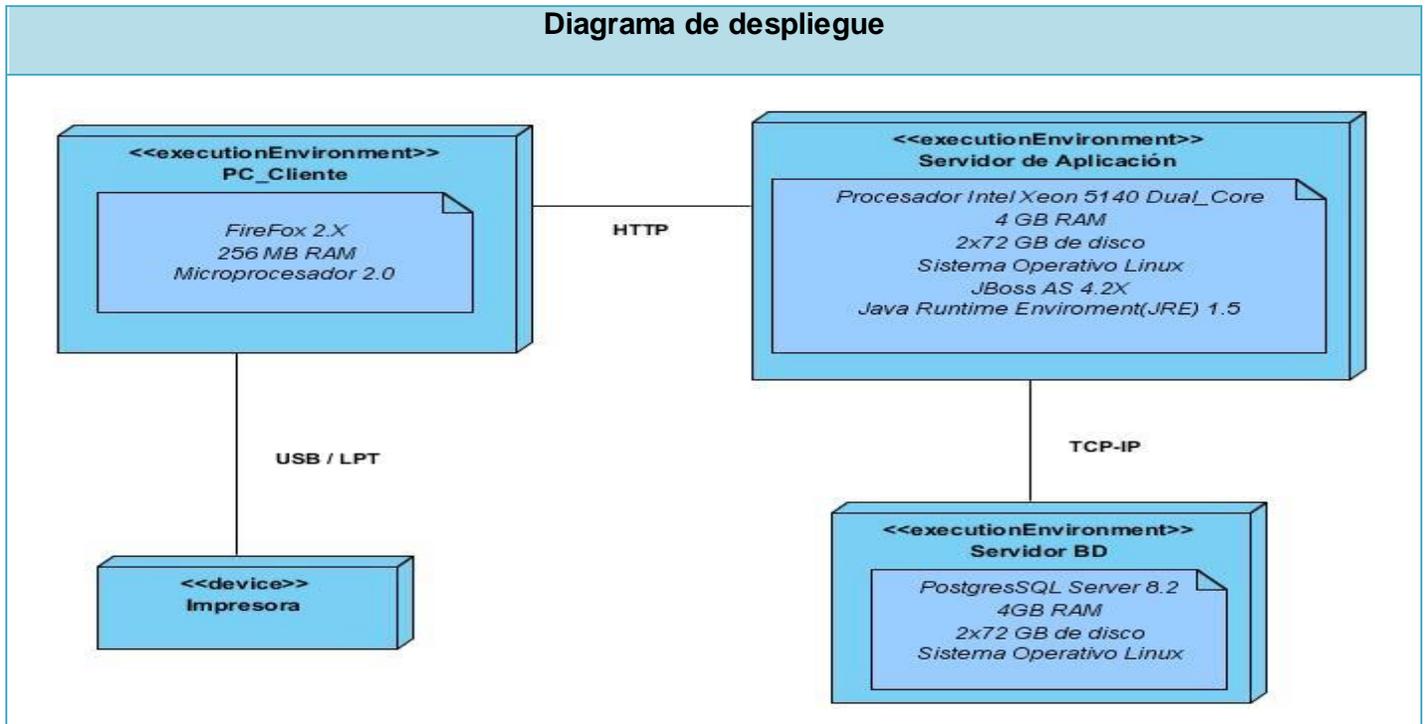


Figura 4.3 Diagrama de despliegue (elaboración propia)

4.3. Tratamiento de errores

Durante el tiempo de ejecución de un sistema pueden fracasar diferentes rutinas; es esto a lo que comúnmente se le llama excepción. Mediante el tratamiento de excepciones se restaura un estado en el que la rutina pueda seguir la ejecución, lo que permite obtener un sistema más robusto y fiable. (51)

En el sistema propuesto, el control de las excepciones se lleva a cabo a toda porción de código, donde pueda surgir alguna situación inesperada, especialmente donde se ejecutan sentencias que manipulan los datos que viajan desde y hacia la base de datos. También se controlan los errores que pueden surgir en la validación de datos provenientes de la interfaz de usuario, puesto que encierran una lógica compleja en cierta medida.

Para el manejo de las excepciones o errores, en las clases controladoras de procesos, se utiliza el bloque **try** para detectar cuando ocurra algún fallo y mediante el **catch** se manejarán dichas excepciones, mediante mensajes que se muestran en la interfaz de usuario, por las facilidades que brinda el FacesMessages, componente del framework Seam.

4.4. Sistema de seguridad

Un sistema informático se considera seguro cuando garantiza la confidencialidad, integridad y disponibilidad de sus datos. (52) Debido a la importancia que poseen el proceso de atención a pacientes, a la necesidad de que estos sean realizados con la calidad requerida y a que el acceso a la información derivada de este proceso solo sea por las personas que se encuentren autorizadas, se hace necesario que se tenga en cuenta una serie de requisitos de seguridad de modo que no pueda afectar la el funcionamiento del submódulo y la integridad de los datos.

Las funcionalidades del sistema encargadas de la seguridad son: iniciar y cerrar sesión de trabajo, registrar trazas y administrar seguridad. Para iniciar la sesión de trabajo un usuario debe acceder al sistema e insertar su nombre de usuario y contraseña. El sistema verifica que los datos introducidos sean válidos y dados los permisos de este usuario tendrá acceso a la funcionalidad deseada. Para terminar las tareas realizadas en el sistema, este permite cerrar sesión o salir de la funcionalidad activa.

El registro de trazas es vital en el sistema ya que permite archivar las acciones que realiza el usuario sobre este, que pueden ser: inicio o cierre de sesión, acceso a un módulo, modificación de un atributo de una entidad o cualquier otra operación. En cualquiera de los casos anteriores la aplicación registra una traza en la base de datos. Se hace necesario administrar los permisos que se asignan a los usuarios para la navegación en el sistema y ello se logra a través de la funcionalidad: administrar seguridad. El sistema brinda la posibilidad de asignar o denegar permiso a roles y usuarios en las funcionalidades de los módulos.

4.5. Estrategias de codificación

Los estándares de codificación son reglas que se aplican para lograr uniformidad en el código producido por un grupo de desarrollo de un sistema. Estos reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran, lo que permite obtener un código de alta calidad. (53)

Para la solución del problema de la investigación se utiliza el estándar de SUN Microsystems (Stanford University Network) para Java, mediante la Notación Camello, para denotar variables, parámetros y métodos.

Variables, parámetros y métodos

En esta notación, el identificador para las variables, los parámetros y los métodos se define escribiendo las palabras de la siguiente forma, la primera con minúsculas y a partir de la segunda palabra, en caso de existir, con letra inicial mayúscula, ejemplo: `int cantidadReal; public void aceptar()`.

A continuación se especifican algunas restricciones para la nomenclatura, basada en el estándar a utilizar:

Empleo de márgenes en el código

El empleo de márgenes será de dos espacios por bloque de código. Los inicios y cierres ({ }) de ámbito estarán alineados de la siguiente forma: la llave de apertura "{" aparece al final de la misma línea de la sentencia de declaración, la llave de cierre "}" empieza una nueva línea indentada para ajustarse a su sentencia de apertura correspondiente, excepto cuando no existen sentencias entre ambas, que debe aparecer inmediatamente después de la de apertura "{".

Comentarios, separadores, líneas, espacios en blanco y márgenes

Ubicación de comentarios: el comentario se realizará al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (tipos de datos, y objetivo del parámetro).

Líneas en blanco: se dejará una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

Espacios en blanco: se usarán espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: `producto = nomproducto`.

Conclusiones parciales

- Con el propósito de representar de forma lógica y física, la información persistente y las relaciones físicas entre los componentes de hardware y software, se obtuvieron los artefactos: Modelo de datos, Diagrama de componente y Diagrama de despliegue.
- Se obtuvo un submódulo acabado y funcional, el mismo se encuentra correctamente documentado y cumple con los estándares definidos en el proceso de codificación.

CONCLUSIONES

Con la realización del presente trabajo se cumplió el objetivo general propuesto, por lo que se concluye lo siguiente:

1. El submódulo desarrollado contribuye a la gestión de la información que se genera en la Unidad de Sangrado con las especificidades del Centro Nacional de Cirugía de Mínimo Acceso.
2. Se garantiza la evolución y usabilidad del submódulo Unidad de Sangrado a partir de la utilización de los estándares de denominación médica.

RECOMENDACIONES

Con el objetivo de enriquecer la solución propuesta se proponen las siguientes recomendaciones para el trabajo presentado:

- Desarrollar el reporte estadístico asociado a la cantidad de pacientes no recibidos en la Unidad de Sangrado.
- Desarrollar el reporte estadístico asociado a la cantidad de procedimientos endoscópicos realizados.
- Desarrollar las funciones necesarias para la creación e interpretación de documentos CDA (Clinical Document Architecture), lo cual permitirá la comunicación con otras aplicaciones que tengan incorporado dicho estándar.

REFERENCIAS BIBLIOGRÁFICAS

1. **Martínez, Anyonia María Moya.** *Nuevas tecnologías.* Granada : Depto Legal, noviembre 2009. ISSN 1988-6047.
2. **Puerto, Francisco J Fernández y Gatica, Florina Lara.** *Sistema de información Hospitalaria.* Mexico : s.n., 2005.
3. **Mansolo, Dr. Athos Sánchez.** *Registro Electrónico de Pacientes.* Ciudad de La Habana : Revista cubana de educación médica, febrero 2009, Vol. 4.
4. **S.L., Grupo Tecma Red.** eSmartCity.es. *Internet de las Cosas aplicado a la e-Salud para médicos y pacientes.* [En línea] 13 de febrero de 2015. [Citado el: 15 de mayo de 2015.] <https://www.esmartcity.es/articulos/internet-de-las-cosas-apliacado-a-la-e-salud-para-medicos-y-pacientes>.
5. **Rodríguez, Ania González.** El Centro Nacional de Cirugía de Mínimo Acceso: Unidad de Ciencia e Innovación Tecnológica de actual impacto y prometedor futuro. *COCO-CMCK "El Periódico del Aire"*. [En línea] 21 de enero de 2014. [Citado el: 3 de Diciembre de 2014.] <http://www.radiococo.icrt.cu/2014/01/21/el-centro-nacional-de-cirurgia-de-minimo-acceso-unidad-de-ciencia-e-innovacion-tecnologica-de-actual-impacto-y-prometedor-futuro/>.
6. **Quintanilla, Dr. Raúl A. Brizuela.** *Estudios de los procesos que se realizan en la Unidad de Sangrado del Centro Nacional de Cirugía de Mínimo Acceso.* La Habana, 27 de noviembre de 2014.
7. **Porras, J. L. Martínez.** *Hemorragia Digestiva Alta: Etiología y Procedimientos diagnósticos.* Universidad Autónoma, Madrid : s.n., 2009. ISSN.
8. **Menocal, Dr. Julio.** *Estudios de los procesos que se realizan en la Unidad de Sangrado del Centro Nacional de Cirugía de Mínimo Acceo.* La Habana, 27 de noviembre de 2014.
9. **Sinaloa_México, Servicios de Salud de.** SIGHO Sistema de Información para la Gerencia. [En línea] 2013. http://sigho.se_sgro.gob.mx/wp_content/uploads/MANUALES/SIGHO/Admision.pdf.
10. **Eireos, Lixandra Córdova y Campos Cosme, Anny.** *Generación de reportes estadísticos para el Módulo Admisión del Sistema de Información Hospitalaria als HIS.* La Habana : Ediciones pensando el futuro, 2013. ISBN: 978-959-7213-02-4.

11. **internacional, Salud.** MediSys(Medical Information System). [En línea] [Citado el: 13 de enero de 2015.] <http://www.tumblr.com/search/salud%20internacional>.
12. **Antelo, Lorena Alemán y Garcia Orduñes, Jua Manuel.** *Tesis de pregrado. Módulo Emergencias del Sistema de Información Hospitalaria alas HIS.* La Habana : s.n., 2009.
13. **Casanova, Dr. Fernando Suazo.** *Gestión. X-HIS en Oncosalud, cuando un software revoluciona la atención médica.* [En línea] 5 de junio de 2014. [Citado el: 2 de febrero de 2015.] <http://gestion.pe/tecnologia/x-his-oncosalud-cuando-software-revoluciona-atencion-medica-2099444>. ISBN.
14. —. **Gestión. X-HIS en Oncosalud, cuando un software revoluciona la atención médica.** [En línea] 5 de junio de 2014. [Citado el: 2 de febrero de 2015.] <http://gestion.pe/tecnologia/x-his-oncosalud-cuando-software-revoluciona-atencion-medica-2099444>. ISBN.
15. **Jiménez, Dr. Oscar Manuel Villa y Peña Villa, Marcos Jesús.** *ProGastro. Sistema automatizado para el registro y procedimientos de los procederes en gastroenterología.* La Habana : Ediciones pensando el futuro, 2013. ISBN: 978-959-7213-02-4.
16. **Wilson, Jose Manuel Ayala.** Blog sobre arquitectura, multitarea, programación y tecnología web y mobile. *Patrones de diseño.* [En línea] [Citado el: 4 de febrero de 2015.] <http://jmaw.blogspot.com/2011/04/h2-margin-bottom-0.html>.
17. **Lorenzo, Amaya Alvarez y Mora Maure, Mirelio.** *Tesis de pregrado. Desarrollo de funcionalidades para la especialidad de Psicología del módulo Consulta Externa del sistema als HIS.* La Habana : s.n., julio 2012.
18. **Java.** [En línea] [Citado el: 5 de febrero de 2015.] https://www.java.com/es/download/faq/whatis_java.xml.
19. **Franky, María Consuelo.** Java EE 5. [En línea] [Citado el: 7 de febrero de 2015.] http://www.acis.org.co/fileadmin/Conferencias/ConfConsueloFranky_Abr19.pdf.
20. **Lucifer, Prometeo.** Java Runtime Environment – JRE. [En línea] [Citado el: 7 de febrero de 2015.] <http://www.elleonplateadodeojosrojos.es/blog/java-runtime-environment-jre/>.
21. **Wilmer Jaramillo.** Software Libre de Venezuela 777, C.A. [En línea] 2006. [Citado el: 7 de febrero de 2015.] <http://wilmer.fedorapeople.org/files/presentations/JBoss.pdf>.
22. **Maldonado, Daniel.M. El CoDiGo K.** Arquitectura de programación en 3 capas. [En línea] [Citado el: 2015 de febrero de 2.] <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/>.

23. **Smart, Jhon Ferguson.** JSF Jumpstar. [En línea] 2007. [Citado el: 4 de febrero de 2015.] http://www.wakaleo.com/public_resources/jsf-jumpstarter.pdf.
24. **Jboss.org.** Community Documentation. Introducción. [En línea] 2008. [Citado el: 4 de febrero de 2015.] <http://www.jboss.org/fileaccess/default/members/jbossrichfaces/freezone/docs/devguide/en/html/Introduction.html>.
25. **w3schools.** AJAX Tutorial. [En línea] [Citado el: 6 de febrero de 2015.] <http://www.w3schools.com/ajax/default.asp>.
26. **Community., JBoss.** JBoss Ajax4jsf. Introducción. [En línea] 2007. [Citado el: 6 de febrero de 2015.] <http://www.jboss.org/jbossajax4jsf/docs/devguide/en/html/Introduction.html>.
27. **Hookom, Jacob.** JSF Central tm. Inside Facelets Part 1: An Introduction. [En línea] 2005. [Citado el: 6 de febrero de 2015.] http://www.jsfcentral.com/articles/facelets_1.html.
28. **Egíluz Pérez, Javier.** Librosweb.es. HTML y XHTML. Capítulo 1: Introducción. [En línea] [Citado el: 6 de febrero de 2015.] http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html.
29. **ignside.net.** Cascade Style Sheets. [En línea] 2007. [Citado el: 6 de febrero de 2015.] <http://www.ignside.net/man/css/index.php>.
30. **Application, Web.** Plataforma J2EE. JBoss Seam Framework. [En línea] 2008. [Citado el: 6 de febrero de 2015.] <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>.
31. **Alvarez, Cecilio.** Hibernate. JPA vs Hibernate. [En línea] 22 de julio de 2014. [Citado el: 7 de febrero de 2015.] <http://www.genbetadev.com/frameworks/jpa-vs-hibernate>.
32. **Hibernate.org.** Hibernate Tools for Eclipse and Ant. [En línea] 2010. [Citado el: 6 de febrero de 2015.] <http://www.hibernate.org/255.html>.
33. **Hennebrueder, Sebastian.** Laliluna. Java tutorials and development. *First EJB 3 Tutorial showing a session and entity beans with annotations and JBoss.* [En línea] [Citado el: 7 de febrero de 2015.] <http://www.laliluna.de/ejb-3-tutorial-jboss.html>.
34. **Ort, B.R.** Sun microsystems. The Java Persistence API - A Simpler Programming Model for Entity. [En línea] [Citado el: 7 de febrero de 2015.] <http://java.sun.com/developer/technicalArticles/J2EE/jpa/>.

35. **Itera**. Rational Unified Process. [En línea] 2008. [Citado el: 7 de febrero de 2015.] http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42.
36. **Mora, Francisco**. UML: Lenguaje Unificado de Modelado. [En línea] [Citado el: 7 de febrero de 2015.] <http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF>.
37. **consulting, Milestone**. El nuevo estándar de modelado de negocio llega por primera vez a México. [En línea] [Citado el: 8 de febrero de 2015.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
38. **Works, IBM developer**. Iniciándose en la plataforma Eclipse. [En línea] [Citado el: 8 de febrero de 2015.] <http://www.ibm.com/developerworks/ssa/library/os-ecov/>.
39. **Ottinger, Joseph**. TheServerSide.com. JBoss releases JBoss Tools, Eclipse Plugins including Exadel. [En línea] [Citado el: 8 de febrero de 2015.] http://www.theserverside.com/news/thread.tss?thread_id=45933.
40. **Vial, Ovalle**. Sistema de gestión de base de datos. [En línea] 2010. [Citado el: 8 de febrero de 2015.] <http://vialovalle.blogcindario.com/2007/01/00061-sistema-de-gestion-de-bases-de-datos.html>.
41. **tldp.org**. Manuales de Ayuda.com. Breve historia de PostgreSQL. [En línea] 2009. [Citado el: 8 de febrero de 2015.] <http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/breve-historia-de-postgresql-01831.html>.
42. **Manager, Free Download**. Paradigma visual para UML (Plataforma Java). Visual Paradigm for UML . [En línea] [Citado el: 8 de febrero de 2015.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataf.
43. **Casos de Uso**. [En línea] [Citado el: 22 de marzo de 2015.] <http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>.
44. **Fúster, Gonzalo Génova**. Secciones técnicas. *Naturaleza de las relaciones entre actores y casos de uso*. [En línea] [Citado el: 24 de marzo de 2015.] <http://www.ie.inf.uc3m.es/ggenova/pub-novatica2004.pdf>.
45. **Pressman, Roger S**. *Ingeniería de software, un enfoque práctico*.
46. **Sáez, Enrique Flores**. **Openlibra**. *Reutilización de Código Fuente entre Lenguajes de Programación*. [En línea] 2012. [Citado el: 24 de abril de 2015.] <http://www.etnassoft.com/biblioteca/reutilizacion-de-codigo-fuente-entre-lenguajes-de-programacion/>.

47. **UML., Diagramas.** Software y Aplicaciones Web. [En línea] 2012 de febrero de 14. [Citado el: 23 de marzo de 2015.] <http://www.jtentor.com.ar/post/UML-Diagramas.aspx>.
48. **Raúl García.** Modelo de datos. [En línea] [Citado el: 2 de abril de 2015.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos>.
49. **Modelo de Implementación:** Diagramas de Componentes y Despliegue. [En línea] [Citado el: 15 de mayo de 2015.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
50. **SPARX Systems.** *Diagrama de Despliegue UML 2.* [En línea] [Citado el: 25 de abril de 2015.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
51. **DavidMarco.es.** *Tratamiento de excepciones en Java.* [En línea] 16 de julio de 2012. [Citado el: 24 de abril de 2015.] <http://www.davidmarco.es/articulo/tratamiento-de-excepciones-en-java>.
52. **Seguridad Informática SMR.** *Tema 1- Seguridad Informática.* [En línea] 24 de abril de 2015. <http://seguridadinformaticasmr.wikispaces.com/TEMA+1-+SEGURIDAD+IFORM%C3%81TICA>.
53. **(s.f.), MSDN.** Revisiones de código y estándares de codificación. [En línea] [Citado el: 24 de abril de 2015.] [http://msdn.microsoft.com/es-es/library/aa291591\(VS.71\).aspx](http://msdn.microsoft.com/es-es/library/aa291591(VS.71).aspx).

BIBLIOGRAFÍA

- Alemán Antelo, Lorena y Garcia Orduñez, Juan Manuel. Tesis de pregrado. Módulo emergencias del sistema de información hospitalaria alas HIS. Ciudad Habana, Cuba: s.n., 2009.
- Bascón Pantoja, Ernesto. El patrón de diseño Modelo-Vista-Controlador (MVC). 2004. Desarrollo web. Arquitectura cliente-servidor. [En línea] 30 de Agosto de 2007. [Citado el: 22 de Diciembre de 2014.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
- Cruz Quispe, Victo Fabio, y otros. Diagrama de Componentes. 2012: s.n. Curso de Java. [En línea] [Citado el: 2 de enero de 2015.] <http://tikal.cifn.unam.mx/~jsegura/LCGII/java2.htm>.
- Definición. De. [En línea] [Citado el: 24 de febrero de 2015.] <http://definicion.de/inteligencia-artificial/#ixzz2u0CZUAH>.
- Diagramas UML. Software y Aplicaciones Web. [En línea] 14 de Febrero de 2008. [Citado el: 12 de diciembre de 2014.] <http://www.jtmentor.com.ar/post/UML-Diagramas.aspx>.
- Distribuidores de Programas de Gestión. @PsicoClinic. [En línea] [Citado el: 13 de Diciembre de 2014.] <http://www.e-rem.net/psicoclinic.html>.
- Dra. Assetta, A., Dr. Fernández Romero, D. y Dr. Rosell, S. Sistemas de Información Hospitalaria (SIH). Su importancia para el desarrollo de los servicios de salud y el control de la gestión. [En línea] [Citado el: 2 de Noviembre de 2014.] <http://www.medicos-municipales.org.ar/bc1104.htm#1>.
- DZone Refcardz. [En línea] [Citado el: 3 de febrero de 2015.] <http://refcardz.dzone.com/refcardz/seam-ui>.
- Egíluz Pérez, Javier. Librosweb.es. HTML y XHTML. Capítulo 1: Introducción. [En línea] [Citado el: 8 de Enero de 2015.] http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html.
- Facelets. [En línea] [Citado el: 10 de febrero de 2015.] <http://www.google.com/notebook/public/06237054388688325750/BDSW5QgoQ38aBIMsi>.
- Fernández Puerto, Francisco J. y Puerto, Florina Lara. Sistema de Información. México: s.n., 2003.
- Fernández, Karina y Toledo, Viviana. DITRITS: Sistema experto para diagnóstico y tratamiento de enfermedades de transmisión sexual. Cuba: s.n.
- García, Joaquín. Ingeniero Software. [En línea] [Citado el: 1 de marzo de 2015.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

- García, Raúl. Modelo de Datos. [En línea] [Citado el: 4 de marzo de 2015.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos>.
- Gómez, Juan. Fundamentos de la metodología RUP. [En línea] [Citado el: 14 de febrero de 2015.] <http://www.scribd.com/doc/297224/RUP>.
- Guia-ubuntu. [En línea] [Citado el: 20 de febrero de 2015.] http://www.guia-ubuntu.com/index.php/PgAdmin_III.
- Gutiérrez, Mariela, Barroso Rodríguez, Yadira y Bedoya Rusenko, Jorge. Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias. Cuba: s.n.
- Hergueta González, Celia. Sistema de Detección y Tratamiento de Enfermedades Cutáneas Madrid: s.n., 2006.
- Hinto, Perry R. Statistic Splained.
- Jaramillo, Wilmer. Software Libre de Venezuela 777, C.A. [En línea] [Citado el: 22 de febrero de 2015.] <http://wilmer.fedorapeople.org/files/presentations/JBoss.pdf>.
- JBoss. [En línea] [Citado el: 3 de febrero de 2015.] <http://www.jboss.com/products/hibernate/>.
- León, Jeimy. Mis primeros pasos por el mundo de la arquitectura del software.
- Loaiza, Roger. De la información a la informática. [En línea] [Citado el: 12 de enero de 2015.] http://www.bvs.sld.cu/revistas/san/vol2_2_98/san15298.htm.
- Presman, Roger. Ingeniería de Software, un enfoque práctico.
- Quisbert Limanchi, Nancy Susana y Marca Hullapar, Hugo Michael. Diagrama de despliegue.
- Reindustria. [En línea] [Citado el: 24 de marzo de 2015.] <http://redindustria.blogspot.com/2008/03/arquitectruas-clienteservidor.html>.
- Sánchez Corrales, Yosvannys. Método computacional para el diagnóstico de la Hipertensión. La Habana : s.n., 2012.
- SeamFramework.org. [En línea] [Citado el: 12 de febrero de 2015.] <http://seamframework.org/>.
- SeamFramework.org. [En línea] [Citado el: 15 de febrero de 2015.] <http://docs.jboss.org/seam/latest/reference/en-US/html/gettingstartedwithjbossstools.html>.
- SeamFramework.org. [En línea] [Citado el: 15 de febrero de 2015.] <http://seamframework.org>.
- Sergio Ramírez Q, Cristina Ponce E, Indira Nolvos A. SISTEMA EXPERTO PARA LA DETECCION DE CANCER A LA GLANDULA TIROIDES - SIECAT. s.l.: Ecuador.

- Soto, Carlos. Sistema Experto para el diagnóstico del síndrome de Guilliam Barre.
- Suárez, José Manuel Sánchez. Introducción a Richfaces. [En línea] [Citado el: 6 de marzo de 2015.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflIntro>.
- Tecnología y Synergix. [En línea] [Citado el: 3 de marzo de 2015.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio>.
- Tello, Jesús Cáceres. [En línea] [Citado el: 12 de marzo de 2015.] <http://msdn.microsoft.com/es-es/library/dd409377.aspx>.
- Vallejo, Juan Santaella. Un Sistema Experto: MyCIN.
- Velázquez Carralero, Alejandro Mario. IH-SW-DR-091 Alas-His_Documento de Arquitectura del Sistema. 2008.
- Venete, Adriana. Introducción a los Patrones de Arquitectura. 2010.
- Visual Paradigm. [En línea] [Citado el: 20 de marzo de 2015.] <http://www.visual-paradigm.com/news/vpsuite33/vpuml63.jsp>.

GLOSARIO DE TÉRMINOS

Arquitectura: se define como un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción de software para un sistema informático. La arquitectura de software establece los fundamentos para que los analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema informático.

Artefacto tecnológico: es cualquier obra manual o digital realizada con un propósito o función técnica específica aplicando la tecnología. Se consideran artefactos los diagramas, informes, modelos, entre otros.

Código: cifras, clave. En informática se utiliza para referirse a un conjunto de instrucciones en un lenguaje de programación.

CIE10: es la décima versión de la Clasificación Estadística Internacional de Enfermedades y otros Problemas de Salud. Provee los códigos para clasificar las enfermedades y una amplia variedad de signos, síntomas, hallazgos anormales, denuncias, circunstancias sociales y causas externas de daños y/o enfermedad. Cada condición de salud puede ser asignada a una categoría y darle un código de hasta seis caracteres de longitud.

Framework: estructura predefinida para la creación de aplicaciones. Puede estar formado por un conjunto de librerías y clases o por una arquitectura que facilita el desarrollo de software.

HTTP: es un protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. Se usa en cada transacción de datos en la web. Es usado por gran cantidad de entidades pues la información que manejan es de suma confidencialidad e importancia y por ello se necesita mantener cifrado el canal de transferencia.

Plataforma: en informática, una plataforma es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software.

TCP/IP: protocolo de control de transmisión /Protocolo de Internet o un conjunto de protocolos. Representa las reglas de comunicación para internet y se basa en las direcciones IP de cada equipo en la red para poder enrutar paquetes de datos.

ANEXOS

Anexo 1: Diagrama de procesos de negocio

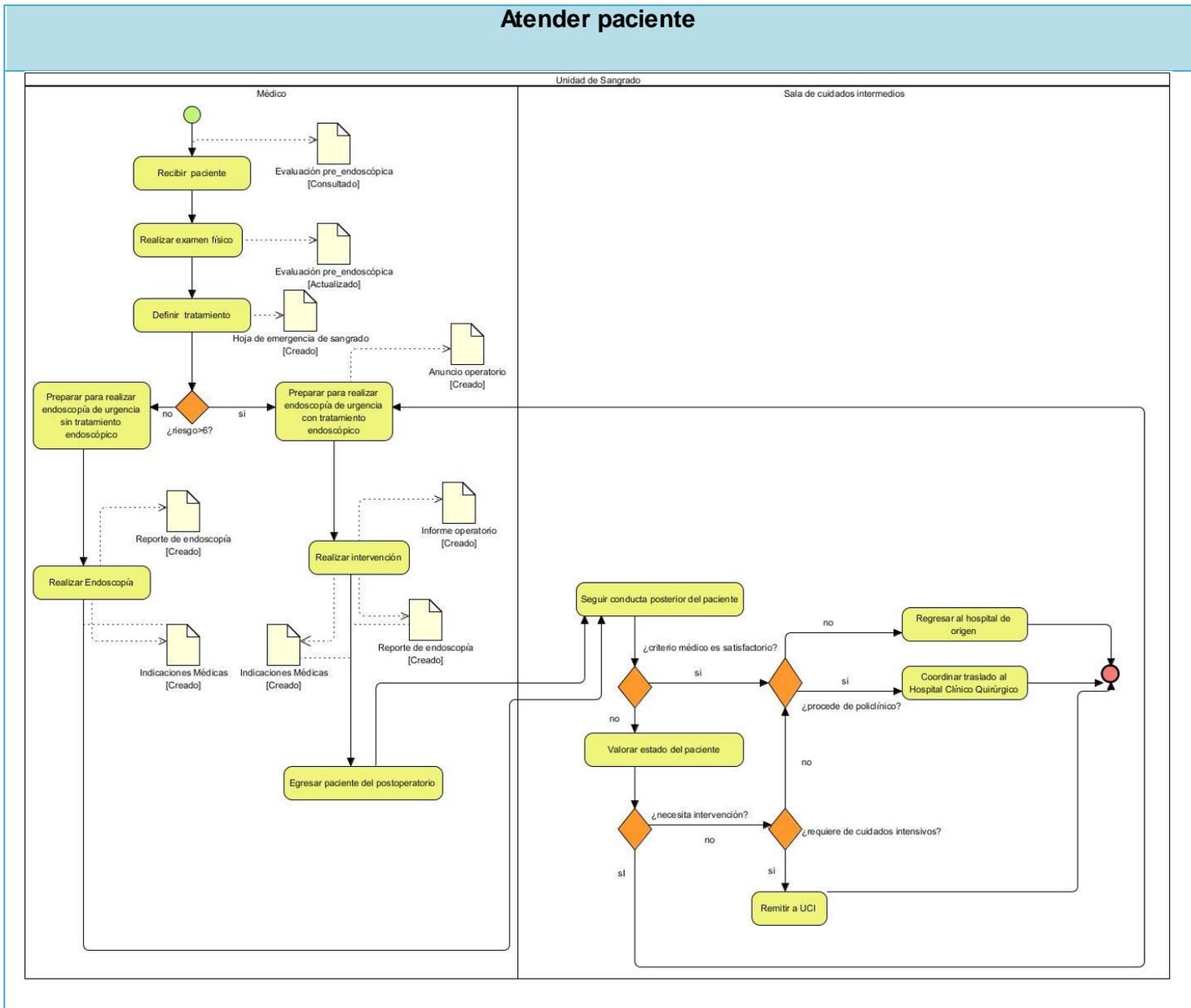


Figura 1 Proceso Atender paciente (elaboración propia)

Anexo 2: Diagramas de casos de uso

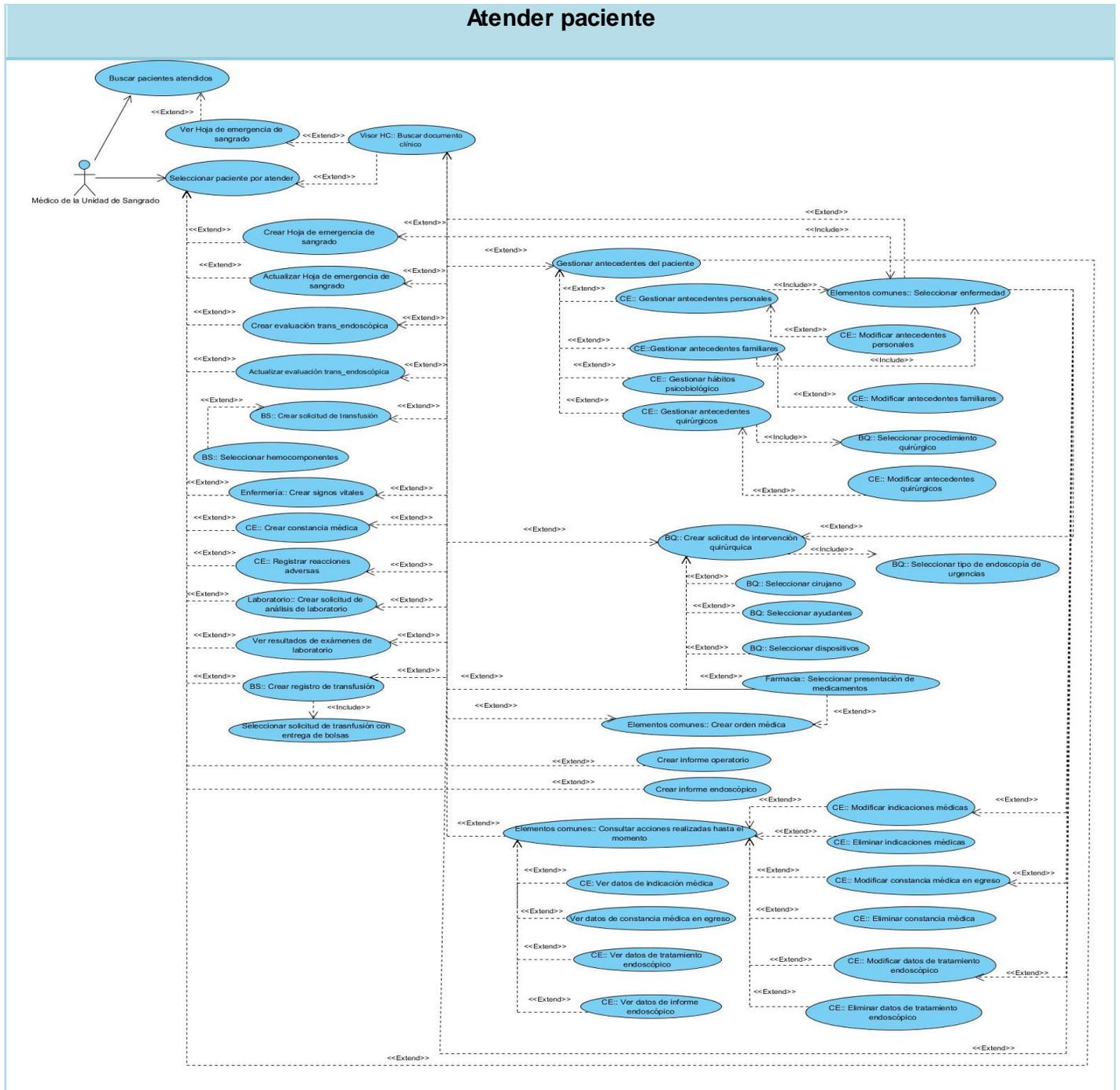


Figura 2: Diagrama de casos de uso del sistema: Atender paciente (elaboración propia)

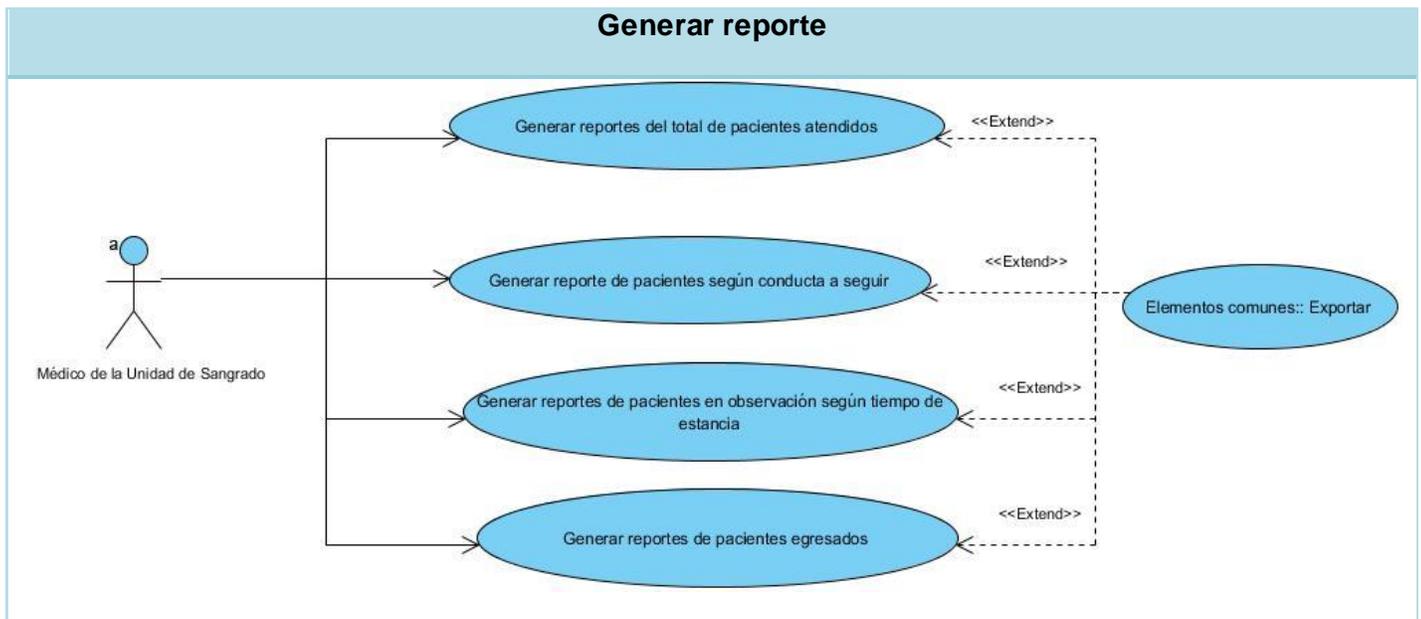


Figura 3: Diagrama de casos de uso del sistema: Generar reporte (elaboración propia)

Anexo 3: Diagrama de clases del diseño

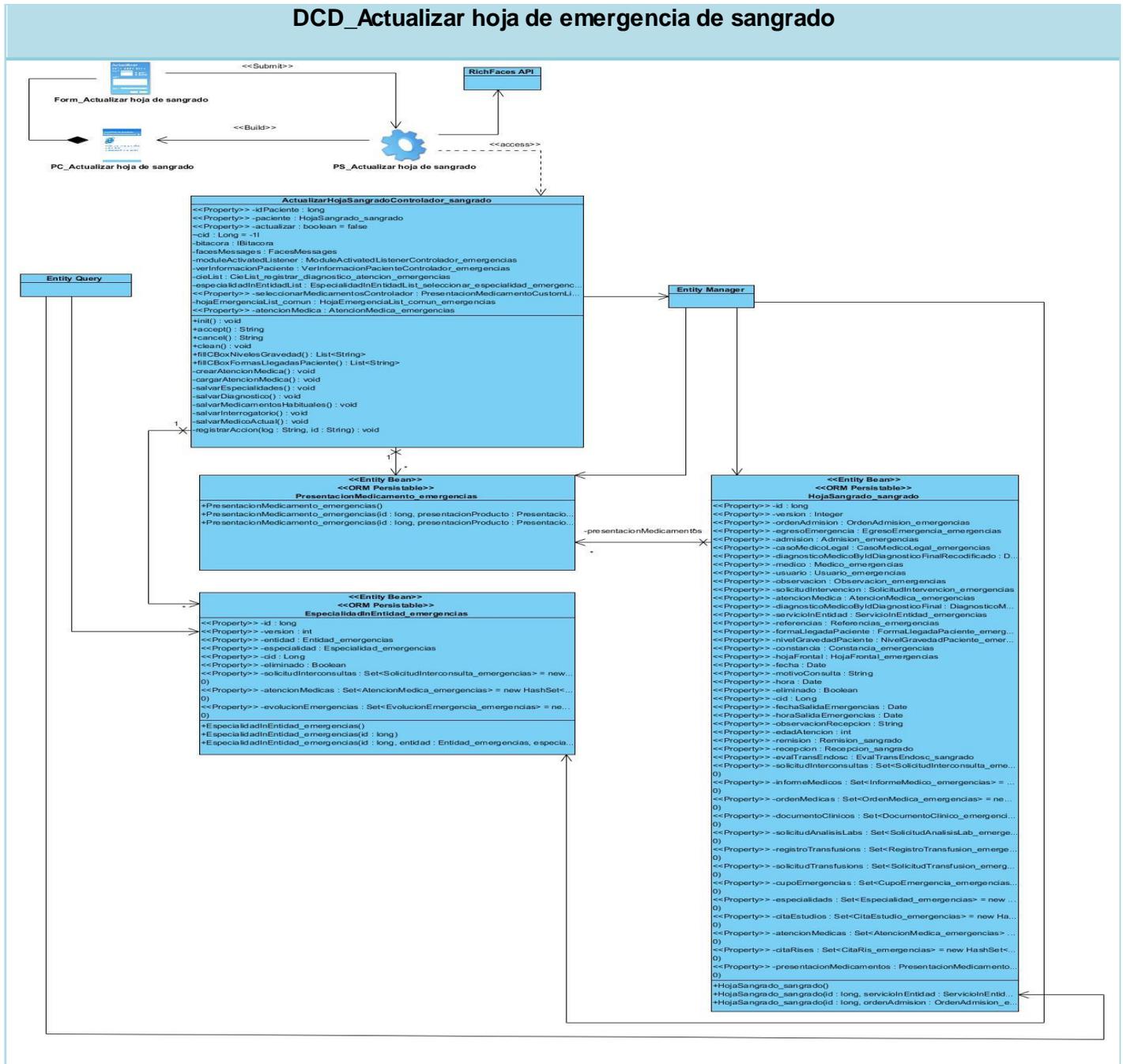


Figura 4: Diagrama de clases del diseño: Actualizar hoja de emergencia de sangrado (elaboración propia)

DCD_Crear evaluación trans_endoscópica

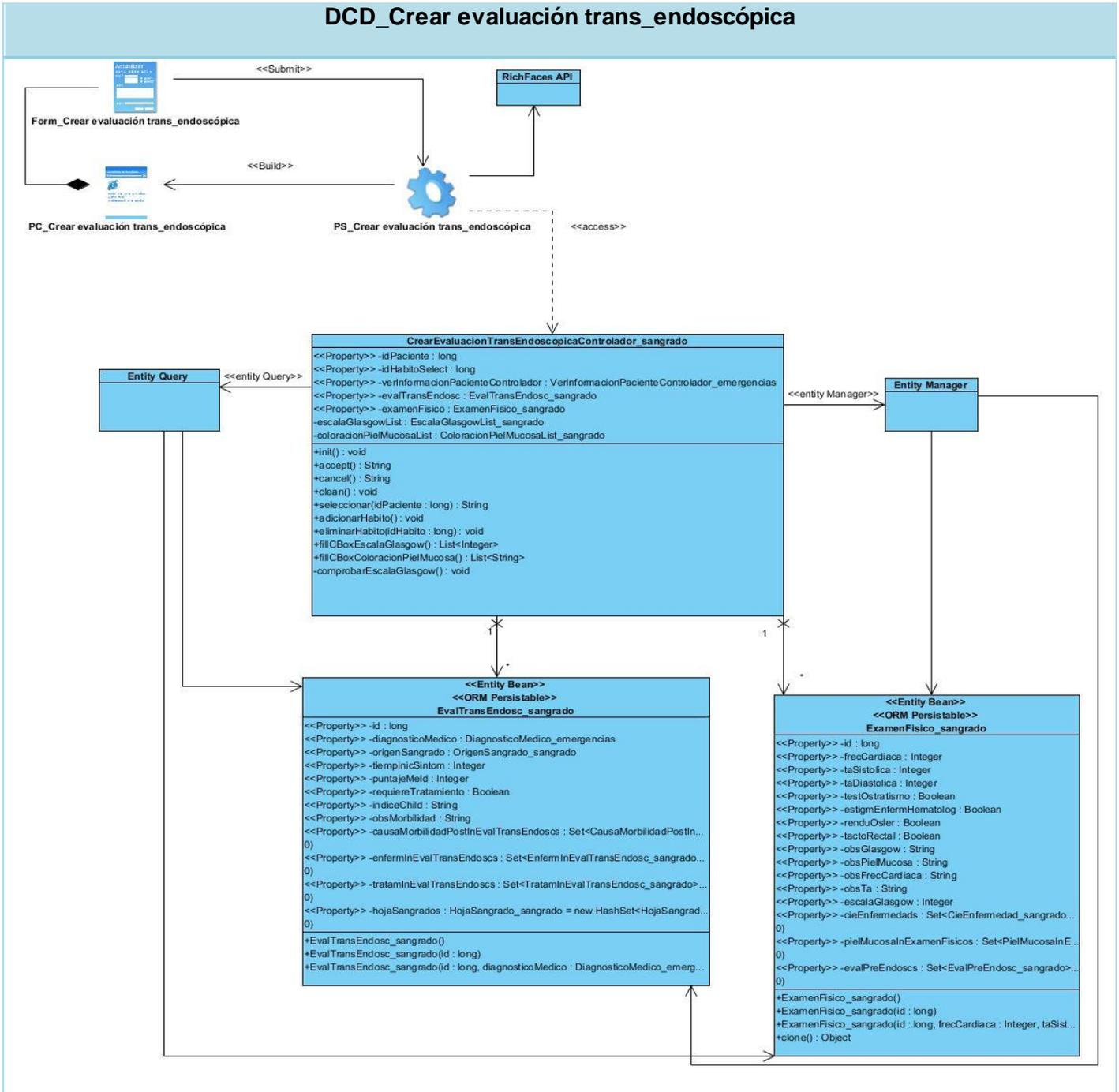


Figura 5: Diagrama de clases del diseño: Crear evaluación trans-endoscópica (elaboración propia)

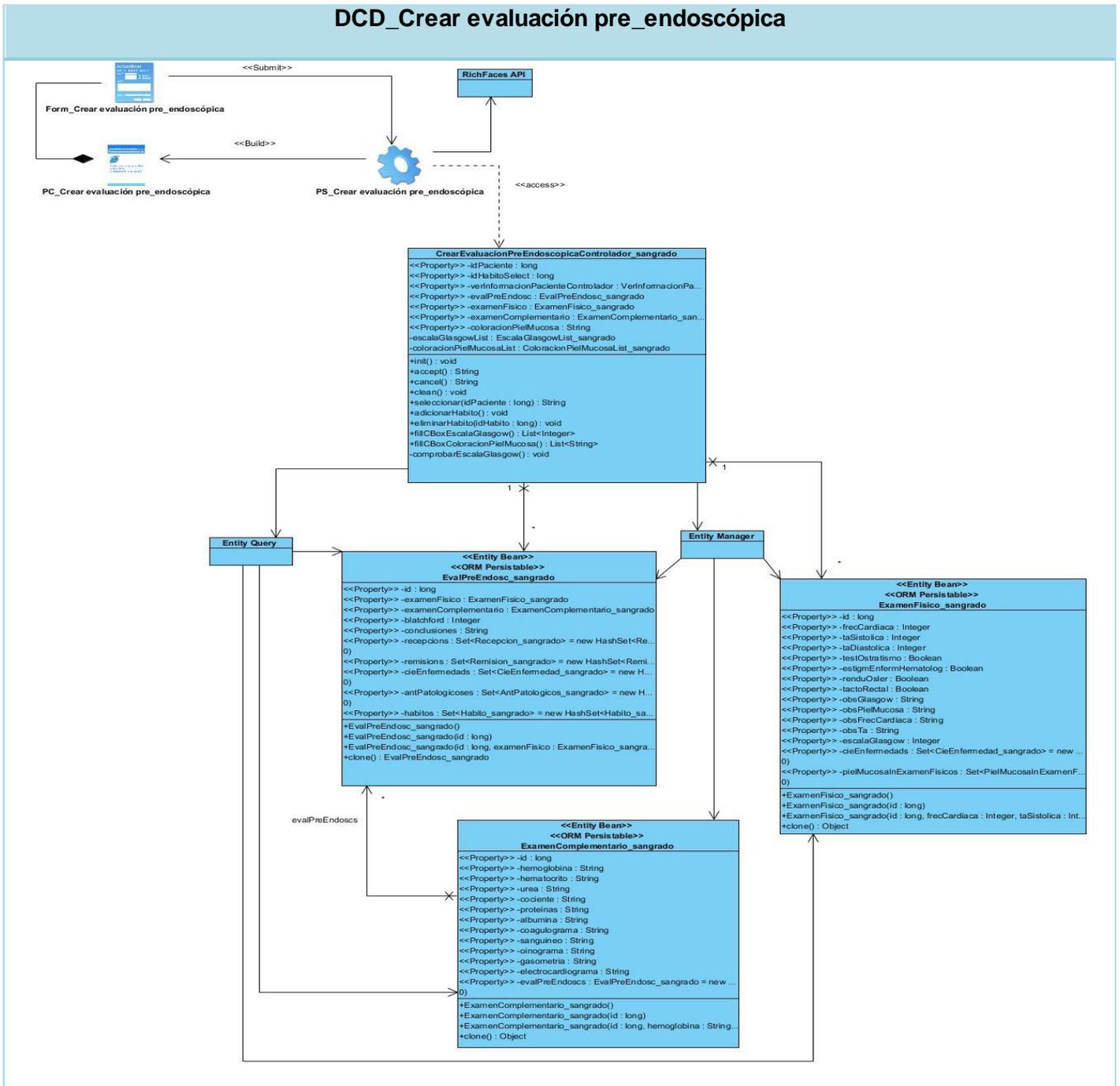


Figura 6: Diagrama de clases del diseño: Crear evaluación pre-endoscópica (elaboración propia)

Anexo 4: Funcionalidades incorporadas del módulo Emergencias

Tabla 1 Funcionalidades de Emergencias

Nombre	Descripción
Ver libro de emergencias	Muestra el listado de pacientes recibidos y atendidos en emergencias, cuyos datos coinciden con los criterios de búsqueda seleccionados.
Listar pacientes atendidos	Muestra el listado de los pacientes atendidos. En caso de ser atendidos por otro médico, muestra el nombre de este.
Gestionar antecedentes del paciente	Permite gestionar los antecedentes del paciente.
Ingresar en observación	Permite ubicar al paciente en la sala de observación.
Crear orden medica	Permite crear la orden médica que debe seguir el paciente una vez abandonada la atención.
Crear solicitud de análisis de laboratorio	Permite enviar una solicitud de análisis de laboratorio.
Asignar cita para estudio imagenológico	Permite enviar una solicitud de estudio imagenológico.
Consultar resultado de exámenes	Permite consultar el resultado de exámenes de laboratorio o estudios imagenológicos indicados.
Crear solicitud de transfusión	Permite enviar una solicitud de transfusión de sangre.
Crear solicitud de intervención quirúrgica	Permite enviar una solicitud de intervención quirúrgica.
Crear orden de admisión	Permite enviar una orden de admisión.