

**Universidad de las Ciencias Informáticas
Facultad 7**



**Título: MÓDULO BANCO DE SANGRE DEL SISTEMA DE
INFORMACIÓN HOSPITALARIA alas HIS**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores: Aliuska Marrero Nieblas
Yoel Odelso Suárez Rivero

Tutores: Ing. Suleydis Suárez Serpa
Lic. Angel Fabra Torres

Ciudad de La Habana, Junio de 2009
“Año del 50 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 24 días del mes de junio del año 2009.

Aliuska Marrero Nieblas

Firma del Autor

Yoel Odelso Suárez Rivero

Firma del Autor

Ing. Suleydis Suárez Serpa

Firma del Tutor

Lic. Angel Fabra Torres

Firma del Tutor

DATOS DE CONTACTO

Síntesis del Tutor: Ing. Suleydis Suárez Serpa.

Instructor recién graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor Facultad # 7. Ha impartido las asignaturas Matemática Discreta y Álgebra Lineal. Forma parte del proyecto Gestión Hospitalaria.

Correo electrónico: ssuarez@uci.cu

Síntesis del Tutor: Lic. Ángel Fabra Torres.

Graduado de Licenciatura en Ciencias de la Computación en el año 2004 en la Universidad de Oriente. Actualmente se desempeña como profesor universitario en la Universidad de las Ciencias Informáticas (UCI) donde ha impartido asignaturas como Sistemas de Bases de Datos, Práctica Profesional 1 e Introducción a la Programación. Posee categoría docente de Asistente. Actualmente se desempeña como Jefe del Módulo de Consulta Externa del Proyecto de Informatización de los Hospitales del Sistema de Salud Pública de Venezuela.

Correo electrónico: afabra@uci.cu

AGRADECIMIENTOS

De Aliuska:

A todos los que me apoyaron y contribuyeron con mi educación. A mis tutores.

DEDICATORIA

De Aliuska:

A mis padres, mis hermanos y mi familia en general.

A Yunior, mi papi.

A un gran amigo, Maykell Sánchez.

RESUMEN

En la actualidad la mayoría de los hospitales no cuentan con un sistema automatizado que lleve el control de las actividades que se realizan en los bancos de sangre. La gestión de las donaciones y transfusiones es de gran importancia para un correcto funcionamiento del banco de sangre. Estos datos son recogidos en papel, por lo que el análisis y actualización de esta información resulta muy engorroso, pues el volumen de documentos es considerable. Esto provoca que en ocasiones, dicha información no sea fiable y puedan cometerse errores que atenten contra la salud de donantes y pacientes.

El presente trabajo tiene como objetivo: desarrollar el módulo Banco de Sangre del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias. Para su desarrollo se siguieron los pasos que propone el Proceso Unificado del Desarrollo (RUP), utilizando el Lenguaje Unificado de Modelado (UML) como soporte a la metodología y Visual Paradigm para el modelado visual. Para la implementación del sistema se utilizó el patrón Modelo Vista Controlador, y para dar soporte a todas las tecnologías utilizadas la Plataforma Java Enterprise Edition.

Se espera que el sistema contribuya a la solución de los problemas actuales que existen en los bancos de sangre hospitalarios. Lo que permitirá mejorar los procesos relacionados con las donaciones y transfusiones y que la información que estos generan, esté actualizada y accesible en tiempo real.

PALABRAS CLAVE

Banco de sangre, sangre, donaciones, transfusiones.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1. <i>Conceptos básicos asociados al problema.....</i>	6
1.2. <i>Sistemas automatizados existentes vinculados al campo de acción.....</i>	7
1.3. <i>Tendencias y tecnologías actuales a considerar.....</i>	13
1.4. <i>Tecnologías y herramientas utilizadas en el proceso de desarrollo.....</i>	15
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	25
2.1. <i>Flujo actual de los procesos involucrados en el campo de acción.....</i>	25
2.2. <i>Objeto de automatización.....</i>	27
2.3. <i>Información que se maneja.....</i>	27
2.4. <i>Modelo de Negocio.....</i>	29
2.5. <i>Propuesta del Sistema.....</i>	35
2.6. <i>Modelo de Casos de Uso del Sistema.....</i>	41
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	54
3.1. <i>Descripción de la arquitectura, fundamentación.....</i>	54
3.2. <i>Modelo de diseño.....</i>	55
3.3. <i>Definición de elementos de diseño.....</i>	57
3.4. <i>Diagramas de clases del diseño.....</i>	59
3.5. <i>Diagramas de interacción.....</i>	65
3.6. <i>Descripción de clases del diseño.....</i>	66
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....	69
4.1. <i>Modelo de datos.....</i>	69
4.2. <i>Descripción de las tablas.....</i>	71
4.3. <i>Diagrama de despliegue.....</i>	79
4.4. <i>Vista de componentes.....</i>	80
4.5. <i>Tratamiento de errores.....</i>	82
4.6. <i>Seguridad.....</i>	82
4.7. <i>Estrategias de codificación. Estándares y estilos a utilizar.....</i>	83
CONCLUSIONES.....	88
RECOMENDACIONES.....	89
REFERENCIAS BIBLIOGRÁFICAS.....	90
ANEXOS 95	
1. <i>Diagramas de secuencia.....</i>	95
2. <i>Pantallas del sistema.....</i>	97
GLOSARIO 101	

ÍNDICE DE FIGURAS

Figura 2.1.	Diagrama de procesos del negocio.....	30
Figura 2.2.	Diagrama de actividades del proceso Atender Donante.....	31
Figura 2.3.	Diagrama de actividades del proceso Realizar Análisis.	32
Figura 2.4.	Diagrama de actividades del proceso Realizar Transfusión.	33
Figura 2.5.	Vista global de actores del sistema.....	42
Figura 2.6.	Diagramas de casos de uso del proceso Atender Donante.....	42
Figura 2.7.	Diagramas de casos de uso del proceso Realizar Análisis.	43
Figura 2.8.	Diagramas de casos de uso del proceso Realizar Transfusión.	43
Figura 3.1.	Diagrama de paquetes del diseño.	56
Figura 3.2.	Diagrama de Clases del Diseño CUS9_ Crear ficha de donación.	60
Figura 3.3.	Diagrama de Clases del Diseño CUS3_ Registrar datos clínicos del donante.	61
Figura 3.4.	Diagrama de Clases del Diseño CUS7_ Asignar serial a la historia clínica del donante..	62
Figura 3.5.	Diagrama de Clases del Diseño CUS15_ Buscar grupo y factor por bolsa de componentes sanguíneos.	63
Figura 3.6.	Diagrama de Clases del Diseño CUS14_ Registrar resultados de exámenes.....	64
Figura 3.7.	Diagrama de Clases del Diseño CUS6_ Actualizar cuaderno de plaquetoaféresis.	65
Figura 4.1.	Modelo de datos.	70
Figura 4.2.	Diagrama de despliegue.....	80
Figura 4.3.	Diagrama de componentes.....	81
Anexo 1.1.	Diagrama de secuencia CU Asignar serial a la historia clínica del donante.	95
Anexo 1.2.	Diagrama de secuencia CU Registrar datos clínicos del donante.	96
Anexo 2.1.	Seleccionar paciente.	97
Anexo 2.2.	Crear Historia clínica del donante.	97
Anexo 2.3.	Registrar datos clínicos.....	98
Anexo 2.4.	Asignar serial a la Historia clínica del donante.....	99
Anexo 2.5.	Registrar donación.....	100

ÍNDICE DE TABLAS

Tabla 1.1.	Sistemas para la gestión de información en Hospitales.	8
Tabla 1.2.	Sistemas para la gestión de información en los bancos de sangre.	8
Tabla 2.1.	Actores del negocio.	33
Tabla 2.2.	Trabajadores del negocio.	34
Tabla 2.3.	Actores del sistema.	41
Tabla 2.4.	Descripción del Caso de Uso del sistema Crear historia clínica del donante.	44
Tabla 2.5.	Descripción del Caso de Uso del sistema Seleccionar historia clínica del donante.	44
Tabla 2.6.	Descripción del Caso de Uso del sistema Registrar datos clínicos del donante.	45
Tabla 2.7.	Descripción del Caso de Uso del sistema Descartar historia clínica del donante.	45
Tabla 2.8.	Descripción del Caso de Uso del sistema Registrar donación.	46
Tabla 2.9.	Descripción del Caso de Uso del sistema Actualizar cuaderno de plaquetoaféresis.	46
Tabla 2.10.	Descripción del Caso de Uso del sistema Asignar serial a la historia clínica del donante. 46	
Tabla 2.11.	Descripción del Caso de Uso del sistema Crear muestra de sangre.	47
Tabla 2.12.	Descripción del Caso de Uso del sistema Crear ficha de donación.	47
Tabla 2.13.	Descripción del Caso de Uso del sistema Actualizar serial de plasmaféresis.	47
Tabla 2.14.	Descripción del Caso de Uso del sistema Generar Reporte de listado de donantes.	48
Tabla 2.15.	Descripción del Caso de Uso del sistema Asociar grupo y factor a la bolsa de componentes sanguíneos.	48
Tabla 2.16.	Descripción del Caso de Uso del sistema Seleccionar solicitud de tipiaje.	49
Tabla 2.17.	Descripción del Caso de Uso del sistema Registrar resultados de exámenes.	49
Tabla 2.18.	Descripción del Caso de Uso del sistema Buscar grupo y factor por bolsa de componentes sanguíneos.	50
Tabla 2.19.	Descripción del Caso de Uso del sistema Asignar bolsa de componentes.	50
Tabla 2.20.	Descripción del Caso de Uso del sistema Buscar grupo y factor por bolsa de componentes sanguíneos.	51
Tabla 2.21.	Descripción del Caso de Uso del sistema Entregar bolsa de componentes a enfermera.	51
Tabla 2.22.	Descripción del Caso de Uso del sistema Desglosar bolsa de componentes.	52
Tabla 2.23.	Descripción del Caso de Uso del sistema Crear muestra de sangre.	52
Tabla 2.24.	Descripción del Caso de Uso del sistema Consultar solicitud de transfusión.	52
Tabla 2.25.	Descripción del Caso de Uso del sistema Realizar pruebas especiales.	53
Tabla 2.26.	Descripción del Caso de Uso del sistema Generar Reporte de trasfusiones preparadas y entregadas.	53
Tabla 3.1.	Descripción página cliente PC_CrearFichaDonacion.	66
Tabla 3.2.	Descripción página cliente PC_Registrar_Examen_Fisico.	66

Tabla 3.3.	Descripción página cliente PC_AsignarSerial.	66
Tabla 3.4.	Descripción página cliente PC_BuscarBolsaGrupoFactor.....	66
Tabla 3.5.	Descripción página cliente PC_RegistrarResultadoExamen.	67
Tabla 3.6.	Descripción página cliente PC_CuadernoPlaquetas.	67
Tabla 3.7.	Descripción clase controladora CrearFichaDonacionControler.	67
Tabla 3.8.	Descripción clase controladora Registrar_Examen_Fisico.....	67
Tabla 3.9.	Descripción clase controladora Asignar_Serial.	67
Tabla 3.10.	Descripción clase controladora BolsaSangreGrupoList.	68
Tabla 3.11.	Descripción clase controladora RegistrarResultadoExamenCONTROLLER.....	68
Tabla 3.12.	Descripción clase controladora Cuaderno_Plaquetas.....	68
Tabla 4.1.	Descripción de la tabla historia_clinica_donante.....	71
Tabla 4.2.	Descripción de la tabla hc_donaciones.....	72
Tabla 4.3.	Descripción de la tabla grado_aceptacion_donante.....	72
Tabla 4.4.	Descripción de la tabla datos_clinicos.	73
Tabla 4.5.	Descripción de la tabla bolsa_sangre.	73
Tabla 4.6.	Descripción de la tabla cuaderno_de_plaquetas.....	74
Tabla 4.7.	Descripción de la tabla solicitud_tipiaje.....	74
Tabla 4.8.	Descripción de la tabla resultado_tipiaje.....	75
Tabla 4.9.	Descripción de la tabla serial_plasma.....	75
Tabla 4.10.	Descripción de la tabla hc_transfusiones.....	76
Tabla 4.11.	Descripción de la tabla solicitud_transfusion.....	76
Tabla 4.12.	Descripción de la tabla prueba_compatibilidad.....	77
Tabla 4.13.	Descripción de la tabla pruebas_especiales.	78
Tabla 4.14.	Descripción de la tabla coombs.....	79
Tabla 4.15.	Descripción de la tabla tipo_reaccion_transfusion.	79
Tabla 4.16.	Convenciones del estándar de codificación de SUN Microsystem.	85
Tabla 4.17.	Especificaciones de los tipos de datos.....	86
Tabla 4.18.	Especificaciones de los controles.	86

INTRODUCCIÓN

El creciente desarrollo de la informática a nivel mundial proporciona beneficios a todos los sectores de la sociedad. Entre ellos se ha observado un progreso paralelo en la utilización de las tecnologías de la información y la comunicación con el objetivo de mitigar algunos de los problemas actuales del sistema de salud a nivel hospitalario y contribuir al funcionamiento para apoyar la salud pública.

El uso de la informática en la medicina es una de las aplicaciones más comunes e importantes desde hace varias décadas. Ha permitido al sector de la salud no sólo contar con métodos novedosos, sencillos y eficaces de gestión administrativa en consultas, hospitales y centros de investigación biomédica, sino también disponer de complejos software que reducen la posibilidad de error en el diagnóstico de las enfermedades y que aceleran su formulación.

Los sistemas de información han existido desde hace mucho tiempo. Aunque no eran automatizados, ni mucho menos sistematizados. A raíz del surgimiento de las computadoras, comenzaron a crearse de forma más sencilla, tanto con fines administrativos como financieros; pero no fue hasta la década del 1970 cuando en el ámbito médico se perfilan los primeros sistemas de información "médica" que, posteriormente, dieron lugar a los Sistemas de Información Hospitalaria.

Actualmente, el interés de múltiples países por adoptar una política de optimización de sus sistemas de salud se ha generalizado, centrándose en los beneficios aportados por los Sistemas de Información Hospitalaria. Estos deben resolver las deficiencias en el funcionamiento de sus instituciones y elevar la capacidad de respuesta ante el aumento del volumen de personas a atender.

Un Sistema de Información Hospitalaria (HIS, por sus siglas en inglés) está orientado a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria. Permite la optimización de los recursos humanos y materiales, además de minimizar los inconvenientes burocráticos que enfrentan los pacientes. Todo HIS genera reportes e informes, en dependencia del área o servicio para el cual se requiera, permitiendo la retroalimentación de la calidad asociada a la atención de los servicios de salud.(1)

Desde las principales potencias -que solicitan el perfeccionamiento de sistemas con algunas carencias- hasta países menos pudientes -con la necesidad de integrar a sus servicios médicos una gestión rápida de los datos con los que se maneja el tratamiento de los pacientes-, se muestran interesados en una solución viable. Entiéndase un producto manuable y eficiente a la vez; capaz de

facilitar la manipulación de la información recopilada hasta la fecha y que gestione la que se obtiene con el trabajo diario.(2)

En Cuba, consciente de la necesidad de perfeccionar la atención al paciente, se ha continuado con la informatización en la salud dada la diversidad de los servicios prestados y la calidad de los mismos. Lo que en ocasiones demora el proceso, en consecuencia los órganos rectores de las diferentes instituciones de salud han priorizado el desarrollo de la informática en la salud.

La práctica de la medicina es “dominada por el modo de procesar o reprocesar la información, recuperarla y comunicarla”. El tratamiento y la obtención de información clara, precisa, confiable y oportuna para el planeamiento, la organización y la toma de decisiones, resulta en este contexto un factor estratégico de éxito. Un exitoso sistema de información se apoya en el contenido, la estructura, la fiabilidad y adecuación de su información y es necesario entender que la información de salud y su administración pueden ser abordadas desde diferentes perspectivas.(3)

La UCI, protagonista actual de todo el proceso de informatización que se desarrolla en Cuba, tiene como una de sus tareas primordiales el desarrollo del sistema alas HIS. Este persigue dar solución a las necesidades de las diferentes entidades médicas que lo soliciten dentro o fuera del país.

El alas HIS consta de varios módulos estrechamente relacionados, entre ellos uno encargado de manejar todo lo referente a las donaciones, transfusiones y gestión de la donación en los bancos de sangre. De esta forma se centralizan y controlan todos los movimientos de las bolsas de sangre según su tipo, caducidad, traslados, pedidos y reservas de las mismas. Todo lo cual permite lograr una gestión adecuada de esta información.

Uno de los grandes problemas que presentan los bancos de sangre con respecto a los donantes, es la fecha de la última donación. Pues si se hace una donación antes del tiempo establecido, puede traer efectos negativos para su salud. Por otra parte, existen transfusiones donde es imprescindible conocer los datos de los pacientes con anterioridad. También es conveniente conocer los antecedentes patológicos o de carácter epidemiológico, que pueden haber sido diagnosticados en el interrogatorio o en el examen médico. Los datos recogidos se archivan en la historia clínica, fichas, tarjetas y otros formularios en formato papel, lo que trae como consecuencia que realizar una consulta se haga muy difícil.

La gestión de forma manual de los procesos mencionados anteriormente, dificulta el funcionamiento óptimo de estas instituciones. Ya que puede perderse o duplicarse información, o pueden cometerse errores que atenten contra la salud de las personas, lo que podría entorpecer el trabajo del personal que allí labora.

Basado en lo antes expuesto se tiene como **problema a resolver**: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de banco de sangre de las instituciones hospitalarias?

En correspondencia con el problema el **objeto de estudio** lo constituye el proceso de gestión de información en las instituciones hospitalarias. El **campo de acción** está centrado en el proceso de gestión de información en el área de banco de sangre de las instituciones hospitalarias.

Para dar solución al problema antes mencionado se define como **objetivo general**: Desarrollar el módulo Banco de Sangre del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento al objetivo general se han definido las siguientes **tareas**:

1. Evaluar las tendencias actuales en el mundo de los HIS.
2. Analizar los procesos de negocio asociados al área de banco de sangre de las instituciones hospitalarias.
3. Asimilar la arquitectura definida por el Área Temática Gestión Hospitalaria para el desarrollo de sus aplicaciones.
4. Obtener mediante el Proceso Unificado de Desarrollo, los flujos de trabajo de “Modelado de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
5. Implementar los procesos de negocio relacionados con la atención al donante, la realización de análisis y la realización de transfusiones del área de banco de sangre.

En este sentido se puede destacar que el desarrollo del módulo Banco de Sangre del Sistema alas HIS, proporcionará un grupo de beneficios, entre los que pueden ser mencionados los siguientes:

1. Seguridad y confidencialidad de toda la información que se obtiene y procesa en los bancos de sangre.
2. Organización de los diferentes procesos que se llevan a cabo en los bancos de sangre.
3. Garantizar que toda la información generada durante el trabajo en los bancos de sangre sea almacenada en Bases de Datos que permitan hacer uso de ellas en cualquier momento.
4. Obtención de datos estadísticos en tiempo real.
5. Contribuirá al mejoramiento de la calidad y cantidad de la sangre y sus componentes que se utilizan en los pacientes que así lo requieren en las instituciones hospitalarias, así como una mayor satisfacción y seguridad a los donantes que acudan a los bancos de sangre hospitalarios.
6. Minimizará los posibles errores de identificación debido a factores humanos.
7. Permitirá la obtención de Estadísticas reales y rápidas.
8. Permitirá una identificación rápida y segura de la sangre.
9. Proporcionará información que no es posible obtener mediante los sistemas tradicionales como la de donantes repetitivos, tipos de donantes y otras.

El presente documento se encuentra estructurado en cuatro capítulos, el primero de ellos es **“FUNDAMENTACIÓN TEÓRICA”**, donde se realiza un estudio del estado del arte en cuanto a sistemas de información hospitalarios, tecnologías, metodologías y herramientas existentes en el mundo.

El siguiente capítulo, **“CARACTERÍSTICAS DEL SISTEMA”**, contiene un marco conceptual asociado a la información que será manipulada por el sistema. Se arriba a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

El tercer capítulo **“DISEÑO DEL SISTEMA”** se centra en la modelación detallada y la construcción de la estructura de la aplicación.

En el cuarto y último, **“IMPLEMENTACIÓN”**, se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los requerimientos de seguridad de los proyectos que pertenecen al Área Temática Gestión Hospitalaria.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se ofrece una introducción a los sistemas informáticos para la gestión de la información en los bancos de sangre, con el objetivo de profundizar en los conceptos que brindan soporte a la investigación. Asimismo, se presenta el estado del arte de los sistemas semejantes al propuesto y sus rasgos fundamentales, para de esta forma entender la necesidad de una nueva solución, acorde a los requerimientos que en la actualidad tienen los hospitales. Por último, se exponen las características de las diferentes tecnologías y herramientas que serán empleadas en el desarrollo de la solución propuesta.

1.1. Conceptos básicos asociados al problema

Banco de sangre

Establecimiento médico hospitalario donde se planifica y promueve la donación de sangre, se efectúa la extracción de la misma y se realizan programas de separación de componentes. También se almacenan y conservan las unidades de sangre y sus hemoderivados que serán luego transfundidos a los pacientes que lo necesiten.

Sangre

La sangre es un tejido líquido que recorre el organismo transportando células, y todos los elementos necesarios para realizar sus funciones vitales (respirar, formar sustancias, defenderse de agresiones) y todo un conjunto de funciones muy complejas e importantes para la vida. La cantidad de sangre de una persona está en relación con su edad, peso, sexo y altura, una persona adulta se puede considerar que tiene entre 4,5 y 6 litros de sangre. Todos los órganos del cuerpo humano funcionan gracias a la sangre que circula por arterias, venas y capilares.

Donación voluntaria de sangre

Se refiere a la práctica de donación de sangre proveniente de donadores exclusivamente voluntarios. En contraparte se reconoce la donación de reposición en la que básicamente un familiar, amigo o conocido atiende una petición de donación de sangre en favor de alguna persona o paciente.

Donante de sangre

Individuo cuya edad debe estar comprendida entre 18 y 65 años, su peso mínimo no debe ser inferior a 50Kg y debe gozar de buena salud, pues la más avanzada tecnología no ha sido capaz de producir este elemento esencial para la vida. La única posibilidad de obtenerla es gracias a la generosidad personal del ser humano, único capaz de fabricarla en su propio organismo.

Transfusión de sangre.

La transfusión de sangre es el procedimiento médico de incorporar sangre o sus derivados procedentes de un individuo en el sistema circulatorio de otro, se lleva a cabo para mantener con vida a los pacientes que han sufrido pérdidas excesivas de sangre por traumas o cirugía, o para proporcionar algún elemento necesario en caso de enfermedades que afectan la producción. (4)

1.2. Sistemas automatizados existentes vinculados al campo de acción

En la actualidad existen diversos software que se encargan de la gestión de la información en los bancos de sangre. Estos productos pueden estar desarrollados como sistemas independientes para gestionar la actividad de los bancos de sangre en una institución hospitalaria, o como un servicio independiente de un hospital. Existen otros que forman parte de un HIS y que interactúan con los demás módulos que componen el sistema.

La mayoría de estos sistemas tienen dos características fundamentales: la primera es que la gran mayoría están desarrollados sobre la base de software propietario, lo cual limita sus posibilidades de usarlo, modificarlo o redistribuirlos y cuyo código fuente no está disponible o tiene acceso restringido. La otra característica refleja la implementación de dichos sistemas sobre aplicaciones de escritorio, las cuales necesitan de una previa instalación en el cliente y consumen grandes cantidades de espacio en disco, de memoria y de CPU, además que no siempre pueden ser instalados en varias plataformas.

A continuación se muestran las tablas donde se recogen las características -anteriormente mencionadas- en sistemas existentes para la gestión de información hospitalaria y banco de sangre respectivamente:

HIS	B. Sangre	País	Lic. propietaria	Desktop	Web	Multiplataforma
Care2x	X	Alemania	—	X	—	X
xHosp	X	México	X	X	—	—
xHIS	X	España	X	—	X	X
Galen Hospital	X	Cuba	X	X	—	—
SoftMed	X	México	X	X	X	X

Tabla 1.1. Sistemas para la gestión de información en Hospitales.

Nombre	País	Lic. propietaria	Desktop	Web	Multiplataforma
TIMHEMO	México	X	X	X	X
e-DELPHYN	Emiratos Árabes	X	—	X	X
Gebans	Argentina	X	X	—	—
Galen Lab Banco de Sangre	Cuba	X	X	—	X
xBlood	México	X	X	—	—

Tabla 1.2. Sistemas para la gestión de información en los bancos de sangre.

Algunos de los sistemas anteriores se investigaron con profundidad y se obtuvo información relevante sobre los requisitos funcionales, los reportes, y la estructura y organización de sus módulos. Esto contribuyó en gran medida a enriquecer los conocimientos referentes a los procesos que se desarrollan en los bancos de sangre y a la comprensión de sistemas diseñados para soluciones similares. Para modelar el sistema propuesto, se tuvieron en cuenta los aspectos más relevantes de cada sistema estudiado:

1.2.1. xBlood

xBlood es un sistema para el apoyo operacional y el control administrativo integral de un banco de sangre o centro transfusional y cuenta con las siguientes funcionalidades: (5)

- Controla el manejo de donantes, receptores, así como los procesos y movimientos efectuados a las unidades de sangre.
- Permite un acceso controlado a las diversas funciones del banco de sangre llevando de la mano al usuario a través de las diferentes etapas de la donación por medio de un flujo de

trabajo así como de los procesos de transfusión, validación, cesión, reservación y destrucción de las unidades.

- Realiza el envío automático de solicitudes de estudios al laboratorio: biometría hemática, grupo y Rh, estudios de serología, anticuerpos irregulares así como de pruebas cruzadas.
- En caso de que el banco de sangre realice sus propios estudios localmente, hace posible la transcripción de resultados en forma manual o por medio de interfaces con equipos automatizados.
- Permite configurar en forma fácil y flexible las preguntas que conforman la historia clínica.
- Es configurable para que pueda rechazarse al pre-donante automáticamente. Captura del grupo sanguíneo y Rh, así como el fenotipo de la unidad.
- Puede configurarse para transferir automáticamente los datos de grupo ABO y Rh desde el laboratorio.
- Permite revisar los resultados de los estudios serológicos realizados a la unidad.
- Los estudios a realizar son configurables y dependen del tipo de donación.
- Los resultados pueden transferirse directamente del equipo autoanalizador por medio de interfaces.
- Permite configurar una validación manual o automática.
- Manejo de los diferentes procesos que pueden realizarse a cada hemocomponente: fraccionamiento, filtración, lavado, irradiación, etc.
- Registro de la fecha del proceso, el usuario que lo llevó a cabo así como la fecha de caducidad del hemocomponente.
- Pueden configurarse nuevos componentes, la cantidad por defecto que se maneja, la fecha de caducidad, el tipo de bolsa y anticoagulante, así como los procesos permitidos a cada componente.

- Permite la descarga de unidad o proceso de transfusión después de validar las pruebas cruzadas y anticuerpos irregulares en su caso.
- La asignación puede realizarse de las unidades disponibles o de las unidades reservadas específicamente a ese receptor.

1.2.2. **SoftMed/TIMHEMO**

SoftMed es una solución informática integral para hospitales y/o instalaciones médicas que requieran de una solución de este tipo, donde la información se encuentra en línea y con los departamentos automatizados en todas las áreas: médicas, hospitalarias y administrativas.

SoftMed cuenta con la solución para banco de sangre: TIMHEMO. Dicha solución permite contar con la información en línea y llevar un control minucioso de las unidades de sangre, receptores y donadores. Almacena y mantiene en línea la información vital para un banco de sangre como: evaluación de donantes, historia de receptores y resultados. Le permite llevar una hemovigilancia registrada, la cual podrá consultar y monitorear constantemente.

El sistema automatiza los procesos; recolección, conservación y distribución de sangre y derivados. Cuenta con un inventario de unidades el cual le permite consultar la disponibilidad de unidades no vencidas llevando así un control de la caducidad de unidades, a través de un tono de alerta para el desecho. Para un buen funcionamiento el sistema cuenta con los módulos siguientes: (6)

- Control de donantes.
- Historia clínica y estudios de serología.
- Etiquetas con código de barras.
- Hemocomponentes y separación de volúmenes.
- Validación y control de unidades.
- Control de transfusiones a receptores.
- Reportes y estadísticas.

- Control de calidad.
- Interfaces con el expediente clínico electrónico.

1.2.3. **Gebancks**

Gebancks es un sistema de gestión de banco de sangre que permite llevar adelante una eficiente manipulación de la información en esta área, desde el proceso de alta de la planilla de donante- pasando por los estudios serológicos e inmunohematológicos de las unidades de sangre y la subdivisión de la unidad en hemoderivados- hasta el registro de salida de unidades y hemoderivados a receptores. Lleva un control de stock de unidades y hemoderivados por fecha de vencimiento, control de donantes y cuenta de receptores. Cuenta con los módulos siguientes: (7)

- Registro de planilla del donante según norma estipulada.
- Historial de donantes.
- Historial de receptores.
- Historial de profesionales intervinientes en extracciones, análisis y entrega de unidades y hemoderivados.
- Consulta de estado de donantes (Según historial de donante).
- Consulta de estado de receptor (Según historial de receptor).
- Gestión de bolsas de sangre propias.
- Gestión de bolsas de sangre externas.
- Gestión de hemoderivados de la bolsa de sangre.
- Carga de análisis serológicos e inmunohematológicos por unidades de sangre.
- Control de stock de sangre por unidades y por hemoderivados con fechas de vencimiento.
- Estado de deuda de sangre por receptor.
- Gestión de donantes y receptores.

- Gestión de las bolsas y de stock de unidades de sangre y hemoderivados.
- Gestión de los estudios serológicos e inmunohematológicos.
- Informes estadísticos.

1.2.4. Galen Lab Banco de Sangre

Galen Lab Banco de Sangre constituye una solución integral para la gestión de los donantes y la distribución de la sangre y componentes sanguíneos. Basada en sistemas de gestión existentes en Cuba, este proyecto incluye los servicios de banco de sangre, transfusiones en las unidades hospitalarias y la industria que produce medicamentos a partir de los componentes sanguíneos recibidos de los bancos. (8)

- Registro del donante: Se comprueba que no esté excluido por alguna causa detectada en una donación anterior o por no haber transcurrido el tiempo adecuado desde la última donación.
- Chequeo médico: Se realiza el chequeo médico establecido y si cumple con los requisitos para donar, continúa para el salón de extracciones donde se efectúa la donación de sangre. Se colocan las etiquetas de código de barra a las bolsas y tubos de ensayo utilizados, los que se envían hacia el Laboratorio Suma para realizar los ensayos y certificar la sangre.
- Producción, aseguramiento de la calidad y distribución de la sangre: Se separan los diferentes componentes de la sangre, se imprimen las etiquetas de código de barra para cada componente que se almacena en bolsas a las que se le realiza el control de calidad y se les coloca una etiqueta de certificación para su utilización en los pacientes o instituciones que lo requieran.
- Producción de sueros y factor VII: Se obtiene la información sobre los antisueros creados y las bolsas de plasma fresco congelado (ya liberadas) que se utilizan en la producción de crioprecipitados.
- Donantes de plasmaféresis: La historia clínica del donante permite obtener toda la información personal relativa a los donantes de plasmaféresis y además los datos particulares de cada donación realizada por cada uno de ellos.

1.3. Tendencias y tecnologías actuales a considerar

En este epígrafe se exponen los elementos asociados al ambiente de desarrollo del sistema para lo cual es preciso definir la Arquitectura de Software, esta es la organización fundamental de un sistema encarnada en las relaciones entre ellos y el ambiente, sus componentes y los principios que orientan su diseño y evolución.(9)

Teniendo en cuenta las características del entorno donde se desplegará el producto a desarrollar y el estado del arte de los HIS en el mundo, se define que las tecnologías a utilizar en el proceso de desarrollo del sistema deben ser libres, multiplataforma y que faciliten u optimicen el mismo. A continuación se mencionan las definiciones de las tecnologías escogidas, así como sus ventajas:

1.3.1. Sistemas distribuidos. Modelo Cliente Servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes: (10)

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

1.3.2. Arquitectura en tres capas

La Arquitectura en tres Capas se basa en que el desarrollo se puede llevar a cabo en 3 niveles y, en caso de que sobrevenga algún cambio, sólo se interviene el nivel requerido sin tener que revisar entre el código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles.(11)

Capa de presentación: es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.

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 solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.(12)

1.3.3. Patrón de arquitectura Modelo-Vista-Controlador

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución. Dentro de las soluciones estudiadas el patrón más recomendado por especialistas para aplicaciones como la propuesta en el trabajo es el Modelo Vista Controlador (MVC).

Modelo: Contiene una lista de usuarios autorizados junto con sus contraseñas y la base de datos sobre la que efectuar las consultas y los módulos (clases o librerías de funciones) que se encarguen de realizar las operaciones (comprobar la información del usuario, construir la consulta, efectuar la consulta).

Vista: Muestra la información del modelo al usuario y al necesitar poco o nada de código, puede ser desarrollada por un equipo de diseñadores independiente del equipo de programadores.

Controlador: Gestiona las entradas del usuario y serviría de "pegamento", recoge las órdenes y datos que el usuario envía desde la vista, las traduce en operaciones del modelo de negocio y, sobre la base de los resultados los muestra en uno u otro formulario de la vista.

Entre sus principales ventajas se encuentran:

- Soporte de vistas múltiples: Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los datos de manera simultánea.
- Adaptación al cambio: Los requerimientos no funcionales de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas del negocio. Los clientes pueden preferir distintas opciones de representación pero dado que el modelo no depende de la vista, agregar nuevas opciones o modificar las ya existentes generalmente no afecta al modelo.(13)(14)(15)

1.4. Tecnologías y herramientas utilizadas en el proceso de desarrollo

La arquitectura en capas y el patrón MVC, pueden relacionarse lógicamente mediante cada uno de sus elementos, donde la capa de presentación podría corresponderse con la Vista, la capa de negocio con el Controlador y la capa de datos con el Modelo. A continuación se realiza una descripción de las tecnologías utilizadas en el proceso de desarrollo, agrupadas cada una en las capas donde son utilizadas:

1.4.1. Capa de presentación

1.4.1.1. Java Server Faces

Java Server Faces (JSF) es un framework de componentes de interfaz de usuario del lado del servidor para aplicaciones web basadas en Java. Contiene dos librerías de etiquetas personalizadas para JavaServer Pages (JSP) que permiten expresar una interfaz JavaServer Faces dentro de una página JSP. Incluye un API¹ para representar componentes de interfaz de usuario (UI), manejar sus

¹ API (Application Programming Interface) o Interfaz de Programación de Aplicaciones

estados, manejar sus eventos, la validación del lado del servidor y la conversión de datos, definir la navegación entre páginas y proporcionar extensibilidad para todas estas características.(16)

JSF es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador). Ofrece una clara separación entre el comportamiento y la presentación. Une los componentes UI con los conceptos de la capa-web sin limitarse a una tecnología de script o lenguaje de marcas particular.(17)

1.4.1.2. Richfaces

RichFaces es una rica biblioteca de componentes para JSF y un avanzado marco para integrar fácilmente capacidades AJAX en el desarrollo de aplicaciones de negocios, por lo que los desarrolladores pueden ahorrar tiempo y aprovechar las características de los componentes para crear aplicaciones Web que proporcionan en gran medida la mejora de la experiencia del usuario más fiable y más rápidamente. Se integra además el framework de extensión AJAX para los controles JSF básicos Ajax4Jsf.(18)

1.4.1.3. Ajax

Para lograr aumentar la interactividad, velocidad y usabilidad en las aplicaciones web se ha hecho muy común que los implementadores usen Ajax, este es el acrónimo para **Asynchronous JavaScript + XML** y con su uso se logra cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en *background*, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma. (19)

1.4.1.4. Ajax4JSF

Ajax4jsf es una librería open source 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 al servidor automáticas, control de cualquier evento de usuario, etc. Ajax4jsf permite dotar a la aplicación JSF de contenido mucho más profesional con muy poco esfuerzo.(20)

1.4.1.5. Facelets

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

- Tiempo de desarrollo cero de los tags para UIComponents.
- Facilidad en la creación del templating para los componentes y páginas.
- Habilidad de separar los componentes de interfaz de usuario en diferentes archivos.
- Un buen sistema de reporte de errores.
- Soporte completo a EL (Expression Language).
- Validación de EL en tiempo de construcción.
- No es necesaria configuración XML.
- Trabaja con cualquier RenderKit.(21)

1.4.1.6. Lenguaje de Marcado de Hipertexto Extensible (XHTML)

XHTML es el acrónimo inglés de eXtensible Hyper Text Markup Language (lenguaje extensible de marcado de hipertexto). Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo (como las hojas de estilo en cascada) y JavaScript su aspecto y diseño en distintos medios.(22)

1.4.1.7. Seam UI

Son una serie de controles JSF altamente integrables con JBoss Seam, adicionan varias mejoras a JSF desde validación, expresiones Extended EL e integración de la navegación en la interfaz de usuario basada en pageflows o procesos del negocio.(23)(24)

1.4.1.8. Extended EL

Extended EL provee una extensión al estándar Lenguaje de Expresión Unificado (EL) aumentando su expresividad y su poder. Este estándar adopta las características ofrecidas por el lenguaje de expresión de JSF. Mediante su utilización se puede reducir drásticamente la cantidad de código en las páginas creadas, lo que aumenta la productividad, y hace más fácil el mantenimiento y más pequeña la curva de aprendizaje en términos del desarrollo.

1.4.2. Capa de negocio

1.4.2.1. JBoss Seam

JBoss Seam es un framework Open Source desarrollado con el fin de unir diferentes tecnologías y estándares de Java en una sola plataforma, a la vez que añade algunas funcionalidades no contempladas por ellos. Este integra la capa de presentación (JSF) con la capa de negocio y persistencia. Con Seam basta agregar anotaciones propias de éste a los objetos Entidad y Session, para lograr escribir menos código Java y XML. Otra característica importante es que se puede hacer validaciones en los POJOs (Plain Object Java) y además manejar directamente la lógica de la aplicación y de negocios desde las Sessions beans. Seam también se integra perfectamente con otros frameworks como: RichFaces, ICEFaces (soportan AjaX) MyFaces, Hibernate.(25)

1.4.3. Capa de acceso a datos

1.4.3.1. Hibernate

Hibernate es un framework de código abierto para el mapeo objeto-relacional, permite desarrollar clases persistentes y lógica persistente sin importar cómo manejar los datos. Este no sólo se ocupa de la abstracción de los datos persistentes en el gestor de base de datos- utilizando el mapeo de clases Java a tablas de la base de datos-, sino que también provee facilidades de consultas de datos y de recuperación. Además, exime al desarrollador de un gran por ciento de las tareas de programación relacionadas con la persistencia de datos. Se utiliza para el mapeo de las clases porque provee mecanismos de soporte de transacciones, propagación de persistencia, recuperación eficiente de objetos, fijación de niveles de aislamiento y bloqueo para datos sensibles.(29)

1.4.3.2. Enterprise JavaBeans (EJB3)

EJB es el único estándar Java que se ocupa de la lógica de negocio del lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia y seguridad) 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.(30)

1.4.3.3. Java Persistence API (JPA)

Java Persistence API (JPA) proporciona un estándar para gestionar datos relacionales en aplicaciones Java SE o Java EE, de forma que además se simplifique el desarrollo de la persistencia de datos. Es una API de persistencia de POJOs (Plain Old Java Object). Es decir, objetos simples que no heredan ni implementan otras clases (como los EJBs). En su definición, ha combinado ideas y conceptos de los principales frameworks de persistencia, como Hibernate, Toplink y JDO, y de las versiones anteriores de EJB. Todos estos cuentan actualmente con una implementación JPA.

El mapeo objeto-relacional (es decir, la relación entre entidades Java y tablas de la base de datos, queries con nombre) se realiza mediante anotaciones en las propias clases de entidad. No se requieren ficheros descriptores XML. También pueden definirse transacciones como anotaciones JPA.(31)

1.4.4. Java

Java nace como un lenguaje de programación fácil de utilizar, lo que lo convierte en uno de los lenguajes más elaborados y utilizados para la creación de software. Este permite a los desarrolladores:

- Desarrollar software en un Sistema Operativo y ejecutarlo en prácticamente cualquier otro.
- Crear programas para que funcionen en un navegador web y en servicios web.
- Desarrollar aplicaciones para servidores como foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados.
- Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y prácticamente cualquier tipo de dispositivo digital.

A continuación se describen algunas de sus principales características:

Orientado a Objetos: Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.

Distribuido: Java se ha construido con extensas capacidades de interconexión TCP/IP¹³. Existen librerías de rutinas para acceder e interactuar con protocolos como http¹⁴ y ftp¹⁵. La característica de ser distribuido es debido a que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que puedan ejecutarse en varias máquinas.

Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores lo antes posible, en el ciclo de desarrollo.

Interpretado: Se traduce el código fuente a un código intermedio denominado ByteCode, que es interpretado por la Máquina Virtual de Java, lo cual permite que se pueda ejecutar en cualquier sistema operativo.

Seguro: No se permite el acceso ilegal a memoria ya que no se trabaja con punteros. El código Java pasa muchas pruebas antes de ejecutarse en una máquina. El código se pasa a través de un verificador de ByteCodes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal, que falsean punteros, violan derechos de acceso sobre objetos o intentan cambiar el tipo o clase de un objeto.

Dinámico: No conecta todos los módulos que comprende una aplicación hasta el tiempo de ejecución, ya que las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales siempre que mantengan el API anterior.⁽³²⁾

1.4.5. JBoss Server

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro, lo que permite que sea utilizado en cualquier sistema operativo que lo soporte. Implementa todo el paquete de servicios de J2EE y es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, que ofrece una plataforma de alto rendimiento para aplicaciones de e-business. Combina una arquitectura orientada a servicios revolucionaria con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia. ⁽³³⁾

Las características destacadas de JBoss Server incluyen:

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

1.4.6. Proceso Unificado de Desarrollo (RUP)

Proceso Unificado de Desarrollo (**RUP**, por sus siglas en inglés, **Rational Unified Process**) es un proceso de desarrollo de software que proporciona un conjunto de metodologías adaptable al contexto y necesidad de cada organización y junto con el Lenguaje Unificado de Modelado, constituyen la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas. Sus principales características son: (34)

- Iterativo e incremental: A medida que avanza el proceso de desarrollo se producen versiones incrementales, las cuales se acercan cada vez más al producto terminado.
- Guiado por los casos de uso: Los casos de uso son los que indican cómo debe actuar el sistema con el usuario final o con otro sistema para conseguir su objetivo.
- Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño lo cual constituye la arquitectura del producto a desarrollar.
- Centrado en los modelos: Los diagramas ayudan a expresar y entender mejor los elementos que componen un sistema que el lenguaje natural. Su objetivo es minimizar al máximo el uso de descripciones textuales.
- Forma disciplinada para la asignación de las tareas y responsabilidades (quién hace qué, cuándo y cómo).

RUP está compuesto por cuatro fases: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. A su vez está compuesto por nueve Flujos de Trabajo de Ingeniería: Modelamiento del negocio, Requerimientos, Análisis y diseño, Implementación, Pruebas y Despliegue, Administración de cambio y configuración, Administración de proyecto y Entorno. En cada una de sus fases se emplean todos los flujos de trabajo pero con diferente énfasis.(35)

1.4.7. Lenguaje Unificado de Modelado (UML)

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Algunas de las propiedades que han contribuido al auge de UML como lenguaje de modelado estándar son:

- Concurrencia: Es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por OMG (Object Management Group).
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soporta tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas. (36)

1.4.8. Notación para el Modelado de Procesos de Negocio (BPMN)

BPMN (Business Process Management Notation) es un nuevo estándar de modelado de procesos de negocio, donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.(37)

La notación BPMN se compone de varios conjuntos de elementos que abarcan la representación, tanto de los Objetos del flujo y sus conexiones, como de los instrumentos de ayuda que son las Bandas (Swimlanes) y los Artefactos. Los Objetos se dividen en tres categorías: Actividades, Eventos y Compuertas y a su vez, las Actividades se dividen en Tareas y Subprocesos.(38)

1.4.9. Eclipse

Eclipse es un Entorno de Desarrollo Integrado (IDE) de código abierto. Permite crear entornos de desarrollo integrados para distintos lenguajes de programación. Es multiplataforma y permite trabajar con varios proyectos a la vez. Dentro de sus principales características están: editor de texto resaltado de sintaxis, compilación en tiempo real, refactorización. Con esta herramienta se desarrolla lo que el proyecto (Fundación Eclipse) llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

1.4.10. PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (Object-Relational Database Management System (ORDBMS)) libre. No tiene costo asociado, por lo que cualquiera puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente. PostgreSQL presenta alta concurrencia, para esto 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 altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

1.4.11. Herramientas

Con los elementos expuestos se pueden definir las herramientas que conforman el ambiente de desarrollo del Módulo Banco de Sangre para el Sistema de Información Hospitalaria alas HIS.

Se decidió utilizar Visual Paradigm como herramienta CASE (Computer-Aided Software Engineering) basada en BPMN para el modelado del negocio y en UML para el resto del modelado. Esta herramienta permite crear los diagramas que se generen como parte de la documentación. Se puede integrar fácilmente con el IDE Eclipse permitiendo generar código desde diagramas y documentación en varios formatos como HTML, MS Word y PDF. Para el desarrollo de la aplicación Web, se tiene al IDE Eclipse SDK en su versión 3.3.2. Se usará además PostgreSQL como Sistema Gestor de Base de Datos.

En este capítulo se hizo referencia a los principales conceptos relacionados con la gestión de la información en los bancos de sangre. Además, se realizó un estudio sobre los principales sistemas para banco de sangre, existentes a nivel nacional e internacional. También se explicaron las tecnologías definidas para el desarrollo del sistema, así como una breve fundamentación de la utilización de cada una de ellas.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se presentan el flujo actual de los procesos involucrados en el campo de acción, descripción de los procesos que serán objeto de automatización y se modela el negocio propuesto.

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

2.1.1. Atender al donante

Este proceso comienza cuando el donante se presenta en el banco de sangre para realizar una donación. La secretaria(o) crea una historia clínica del donante que lo va a acompañar durante todo el proceso de donación. Al donante se le realiza el examen físico y el hemoterapista revisa los datos entrados en el cuestionario de la historia clínica del donante, rectifica estos datos y adiciona nuevos exámenes realizados. Si el hemoterapista no aprueba la donación, la secretaria(o) registra el motivo de que se descarte esta en el registro de Diferidos y Descartados y desecha la historia clínica del donante. Si la donación es aprobada la secretaria(o) le asigna un serial a la historia clínica del donante.

La Secretaria(o) registra la donación en el Libro de Registro, registra las pruebas a realizarle al donante en la Lista de Serología y crea la Tarjeta de donación que le es entregada al donante. En caso de que la extracción no pueda ser realizada el hemoterapista lo refleja en el Registro de diferidos y descartados, y cancela de esta forma la donación.

El hemoterapista recibe la Tarjeta de Donación que acompaña al donante y realiza la extracción obteniendo una Bolsa de Componente Sanguíneo a la cual le asigna el número de serial expresado en la Tarjeta de Donación. Si la donación realizada es una plasmaféresis, el hemoterapista registra la donación de plasma. Si la donación realizada es una plaquetoaféresis, el hemoterapista registra la donación de plaquetas.

Luego el hemoterapista extrae dos muestras de sangre para realizar las pruebas a la misma. Estas muestras se envían una al laboratorio del banco de sangre y la otra al laboratorio del hospital. El hemoterapista recibe el resultado de los análisis de laboratorio y los registra en la lista de serología que posteriormente le envía a la Secretaria(o). Si los resultados del laboratorio son alterados el hemoterapista descarta la bolsa de componentes sanguíneos y registra el evento en el libro de componentes sanguíneos.

La Secretaria(o) recibe la lista de serología actualizada y la almacena. Luego busca los resultados de las pruebas hechas al donante. Si los resultados son negativos se le entrega al donante su ficha de donación y en caso de que sean positivos se le crea una cita para analizar los resultados de sus pruebas.

2.1.2. Realizar análisis

Este proceso comienza cuando el hemoterapista recibe una muestra a procesar, la cual debe venir acompañada de una Solicitud de análisis si proviene de uno de los servicios autorizados. El hemoterapista procesa los datos de la muestra y actualiza el Registro de Grupos Sanguíneos con el tipo y factor de la sangre. Si la muestra proviene de una donación, registra además en la bolsa de componentes Sanguíneos el tipo, factor de sangre y las pantallas a pacientes femeninas y Orh negativo.

2.1.3. Realizar transfusión

Este proceso comienza cuando el auxiliar de Registro y Control recibe la solicitud de transfusión proveniente de un servicio del hospital. El hemoterapista auxiliar de Registro y Control valida los datos de la Solicitud de Transfusión. Si la solicitud es incorrecta termina el proceso y en caso de que esté correcta, la enfermera(o) consulta los datos de la solicitud de transfusión con los que localiza al paciente y le extrae una muestra de sangre. El hemoterapista recibe las muestras de sangre obtenidas por la enfermera(o) y le realiza a estas las pruebas con el fin de determinar el tipo de sangre y factor.

El hemoterapista consulta el registro de componentes sanguíneos y selecciona una bolsa de componentes sanguíneos del mismo grupo y factor (obtenidos de la solicitud de transfusión) que el paciente para realizar con esta la prueba de compatibilidad. Realiza esta prueba de compatibilidad utilizando la muestra de sangre extraída al paciente y la bolsa de componentes sanguíneos previamente seleccionada.

En caso de que no sea compatible, el hemoterapista realiza pruebas especiales para localizar una bolsa de componente sanguíneo compatible con el receptor. Si las pruebas especiales arrojan resultados positivos, el hemoterapista crea la Boleta de Transfusión Preparada, la anexa a una bolsa de componente sanguíneo y registra este hecho en el libro de Transfusiones Preparadas y Entregadas. El hemoterapista en caso de no tener un componente sanguíneo determinado y estar en presencia de una urgencia, desglosa una bolsa de componentes sanguíneos asignada previamente a otro paciente.

Al llegar la enfermera al banco de sangre a retirar la bolsa de componente sanguíneo para la realización de la transfusión, el hemoterapista le entrega la bolsa de componente sanguíneo y lo registra en el libro de Transfusiones Preparadas y Entregadas, posteriormente procede a rebajar del Registro de Componentes Sanguíneos la bolsa de componente sanguíneo entregada.

2.2. Objeto de automatización

Se desean automatizar todos los procesos necesarios para un óptimo funcionamiento en el banco de sangre:

Atención al donante: Se desean automatizar todas las actividades que se encuentran enmarcadas en el proceso de la atención al donante: llenar la historia clínica del donante, realizar el examen físico, descartar la historia clínica del donante, registrar la donación, actualizar serial de plasmaféresis transfusional, actualizar cuaderno de plaquetoaféresis, enviar solicitud de análisis al laboratorio, recibir resultado de exámenes, almacenar estos resultados.

Realizar análisis: Se desean automatizar todas las actividades que se encuentran enmarcadas en el proceso de la realización de análisis. Registrar el grupo y factor sanguíneo arrojados del resultado de procesar las solicitudes de examen de tipiaje recepcionadas.

Realizar transfusión: Se desean automatizar todas las actividades que se encuentran enmarcadas en el proceso de la realización de transfusiones: recibir la solicitud de transfusión, validar esta, realizar pruebas de compatibilidad, asignar bolsas de componentes, realizar pruebas especiales, desglosar bolsas de componentes.

2.3. Información que se maneja

En el banco de sangre, se hace uso de un conjunto de documentos importantes, que son vitales para el correcto funcionamiento del mismo. Algunos de estos, se utilizaron como apoyo para llevar a cabo el modelado de los procesos que fueron identificados como objeto de automatización.

Historia clínica del donante: Recoge toda la información referente al donante: archiva un identificador numérico, los datos generales del donante (Nombre(s), Apellidos, Sexo, Lugar de Nacimiento, Fecha de Nacimiento, Edad, No. Cédula de Identidad, Profesión Ocupación, Dirección, Estado, Teléfono, Dirección del Centro de Trabajo, Población del Centro de Trabajo, Estado del Centro de Trabajo,

Teléfono del Centro de Trabajo), los datos clínicos (Peso, Hemoglobina, Tensión Arterial, Pulso, Temperatura, Aceptación, Causa de Diferido-Descartado, Responsable de la selección del Donante). Contiene además una sección de las Donaciones y una Sección de Datos de Laboratorio (Serología).

Cuaderno de Plaquetoféresis: Recoge toda la información referente a un cuaderno de plaquetoféresis. Contiene fecha de la donación, Nombre y Apellidos, Peso, Talla, valor del Hematocrito del donante, Grupo y Factor sanguíneo del donante, Serial de la donación, Nombre y Apellidos del paciente al que se dirige la donación, Recuento Plaquetario del donante previo a la donación, Solución Anticoagulante utilizada, Flujo de extracción, Equipo utilizado, Tiempo de duración del Procedimiento.

Ficha de Donación: Recoge toda la información referente a una ficha de donación. Archiva: nombre del donante, cédula del donante, grupo sanguíneo del donante, resultados de las serologías hechas al donante, y la fecha en que se entrega la ficha al donante.

Serial de plasma transfusional: Recoge datos como: Fecha en que se realiza la donación, Nombre y Apellidos del donante, Número de cédula de identidad del donante, Grupo y factor sanguíneo del donante y Volumen de plasma extraído (ml).

Solicitud de Examen: Recoge toda la información referente a una solicitud. Contiene: laboratorio al que se dirige la solicitud, fecha en que se solicita el examen, funcionario que solicita el examen, número de registro del ministerio de salud del médico que solicita el examen, número de registro del colegio de medicina del médico que solicita el examen, número de cédula del médico que solicita el examen, servicio en el que se encuentra el paciente, tipo de muestra, examen solicitado, resultado, observaciones, fecha de salida, número de la muestra.

Boleta de Transfusión Preparada: Recoge toda la información referente a una boleta de una transfusión preparada. Recoge los siguientes datos: Donación, Grupo Sanguíneo, Factor Rh, Componente, Cantidad, Serología. Además contiene una sección de Datos del Receptor (Nombre, Grupo Sanguíneo, Factor Rh Compatibilidad, Mayor, Fecha, Ubicación, Observaciones, Preparada Por, Médico que administra).

Transfusiones Preparadas y Entregadas: Recoge toda la información referente al cuaderno de transfusiones preparadas y entregadas. Contiene la fecha en que se prepara la transfusión, serial de la

donación de la que se obtuvo la bolsa de componentes, hora en que se prepara la transfusión, nombre y apellidos del paciente, lugar físico donde se encuentra el paciente, donaciones anteriores que tiene el paciente, grupo de sangre de la bolsa de componentes, factor de la Bolsa de Componentes.

Solicitud de Transfusión: Recoge toda la información referente a una solicitud de transfusión. Contiene una sección de datos del paciente (Nombre y apellidos del paciente, número de historia clínica del paciente, Edad, Diagnóstico, Ubicación Hemoglobina, Tipo y factor de la sangre del paciente), una sección de datos de la Transfusión (Carácter de la transfusión, Transfusión anterior, Fecha de la última transfusión, Reacciones, Tipo de reacción, Tratamiento, Acto operatorio, Tipo de intervención).

Contiene además una sección de datos de los funcionarios (Nombre del Médico, Número de registro del ministerio de salud, Número del colegio de medicina, Cédula, Recibido, Muestra tomada por Aceptación, Fecha, Hora, Observaciones) y una sección para el personal de banco de sangre (Grupo-Rh del receptor, Serial de las bolsas de componentes, Grupo-Rh, Segmento, Componente, Fecha, Hora, Pruebas Cruzadas, Responsable).

2.4. Modelo de Negocio

La disciplina de Modelado de Negocio es la primera que propone RUP dentro del ciclo de desarrollo de un software, tiene su mayor peso durante la fase de inicio debido a que permite conocer los procesos existentes actuales de cualquier entidad o empresa para la que se vaya a desarrollar el sistema. Los objetivos del modelo del negocio son:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

Para lograr los propósitos anteriormente mencionados, el proceso de modelamiento permite obtener una visión de la organización que facilite definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. (35)

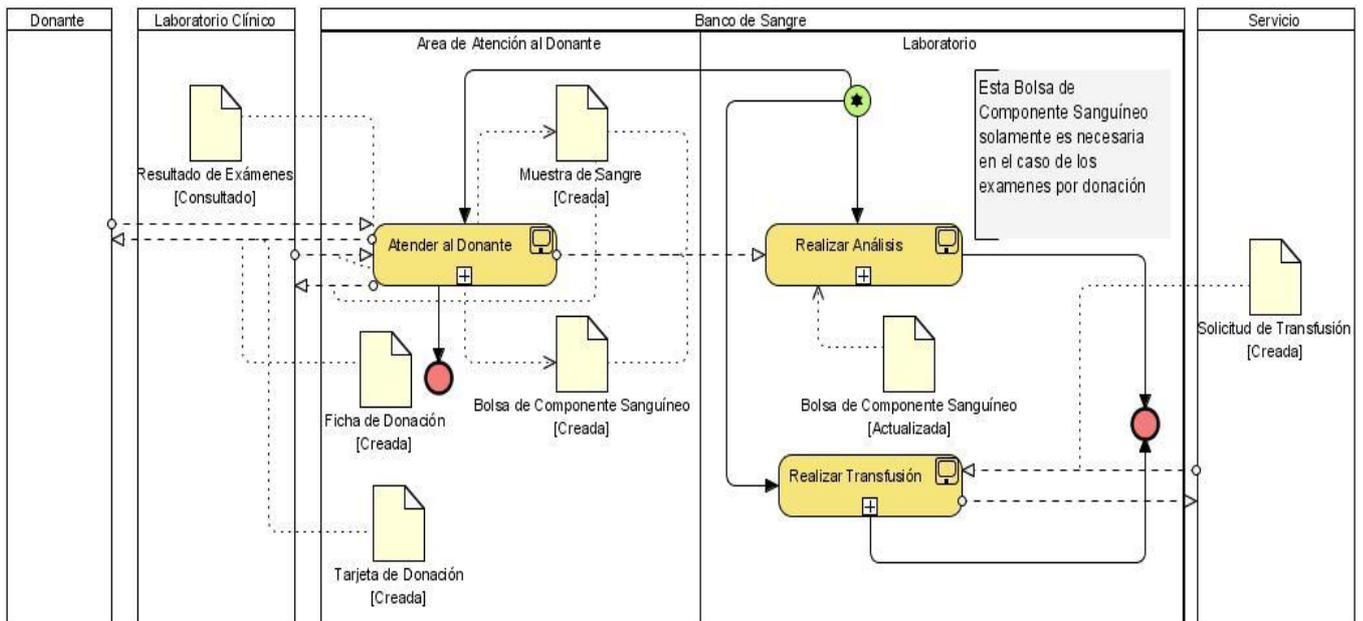


Figura 2.1. Diagrama de procesos del negocio.

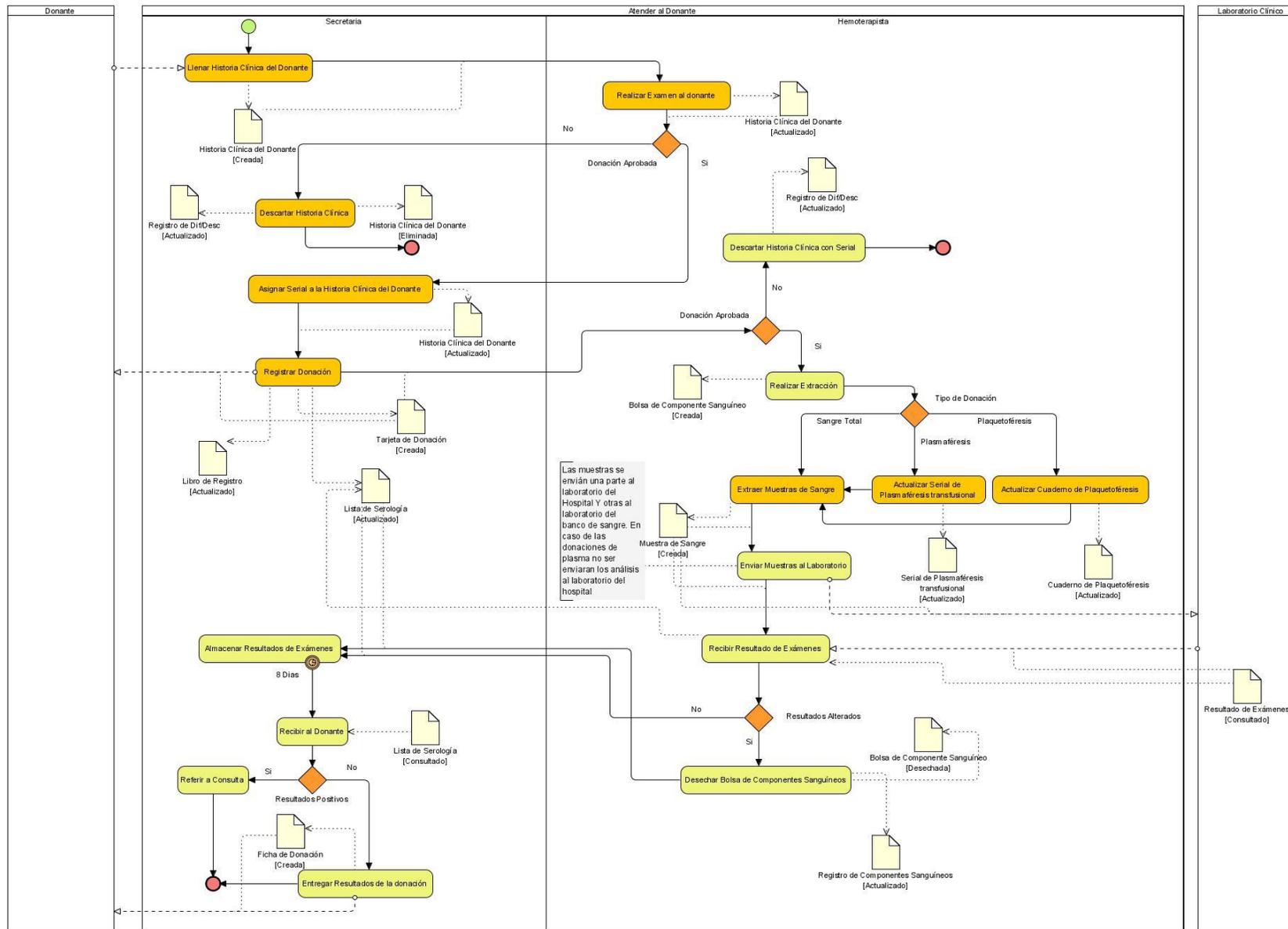


Figura 2.2. Diagrama de actividades del proceso Atender Donante.

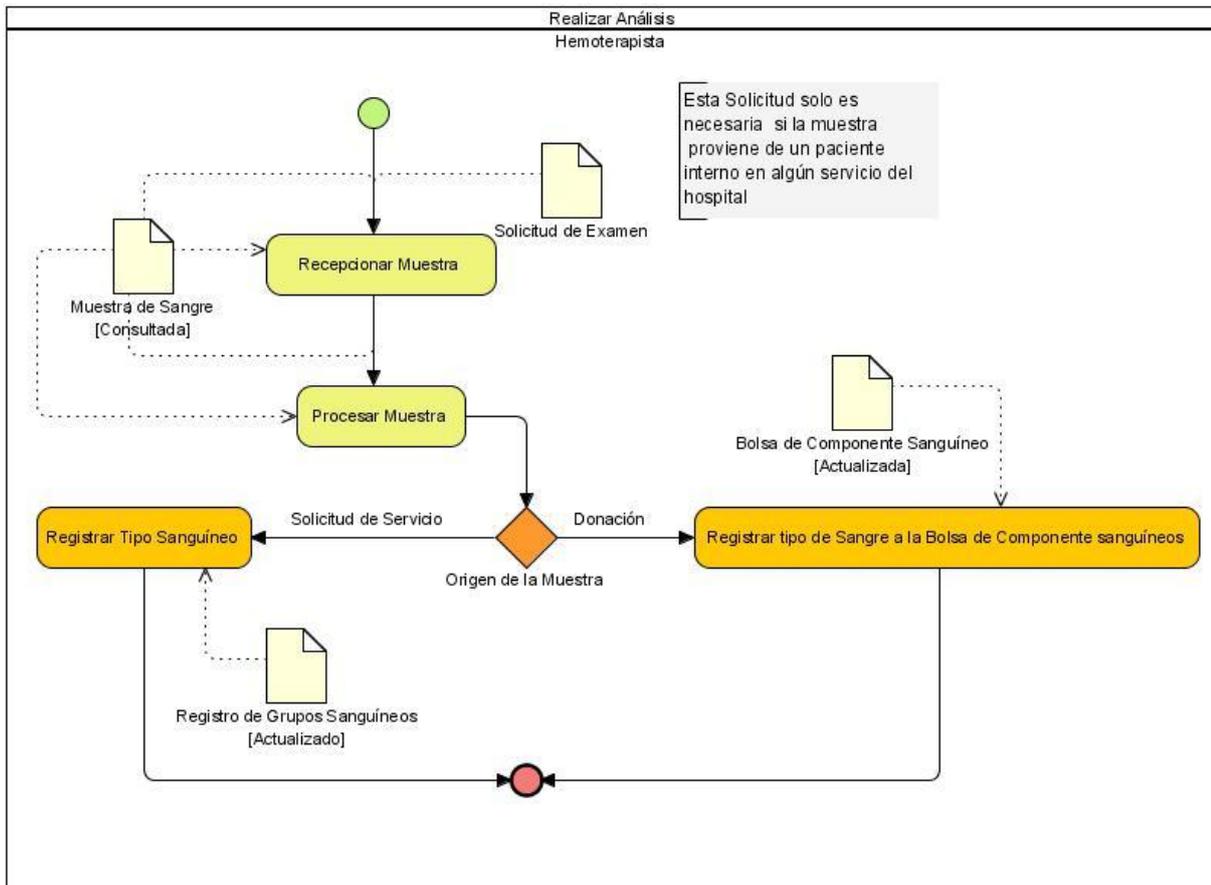


Figura 2.3. Diagrama de actividades del proceso Realizar Análisis.

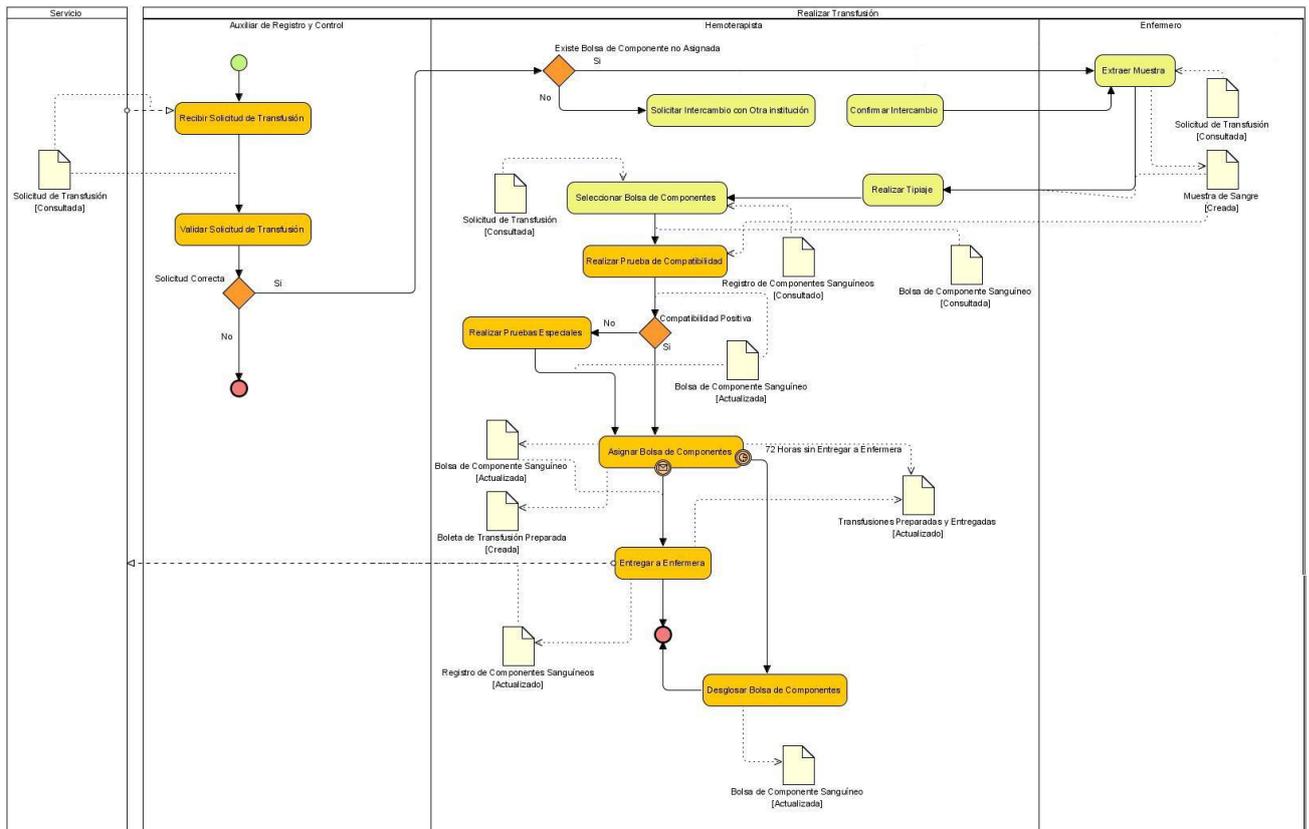


Figura 2.4. Diagrama de actividades del proceso Realizar Transfusión.

2.4.1. Actor del negocio

Es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. (39)

Rol	Función
Donante	Es el principal beneficiado de todas las actividades relacionadas con las donaciones en el banco de sangre.
Paciente	Es el principal beneficiado de todas las actividades relacionadas con las transfusiones y las realizaciones de análisis en el banco de sangre.

Tabla 2.1. Actores del negocio.

2.4.2. Trabajador del Negocio

Es quién define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Rol	Función
Secretaria(o)	Es la persona encargada de crear la historia clínica del donante y asignarle un serial a esta, mantener actualizado el libro de registro y la lista de serología, además debe entregar los resultados de los exámenes a los donantes y en caso de ser necesario referirlos a consulta de serología.
Enfermera(o)	Es la persona encargada de realizarle el examen físico al donante (con la posibilidad de descartar su historia clínica del donante en caso de ser necesario), de realizar la extracción de la sangre y la recolección de las muestras para analizar, también es el encargado de recepcionar los resultados de estos análisis. También es quien extrae las muestras de sangre al paciente que va a ser transfundido.
Hemoterapista	Es la persona encargada de recepcionar las muestras, procesarlas y registrar el resultado de los exámenes. Selecciona las bolsas de componentes sanguíneos que van a ser transfundidas, realizar con estas las pruebas de compatibilidad, asignarlas a un paciente en específico y entregarlas a la enfermera del servicio. También tiene la potestad de desglosar la bolsa de componentes sanguíneos en casos de emergencia o al no ser retiradas por la enfermera del servicio.
Auxiliar de Registro y Control	Es la persona encargada de recepcionar y validar las solicitudes de transfusión.

Tabla 2.2. Trabajadores del negocio.

2.5. Propuesta del Sistema

2.5.1. Especificación de los requerimientos de software.

Con el objetivo de construir un producto que responda a las necesidades actuales en las instituciones hospitalarias se realiza la captura de requisitos. Para realizar esta tarea, se ha tomado como punto de partida las deficiencias que se observa en los procesos realizados en los bancos de sangre.

2.5.1.1. Requerimientos funcionales.

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales:(40)

2.5.1.1.1. Atender al donante

RF1 Crear historia clínica del donante.

1.1. Validar datos incompletos

RF2 Seleccionar historia clínica del donante.

2.1 Realizar una búsqueda elemental dado criterios.

2.2 Realizar una búsqueda avanzada dado criterios.

2.3 Mostrar un mensaje cuando no se encuentre información que cumpla con los criterios de búsqueda.

2.4 Ordenar el resultado ascendente o descendientemente por un atributo.

2.5 Navegar por el resultado.

RF3 Registrar datos clínicos del donante.

3.1 Validar datos incompletos.

3.2 Validar datos incorrectos.

RF4 Descartar historia clínica del donante.

4.1 Validar datos incompletos.

RF5 Crear bolsa de componentes sanguíneos.

5.1 Validar datos incompletos.

RF6 Crear registro del cuaderno de plaquetoaféresis.

6.1 Validar datos incompletos.

6.2 Validar datos incorrectos.

RF7 Asignar serial a historia clínica del donante.

7.1 Validar datos incompletos.

7.2 Validar datos incorrectos.

RF8 Crear muestra de sangre a partir de la historia clínica del donante.

8.1 Validar datos incompletos.

RF9 Generar reporte de ficha de donación.

9.1 Generar el reporte a partir de una búsqueda.

9.2 Mostrar un mensaje cuando no se encuentre información que cumpla con los criterios de búsqueda.

9.3 Imprimir el reporte.

RF10 Crear registro del serial de plasmaféresis.

10.1 Validar datos incompletos.

10.2 Validar datos incorrectos.

RF11 Generar reporte de listado de donantes.

11.1 Generar el reporte dado criterios elementales de búsqueda.

2.5.1.1.2. Realizar análisis

RF12 Asociar grupo y factor a bolsa de componentes sanguíneos.

RF13 Seleccionar solicitud de tipiaje.

13.1 Realizar una búsqueda elemental dado criterios.

13.2 Realizar una búsqueda avanzada dado criterios.

13.3 Mostrar un mensaje cuando no se encuentre información que cumpla con los criterios de búsqueda.

13.4 Ordenar el resultado ascendente o descendientemente por un atributo.

13.5 Navegar por el resultado.

RF14 Registrar resultado de exámenes.

14.1 Validar datos incompletos.

14.2 Validar datos incorrectos.

RF15 Generar reporte de grupo y factor de bolsa de componentes.

15.1 Generar el reporte dado criterios elementales de búsqueda.

15.2 Generar el reporte a partir de una búsqueda.

15.3 Mostrar un mensaje cuando no se encuentre información que cumpla con los criterios de búsqueda.

15.4 Ordenar el reporte ascendente o descendientemente por un atributo.

2.5.1.1.3. Realizar transfusión

RF16 Crear boleta de transfusión preparada.

16.1 Validar datos incompletos.

16.2 Validar datos incorrectos.

RF17 Crear registro en el libro de transfusiones preparadas y entregadas.

17.1 Validar datos incompletos.

17.2 Validar datos incorrectos.

RF18 Aceptar solicitud o rechazar datos incoherentes.

18.1 Validar datos incompletos.

RF19 Entregar bolsa de componentes a enfermera.

19.1 Validar datos incompletos.

19.2 Validar datos incorrectos.

RF20 Desglosar bolsa de componentes.

RF21 Crear muestras de sangre a partir de solicitud de transfusión.

21.1 Validar datos incompletos.

RF22 Crear estudio inmunohematológico.

22.1 Validar datos incompletos.

22.2 Validar datos incorrectos.

RF23 Consultar solicitud de transfusión.

23.1 Ver datos de la entidad.

RF24 Generar reporte de Transfusiones preparadas y entregadas.

24.1 Generar el reporte dado criterios elementales de búsqueda.

2.5.1.2. Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Existen múltiples categorías para clasificarlos, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros

Usabilidad.

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

Usuarios normales: 20 días.

Usuarios avanzados: 30 días.

Seguridad.

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

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

El sistema proporcionará un registro de actividades (log) de cada usuario en el sistema.

Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD.

El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

Rendimiento.

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

Soporte.

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

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

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

Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema. Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

Hardware. Estaciones de trabajo.

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

Hardware. Servidores.

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables.

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

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

Software.

El servidor debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java virtual machine, JBoss AS y PostgreSQL).

Los clientes deberán disponer de un navegador web, estos pueden ser IE 7 o superior, Opera 9 superior, Google chrome 1 o superior y Firefox 2 o superior.

Restricciones de diseño.

La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio.

La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario.

La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación de éste.

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

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

Interfaz. Interfaces de usuario.

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

La interfaz contará con accesos directos y menús desplegable que faciliten y aceleren su utilización.

La entrada de datos incorrecta será detectada claramente e informada al usuario.

Todos los textos y mensajes en pantalla aparecerán en idioma español.

Interfaz. Interfaces de comunicación.

El sistema usará el formato estándar WSDL para la descripción de los servicios web.

El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos.

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

Portabilidad.

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

2.6. Modelo de Casos de Uso del Sistema

El modelo de Casos de Uso del sistema es utilizado para representar los casos de uso y actores con las relaciones entre sí.

2.6.1. Definición de actores

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

Actores del Sistema	Justificación
Funcionario del banco de sangre	Actor genérico que representa cualquier usuario que pertenezca al área en cuestión (banco de sangre).
Secretaria en banco de sangre	Actor que representa un funcionario no médico del área en cuestión (banco de sangre) que participa en el proceso de atención al donante.
Hemoterapista	Actor que representa a los enfermeros especializados que laboran en el área en cuestión (banco de sangre).
Enfermero en banco de sangre	Actor que representa a un enfermero no especializado.

Tabla 2.3. Actores del sistema.

2.6.2. Vista global de actores

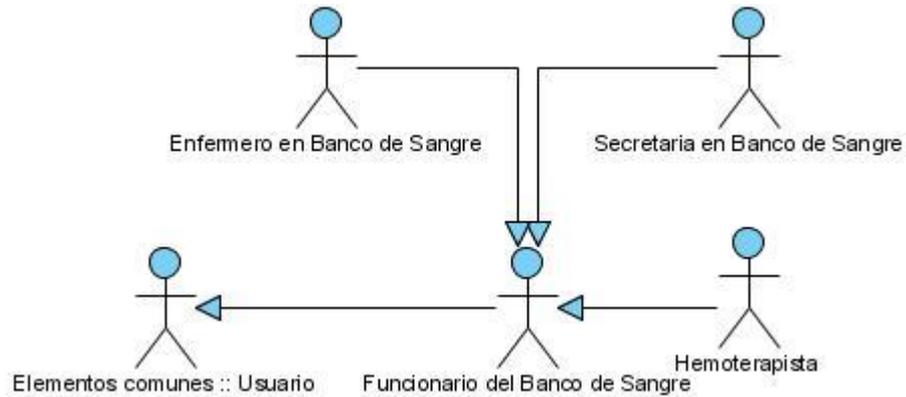


Figura 2.5. Vista global de actores del sistema.

2.6.3. Diagrama de Casos de Uso

2.6.3.1. Atender al donante

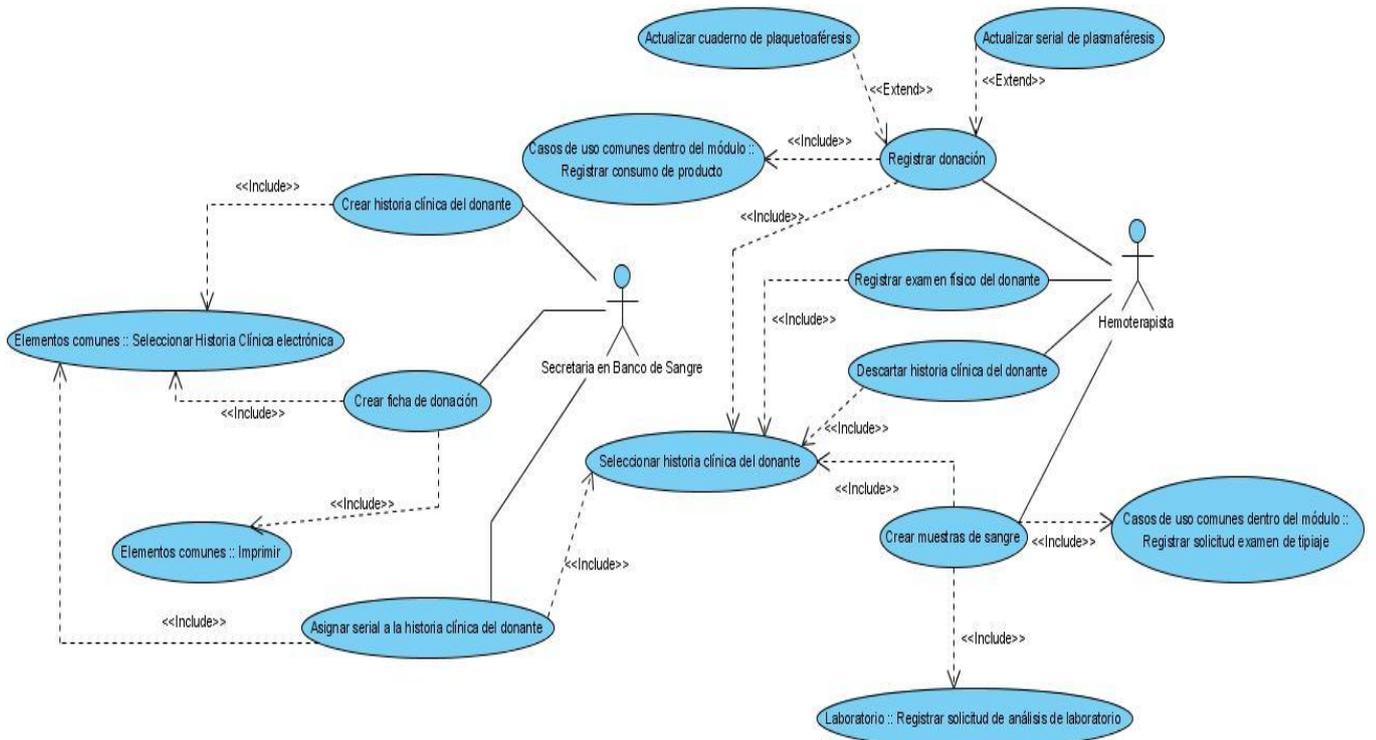


Figura 2.6. Diagramas de casos de uso del proceso Atender Donante.

2.6.3.2. Realizar análisis

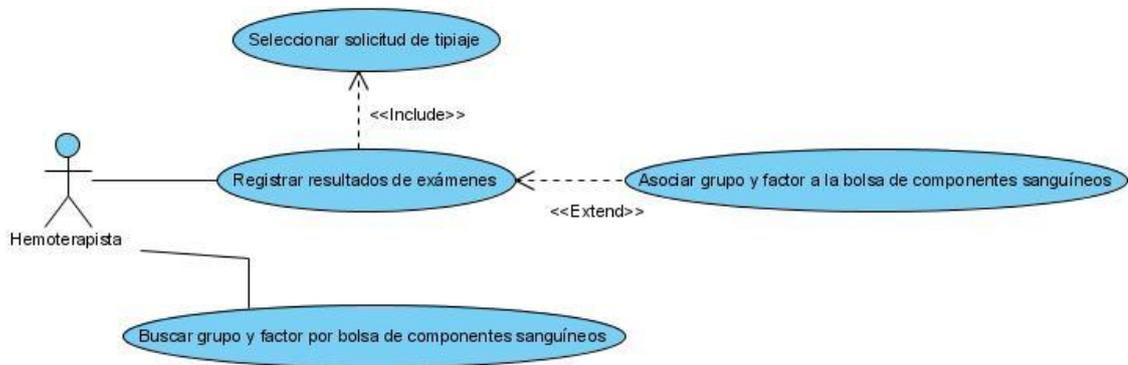


Figura 2.7. Diagramas de casos de uso del proceso Realizar Análisis.

2.6.3.3. Realizar transfusión

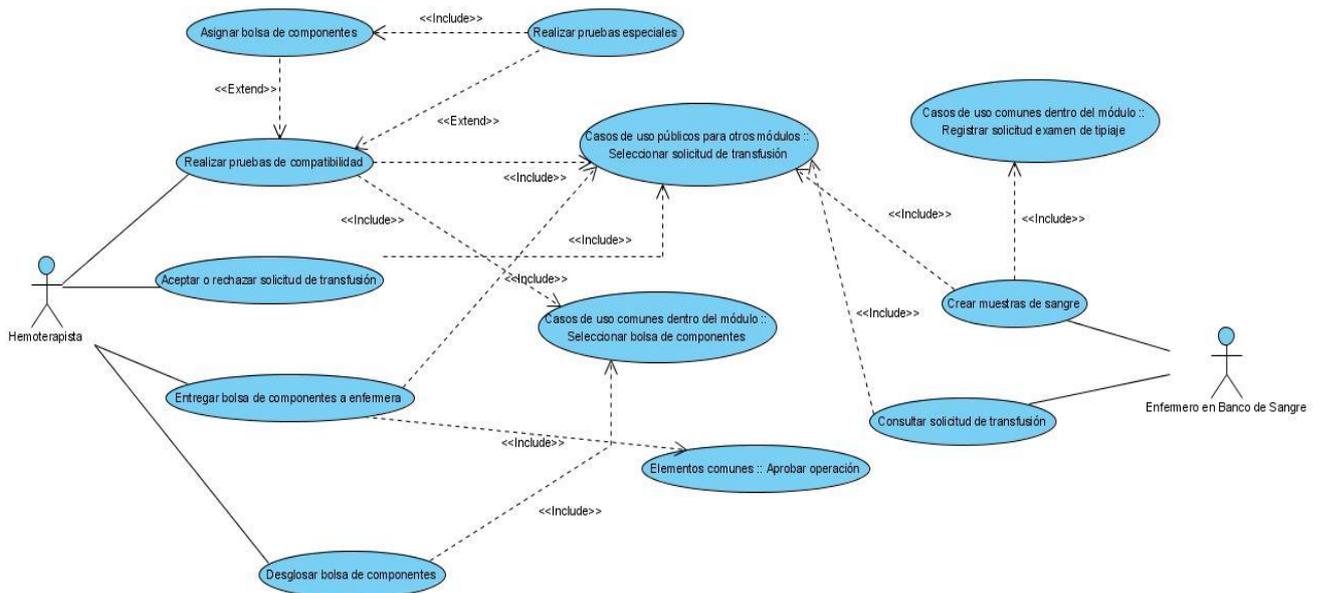


Figura 2.8. Diagramas de casos de uso del proceso Realizar Transfusión.

2.6.4. Descripción Textual de los Casos de Uso

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Proporcionan un medio para que los

desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema. Además ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo. Un caso de uso describe un proceso de principio a fin, es decir, una secuencia de eventos, las acciones y las transacciones que se requieren para realizarlo y produce un resultado observable para un actor.

2.6.4.1. Atender al donante

CUS1: Crear historia clínica del donante.	
Propósito:	Permitir crear una historia clínica de donante.
Actores:	Secretaria en banco de sangre.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear historia clínica del donante y el sistema brinda la posibilidad de introducir los datos para crearla. El actor introduce los datos de la historia clínica del donante. el sistema crea la historia clínica del donante. el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se modificó una historia clínica del donante por el actor.
Referencias:	RF1

Tabla 2.4. Descripción del Caso de Uso del sistema Crear historia clínica del donante.

CUS2: Seleccionar historia clínica del donante.	
Propósito:	Permitir seleccionar una historia clínica de donante.
Actores:	Funcionario del banco de sangre.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Seleccionar historia clínica del donante. El sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la historia clínica del donante. El actor introduce los datos que considera como criterios para buscar la historia clínica del donante. El sistema busca y muestra las historias clínicas del donante que cumplen con los criterios de búsqueda. El actor selecciona la historia clínica del donante. El sistema carga en la vista anterior cada una de las historias clínicas del donante seleccionadas. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se seleccionó la historia clínica del donante dado criterios.
Referencias:	RF2

Tabla 2.5. Descripción del Caso de Uso del sistema Seleccionar historia clínica del donante.

CUS3: Registrar datos clínicos del donante.	
Propósito:	Permitir registrar los datos clínicos del donante en la historia clínica del donante.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Registrar datos clínicos del donante y selecciona una historia clínica del donante. El sistema muestra los datos de la historia clínica del donante y brinda la posibilidad de cambiar datos o de introducir nuevos, el actor introduce los datos que necesita. El sistema actualiza los datos de la historia clínica del donante. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se modificó una historia clínica del donante por el actor.
Referencias:	RF3

Tabla 2.6. Descripción del Caso de Uso del sistema Registrar datos clínicos del donante.

CUS4: Descartar historia clínica del donante.	
Propósito:	Permitir descartar una historia clínica del donante.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor selecciona una historia clínica del donante y accede a la opción Descartar historia clínica del donante, el sistema muestra los datos de la historia clínica del donante y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. El actor modifica los datos que necesita, el sistema actualiza los datos de la historia clínica del donante. El caso de uso termina.
Precondición:	Para modificar los datos de una historia clínica del donante, esta debe haber sido seleccionada.
Poscondición:	Se modificó una historia clínica del donante por el actor.
Referencias:	RF4

Tabla 2.7. Descripción del Caso de Uso del sistema Descartar historia clínica del donante.

CUS5: Registrar donación.	
Propósito:	Permitir registrar una donación.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Registrar Donación, el sistema brinda la posibilidad de introducir los datos para crear la Bolsa de componentes sanguíneos. El actor introduce los datos de la Bolsa de componentes sanguíneos, el sistema crea la Bolsa de componentes sanguíneos. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se creó una Bolsa de componentes sanguíneos.
Referencias:	RF5

Tabla 2.8. Descripción del Caso de Uso del sistema Registrar donación.

CUS6: Actualizar cuaderno de plaquetoaféresis.	
Propósito:	Permitir actualizar el cuaderno de plaquetoaféresis.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Actualizar Cuaderno de Plaquetoaféresis, el sistema brinda la posibilidad de introducir los datos para crear una fila en el cuaderno de plaquetoaféresis, el actor introduce los datos de la fila en el cuaderno de plaquetoaféresis, el sistema crea la fila en el cuaderno de plaquetoaféresis, el caso de uso termina.
Precondición:	Que se haya registrado la donación.
Poscondición:	Se creó una fila del cuaderno de plaquetoaféresis.
Referencias:	RF6

Tabla 2.9. Descripción del Caso de Uso del sistema Actualizar cuaderno de plaquetoaféresis.

CUS7: Asignar serial a la historia clínica del donante.	
Propósito:	Permitir asignar un serial a la historia clínica del donante.
Actores:	Secretaria en banco de sangre.
Resumen:	El caso de uso inicia cuando el actor selecciona una historia clínica del donante y accede a la opción Asignar Serial a la historia clínica del donante. El sistema muestra los datos de la historia clínica del donante y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos de la historia clínica del donante. El caso de uso termina.
Precondición:	Para modificar los datos de una historia clínica del donante, esta debe haber sido seleccionada.
Poscondición:	Se modificó una historia clínica del donante por el actor.
Referencias:	RF7

Tabla 2.10. Descripción del Caso de Uso del sistema Asignar serial a la historia clínica del donante.

CUS8: Crear muestra de sangre.	
Propósito:	Permitir crear una muestra de sangre.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear muestra de sangre. El sistema brinda la posibilidad de introducir los datos para crear la Muestra de Sangre, el actor introduce los datos de la muestra de sangre, el sistema crea la muestra de sangre. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se creó una muestra de sangre.
Referencias:	RF8

Tabla 2.11. Descripción del Caso de Uso del sistema Crear muestra de sangre.

CUS9: Crear ficha de donación.	
Propósito:	Permitir crear ficha de donación.
Actores:	Secretaria en banco de sangre.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear ficha de donación, el sistema brinda la posibilidad de introducir los criterios para generar el reporte. El actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando las entidades historia clínica del paciente, Registro de Grupos Sanguíneos, Informe de Resultado de Laboratorio, genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema imprime los datos del reporte. El caso de uso termina.
Precondición:	Se deben haber terminado los exámenes para este donante.
Poscondición:	Se generó un reporte de ficha de donación dado criterios.
Referencias:	RF9

Tabla 2.12. Descripción del Caso de Uso del sistema Crear ficha de donación.

CUS10: Actualizar serial de plasmaféresis.	
Propósito:	Permitir actualizar serial de plasmaféresis.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Actualizar serial de plasmaféresis, el sistema brinda la posibilidad de introducir los datos para crear un nuevo registro en el Serial de plasma transfusional, el actor introduce los datos del registro en el Serial de plasma transfusional, el sistema crea el registro en el Serial de plasma transfusional, el caso de uso termina.
Precondición:	Para actualizar el serial de plasmaféresis debe haberse una registrado la donación de plasma anteriormente.
Poscondición:	Se creó un nuevo registro en el Serial de plasma transfusional.
Referencias:	RF10

Tabla 2.13. Descripción del Caso de Uso del sistema Actualizar serial de plasmaféresis.

CUS11: Generar Reporte de listado de donantes.	
Propósito:	Permitir generar un reporte de listado de donantes.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de listado de donantes, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando la entidad historia clínica del donante genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema imprime los datos del reporte, el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte de listado de donantes dado criterios.
Referencias:	RF11

Tabla 2.14. Descripción del Caso de Uso del sistema Generar Reporte de listado de donantes.

2.6.4.2. Realizar análisis.

CUS12: Asociar grupo y factor a la bolsa de componentes sanguíneos.	
Propósito:	Permitir asociar grupo y factor a la bolsa de componentes sanguíneos.
Actores:	Cualquier caso de uso que necesite asociar grupo y factor a una bolsa de componentes.
Resumen:	Permitir a otros casos de uso asociar el grupo y factor a una bolsa de componentes sanguíneos.
Precondición:	No existe.
Poscondición:	Se actualizó una bolsa de componentes sanguíneos.
Referencias:	RF12

Tabla 2.15. Descripción del Caso de Uso del sistema Asociar grupo y factor a la bolsa de componentes sanguíneos.

CUS13: Seleccionar solicitud de tipiaje.	
Propósito:	Permitir seleccionar una solicitud de tipiaje.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Seleccionar Solicitud de tipiaje, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la Solicitud de tipiaje deseada, el actor introduce los datos que considera como criterios para buscar la Solicitud de tipiaje, el sistema busca y muestra las Solicitud de tipiaje que cumplen con los criterios de búsqueda, el actor selecciona la Solicitud de tipiaje deseada, el sistema carga en la vista anterior cada una de las Solicitud de tipiaje seleccionadas, el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se seleccionó Solicitud de examen dado criterios.
Referencias:	RF13

Tabla 2.16. Descripción del Caso de Uso del sistema Seleccionar solicitud de tipiaje.

CUS14: Registrar resultados de exámenes.	
Propósito:	Permitir registrar los resultados de exámenes.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor selecciona una Solicitud de examen y accede a la opción Registrar resultados de exámenes, el sistema muestra los datos de la Solicitud de examen y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos de la Solicitud de examen, el caso de uso termina.
Precondición:	Para registrar los resultados en una Solicitud de examen, esta debe haber sido seleccionada.
Poscondición:	Se modificó una Solicitud de examen por el actor.
Referencias:	RF14

Tabla 2.17. Descripción del Caso de Uso del sistema Registrar resultados de exámenes.

CUS15: Buscar grupo y factor por bolsa de componentes sanguíneos.	
Propósito:	Permitir buscar el grupo y factor por bolsa de componentes sanguíneos.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Buscar Grupo y Factor por bolsa de componentes sanguíneos, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando las entidades Bolsa de componentes sanguíneos, Muestra de Sangre y Solicitud de tipaje, genera el reporte de información. El actor selecciona la opción de imprimir el reporte, el sistema imprime los datos del reporte, el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte de Grupo y Factor por bolsa de componentes dado criterios.
Referencias:	RF15

Tabla 2.18. Descripción del Caso de Uso del sistema Buscar grupo y factor por bolsa de componentes sanguíneos.

2.6.4.3. Realizar transfusión.

CUS16: Asignar bolsa de componentes.	
Propósito:	Permitir asignar bolsa de componentes.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Asignar Bolsa de componentes, el sistema brinda la posibilidad de introducir los datos para crear la Boleta de transfusión preparada, el actor introduce los datos de la Boleta de transfusión preparada, el sistema crea la Boleta de transfusión preparada, el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se creó una Boleta de transfusión preparada.
Referencias:	RF16

Tabla 2.19. Descripción del Caso de Uso del sistema Asignar bolsa de componentes.

CUS17: Realizar pruebas de compatibilidad.	
Propósito:	Permitir realizar las pruebas de compatibilidad.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Realizar Pruebas de compatibilidad, el sistema brinda la posibilidad de introducir los datos para crear el registro en el Libro de Transfusiones preparadas y entregadas, el actor introduce los datos del registro en el Libro de Transfusiones preparadas y entregadas, el sistema crea el registro en el Libro de Transfusiones preparadas y entregadas, el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se creó un registro en el Libro de Transfusiones preparadas y entregadas.
Referencias:	RF17

Tabla 2.20. Descripción del Caso de Uso del sistema Buscar grupo y factor por bolsa de componentes sanguíneos.

CUS19: Entregar bolsa de componentes a enfermera.	
Propósito:	Permitir entregar bolsa de componentes a enfermera.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor selecciona una Solicitud de transfusión y accede a la opción Entregar Bolsa de componentes a Enfermera, el sistema muestra los datos del registro en el Libro de Transfusiones preparadas y entregadas y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos del registro en el Libro de Transfusiones preparadas y entregadas, el caso de uso termina.
Precondición:	Para modificar los datos de un registro en el Libro de Transfusiones preparadas y entregadas, debe haber sido seleccionada la solicitud de la transfusión.
Poscondición:	Se modificó un registro en el Libro de Transfusiones preparadas y entregadas por el actor.
Referencias:	RF19

Tabla 2.21. Descripción del Caso de Uso del sistema Entregar bolsa de componentes a enfermera.

CUS20: Desglosar bolsa de componentes.	
Propósito:	Permitir desglosar una bolsa de componentes.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor selecciona una Bolsa de componentes y accede a la opción Desglosar Bolsa de componentes, el sistema elimina la Boleta de Transfusión preparada asociada a la Bolsa de componentes seleccionada, el caso de uso termina.
Precondición:	Para desglosar una bolsa, esta debe haber sido seleccionada.
Poscondición:	Se eliminó una Boleta de transfusión preparada por el actor.
Referencias:	RF20

Tabla 2.22. Descripción del Caso de Uso del sistema Desglosar bolsa de componentes.

CUS21: Crear muestra de sangre.	
Propósito:	Permitir crear una muestra de sangre.
Actores:	Enfermero en banco de sangre.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear muestra de sangre, el sistema brinda la posibilidad de introducir los datos para crear la muestra de sangre, el actor introduce los datos de la muestra de sangre, el sistema crea la Muestra de sangre, el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se creó una muestra de sangre.
Referencias:	RF21

Tabla 2.23. Descripción del Caso de Uso del sistema Crear muestra de sangre.

CUS22: Consultar solicitud de transfusión.	
Propósito:	Permitir consultar una solicitud de transfusión.
Actores:	Enfermera en banco de sangre.
Resumen:	El caso de uso inicia cuando el actor accede a la opción de Consultar solicitud de transfusión, el sistema muestra los datos de la Solicitud de transfusión, el caso de uso termina.
Precondición:	Para ver los datos de la solicitud de transfusión, esta debe estar seleccionada.
Poscondición:	Se vieron los datos de la solicitud de transfusión.
Referencias:	RF22

Tabla 2.24. Descripción del Caso de Uso del sistema Consultar solicitud de transfusión.

CUS23: Realizar pruebas especiales.	
Propósito:	Permitir realizar pruebas especiales.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Realizar pruebas especiales, el sistema brinda la posibilidad de introducir los datos para crear el Estudio Inmunohematológico, el actor introduce los datos del Estudio Inmunohematológico, el sistema crea el Estudio Inmunohematológico, el caso de uso termina.
Precondición:	No existe.
Poscondición:	Se creó un Estudio Inmunohematológico.
Referencias:	RF23

Tabla 2.25. Descripción del Caso de Uso del sistema Realizar pruebas especiales.

CUS24: Generar Reporte de transfusiones preparadas y entregadas.	
Propósito:	Permitir generar el Reporte de transfusiones preparadas y entregadas.
Actores:	Hemoterapista.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de transfusiones preparadas y entregadas, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando la entidad Transfusiones preparadas y entregadas genera el reporte de información.
Precondición:	No existe.
Poscondición:	Se generó un reporte de transfusiones preparadas y entregadas dados criterios.
Referencias:	RF24

Tabla 2.26. Descripción del Caso de Uso del sistema Generar Reporte de transfusiones preparadas y entregadas.

En el presente capítulo se analizaron los procesos de negocio que son objeto de automatización. Se describió el sistema propuesto, exponiendo el modelo del negocio del mismo, los requerimientos funcionales y no funcionales, mostrándose además los diagramas de casos de uso que se han modelado.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo se expone el diseño propuesto para la solución de la aplicación, modelándose los artefactos necesarios que contribuyen a la implementación del sistema. Se muestran los principales diagramas de clases y secuencia del modelo diseño.

3.1. Descripción de la arquitectura, fundamentación

La Arquitectura de Software es la organización fundamental de un sistema encarnada en las relaciones entre ellos y el ambiente, sus componentes y los principios que orientan su diseño y evolución. Como patrón arquitectónico se usa para el desarrollo del sistema el Modelo Vista Controlador (MVC). Este patrón de arquitectura de software separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (vistas, controladores y modelos). En los diagramas, el conjunto de clases clientes, servidoras y formularios representan las vistas, las clases entidades representan los modelos y las clases controladoras a los controladores.

Los componentes del módulo se agrupan en tres capas fundamentales: presentación, negocio y acceso a datos. La capa de presentación está formada por los formularios (contenidos en las páginas xhtml) que mediante controles JSF y RichFaces gestionan los datos con los cuales el usuario realiza determinadas operaciones. La capa de negocio está integrada por las clases controladoras quienes son las responsables de recibir los eventos de entrada desde la vista y definen el estado de los datos y las entidades que manejan.

La capa de acceso a datos está constituida por componentes Hibernate, mediante los cuales se realiza el mapeo de clases Java a tablas de la base de datos y provee facilidades de consultas de datos y de recuperación.

La comunicación entre los elementos de estas capas está regida por el framework Seam ya que integra la capa de presentación (JSF) con la capa de negocios y persistencia. Seam permite además que mediante anotaciones propias de éste las páginas de interfaz de usuario referencien las funcionalidades definidas en las clases controladoras, que estas puedan usar los componentes de acceso a datos y otros de la capa del negocio.

3.2. Modelo de diseño

En el diseño se modela el sistema, para que soporte todo los requisitos –incluyendo los requisitos no funcionales y otras restricciones– y adquiera una arquitectura sólida, robusta y flexible. El propósito concreto del diseño es especificar la forma y comportamiento del sistema desde una vista menos abstracta que el análisis, es decir, se centra en una implementación concreta que debe tomar en cuenta los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia.

3.2.1. Estructuración

A partir de los principales aspectos a tener en cuenta para la elaboración del modelo de diseño, se define una estructura de paquetes que permita dividir el sistema en piezas manejables para su futura implementación. A continuación se muestra el diagrama con los paquetes del diseño, los cuales están estructurados siguiendo el paradigma de la Arquitectura en 3 capas:

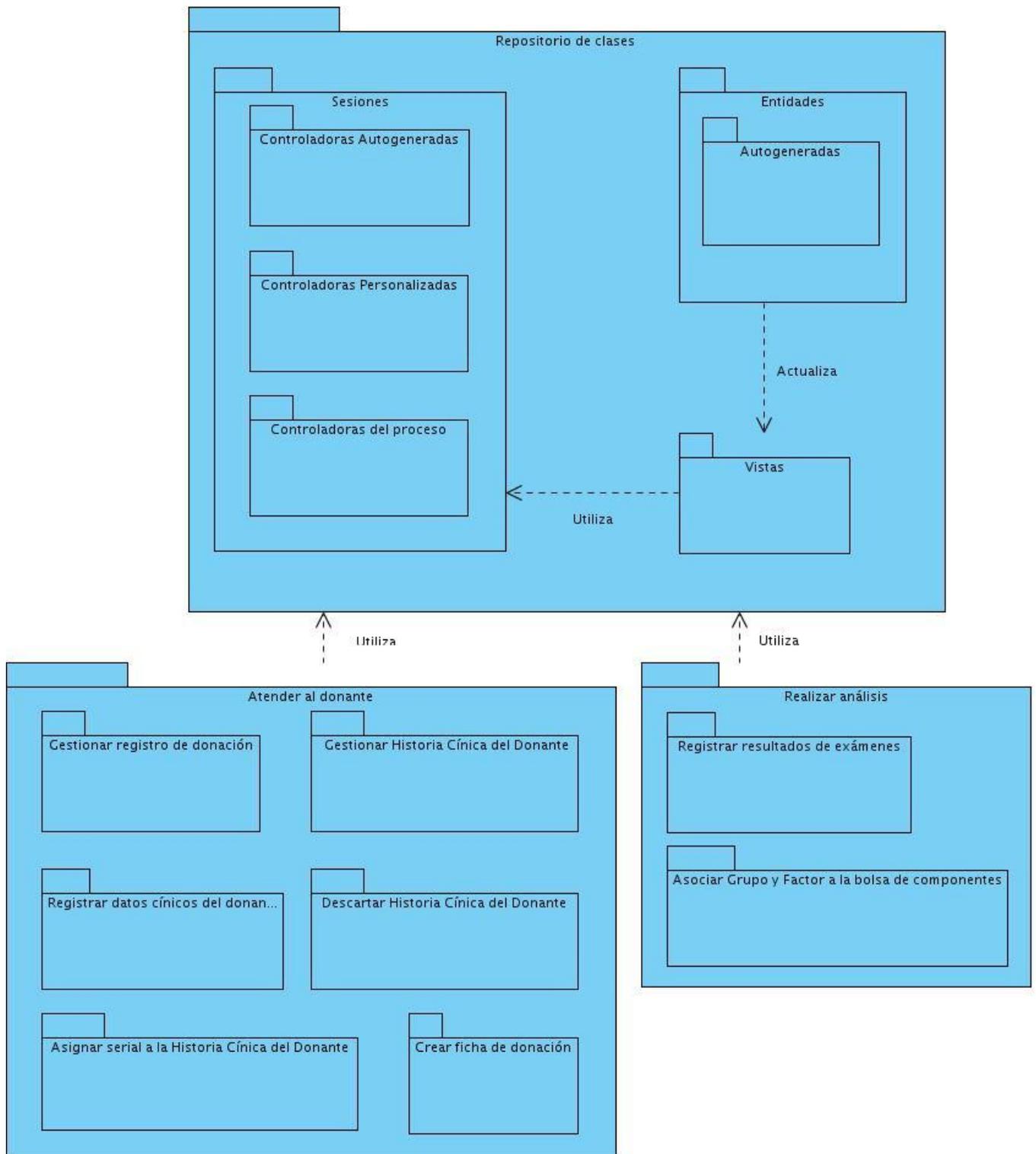


Figura 3.1. Diagrama de paquetes del diseño.

El empaquetamiento se hace por procesos y por clases. A partir de los procesos definidos en el sistema se grafican los paquetes correspondientes: estos utilizan el paquete repositorio de clases para su funcionamiento. Se representan los paquetes correspondientes a los procesos donde se define un paquete para cada uno de ellos. Dentro de cada paquete del proceso se representan los paquetes que contienen las realizaciones de los casos de usos

El paquete Repositorio de clases contendrá todas las clases definidas en el diseño de acuerdo a la tecnología, que serán usadas en la implementación del módulo.

Entidades: Contiene las entidades Autogeneradas y Personalizadas de la base de datos. Las Autogeneradas son aquellas que se autogeneran desde la base de datos utilizando para ello las herramientas ORM de Hibernate. Las Personalizadas son aquellas que son modificadas pudiendo -en algunos casos- presentar relaciones de herencia o composición con las Autogeneradas.

Vistas: Contiene el conjunto de clases que conforman o representan la interfaz de usuario

Sesiones: Contiene las clases Controladoras autogeneradas, las Controladoras personalizadas y las Controladoras del proceso. Las primeras son autogeneradas por el entorno de desarrollo, las segundas son aquellas que son modificadas y las terceras son propias del proceso.

3.3. Definición de elementos de diseño

Para la realización de los Diagramas de Clases del Diseño se utiliza la extensión de UML para la utilización de estereotipos web. Esta extensión presenta como elementos más significativos tres clases UML "Server Page", "Client Page" y "Form" empleadas para el código servidor, código cliente y formularios respectivamente.

El código servidor se encarga de construir o generar el resultado XHTML que conforma el código cliente (<<build>>), los formularios envían sus datos al código servidor para ser procesados los pedidos (<<submit>>), además forman parte del código cliente o resultado XHTML; es por esto que la relación entre la clase empleada para el código cliente y la clase empleada para el formulario es de agregación. Entre las páginas clientes pueden existir vínculos (<<link>>) o redireccionamientos (<<redirect>>). Es importante destacar que una página cliente es construida por una sola página servidora; esta a su vez, puede completar su funcionamiento incluyendo código existente en otra página de este mismo tipo, utilizando la relación de inclusión (<<include>>), que aunque no es propia

de la extensión de UML, las herramientas de modelado la consideran para representar todas las relaciones existentes en el modelo.

Para la nomenclatura de estas clases en los Diagramas de Clases, se siguió la siguiente estructura: PC_<NombreClaseCliente>, FR_<NombreFormulario>, PS_<NombreClaseServidora> y <NombreClaseControladora>, para las páginas clientes, formularios, clases servidoras ubicadas en la Capa de Presentación y clases de la Lógica de Negocio respectivamente. En los diagramas también aparecen las clases entidades y sus relaciones con las clases controladoras. Las mismas pueden ser reconocidas mediante los estereotipos <<Entity Bean>> y <<ORM Persistable>>.

Pueden apreciarse además dos clases muy importantes, RichFaces API y Entity Manager. RichFaces API representa la integración de los componentes de interfaz de usuario, validación y conversión de datos de RichFaces con las páginas clientes. La clase Entity Manager esta estereotipada como <<Interface>> y representa la clase con las que se relacionan las controladoras para llevar a cabo las operaciones de persistencia de datos.

En la estructura de los Diagramas de Clases del Diseño se manifiesta la aplicación de los patrones de diseño, así como las restricciones que establecen los mismos sobre la arquitectura definida. Estos diagramas están estructurados siguiendo el patrón MVC, el cual permite la separación en capas de los objetos que componen el diseño del sistema, e introduce la necesidad de crear tres categorías de objetos principales: vistas, controladores y modelos.

Los modelos encapsulan el estado y exponen la funcionalidad de la aplicación, responden a consultas y notifican a las vistas sobre los cambios en su estado actual. Las vistas muestran y actualizan los modelos y le envían las acciones realizadas por el usuario a los controladores. Estos últimos por su parte, definen el comportamiento de la aplicación convirtiendo las acciones del usuario a actualizaciones de los modelos, seleccionando las vistas para la respuesta, una para cada responsabilidad. En los diagramas, el conjunto de clases clientes, servidoras y formularios representan las vistas, las clases entidades representan los modelos y las clases controladoras a los controladores.

La capa de persistencia o datos en el patrón MVC también forma parte del modelo. En esta capa se utilizan los patrones de diseño Abstract Factory, Singleton y los patrones de arquitectura de aplicaciones empresariales, más específicamente los patrones de persistencia o mapeo objeto relacional, descritos por Martin Fowler, en su libro "Patterns of Enterprise Application Architecture".

Entre estos se encuentran los patrones Active Record, Identity map, Identity field, Foreign key mapping, Association table mapping, Lazy load y Query object.

Otros de los patrones de diseño que se utilizan son los patrones GRASP, los cuales tienen como objetivo la descripción de los principios fundamentales de diseño de objetos para la asignación de responsabilidades y dentro de estos, los patrones Experto, Creador, Alta cohesión y Bajo acoplamiento. Mediante la asignación a cada clase de las tareas o responsabilidades que estas pueden realizar en dependencia de la información que contienen se evidencia el uso de los patrones Experto y Creador, que conservan el encapsulamiento y definen quien será el responsable de crear una instancia de una clase respectivamente.

Al utilizar los patrones Alta cohesión y Bajo acoplamiento se permite la colaboración entre clases o elementos del diseño sin que se afecte su reutilización y entendimiento cuando los mismos se encuentren aislados.

3.4. Diagramas de clases del diseño

Clases del Diseño: Una clase de diseño es una abstracción de una clase o construcción en la implementación del sistema.

Diagrama de Clases del Diseño: Una clase de diseño es una construcción similar en la implementación del sistema. Los diagramas de clases de diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema.

Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa. Los diagramas de clases de diseño del sistema están separados en casos de usos; es decir, cada caso de uso cuenta con un diagrama de clases, para hacer más fácil la comprensión.

Los diagramas de clases también pueden contener paquetes o subsistemas, los cuales se usan para agrupar los elementos de un modelo en partes más grandes.

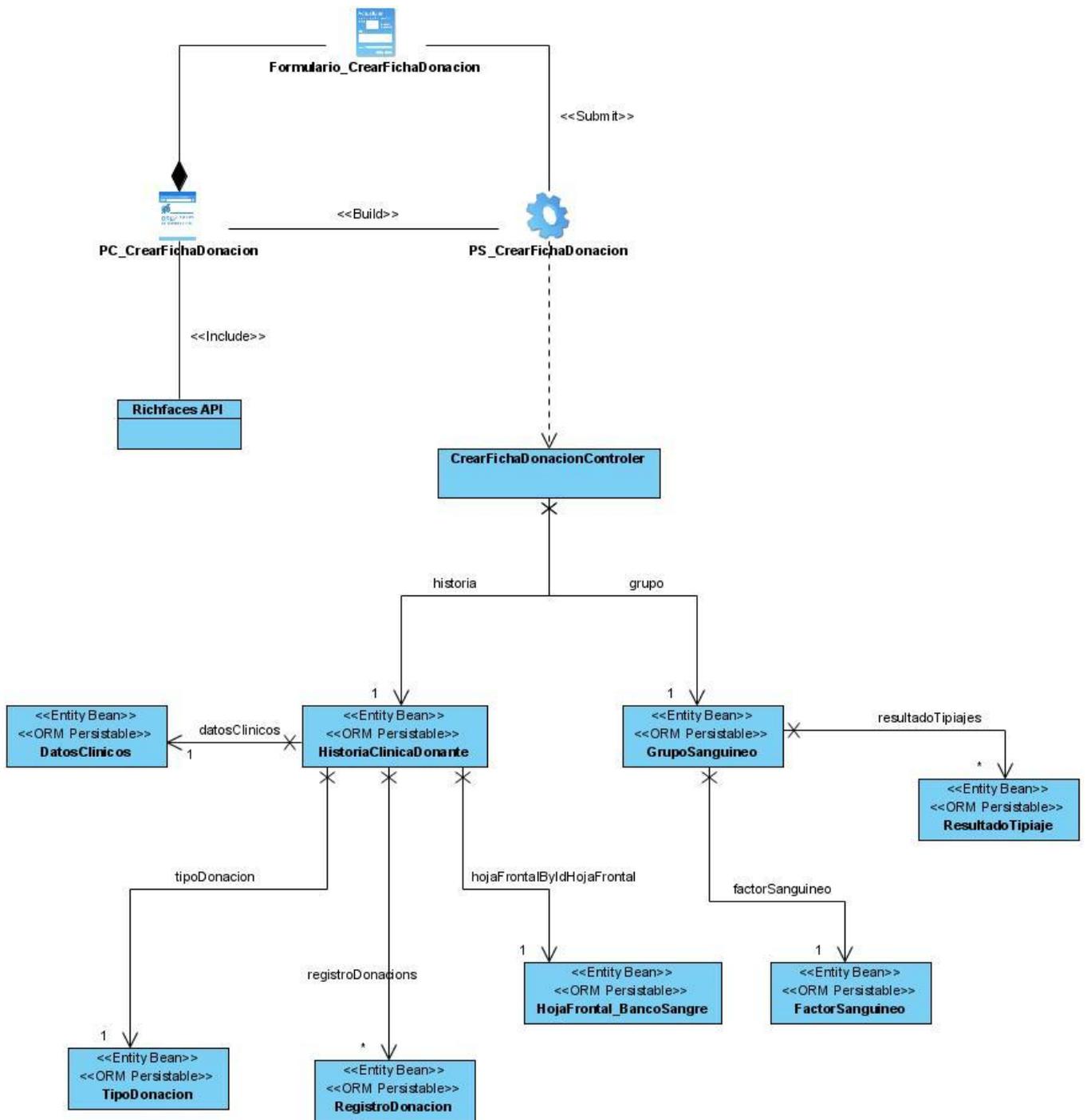


Figura 3.2. Diagrama de Clases del Diseño CUS9_ Crear ficha de donación.

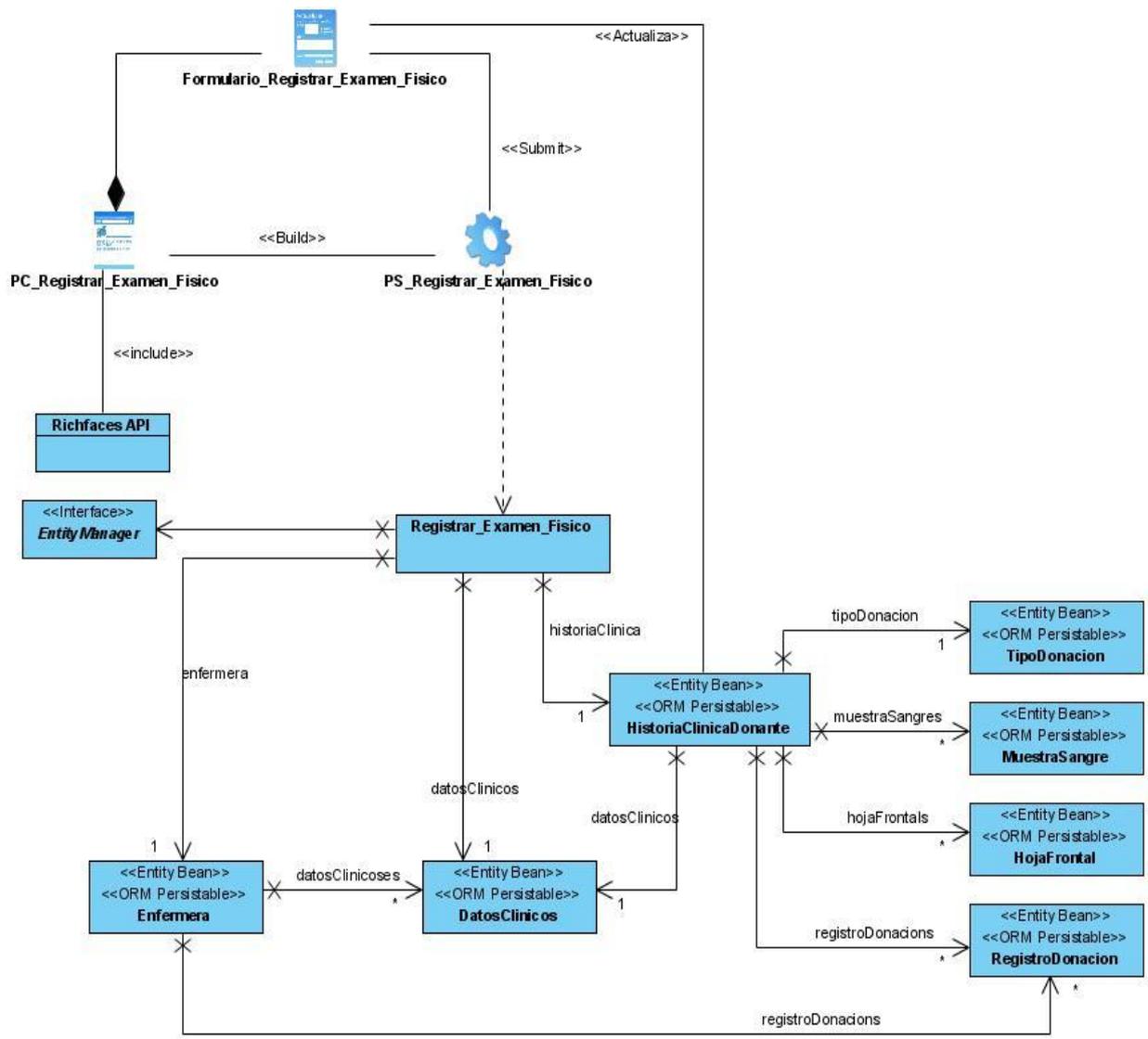


Figura 3.3. Diagrama de Clases del Diseño CUS3_ Registrar datos clínicos del donante.

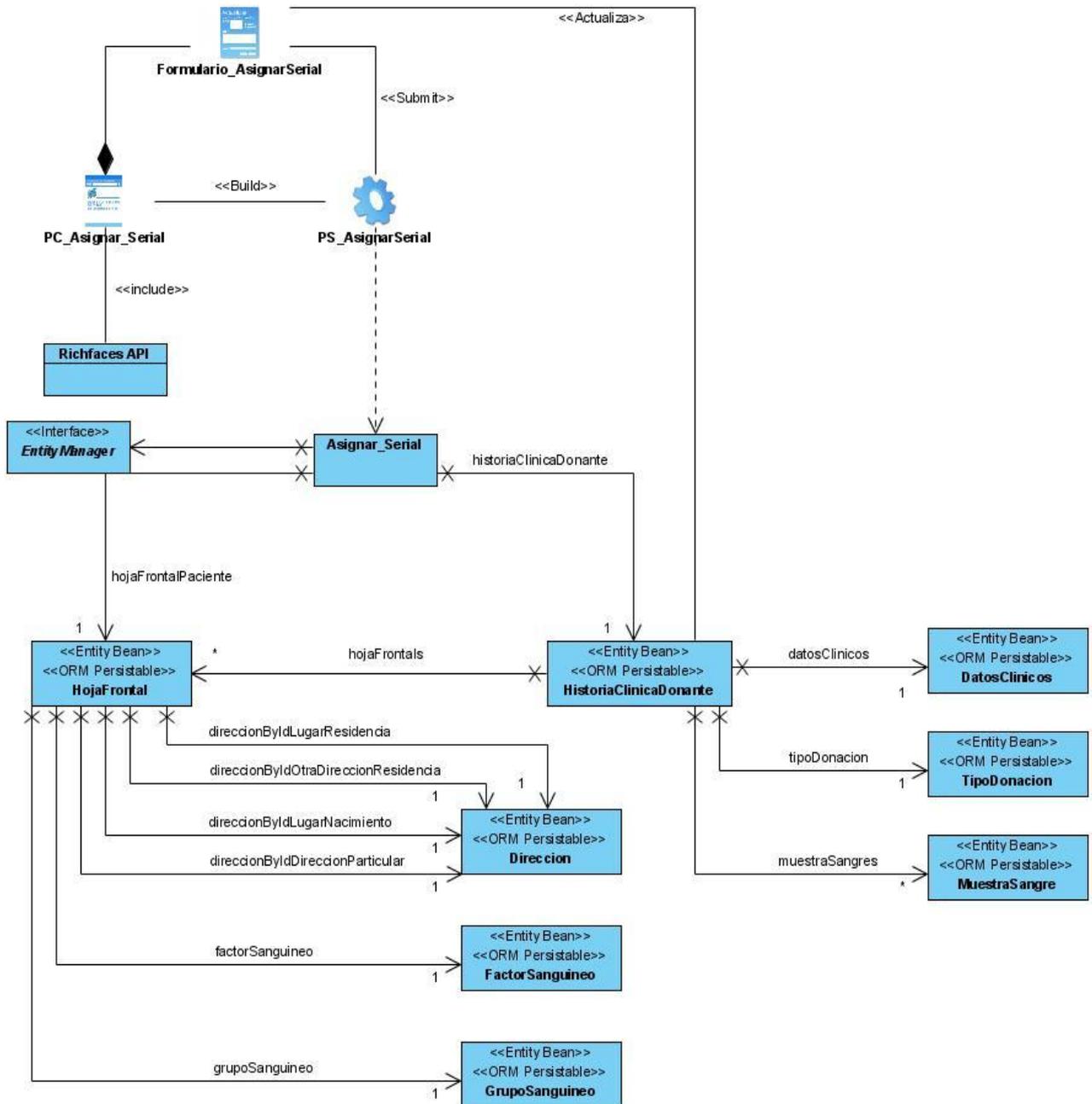


Figura 3.4. Diagrama de Clases del Diseño CUS7_ Asignar serial a la historia clínica del donante.

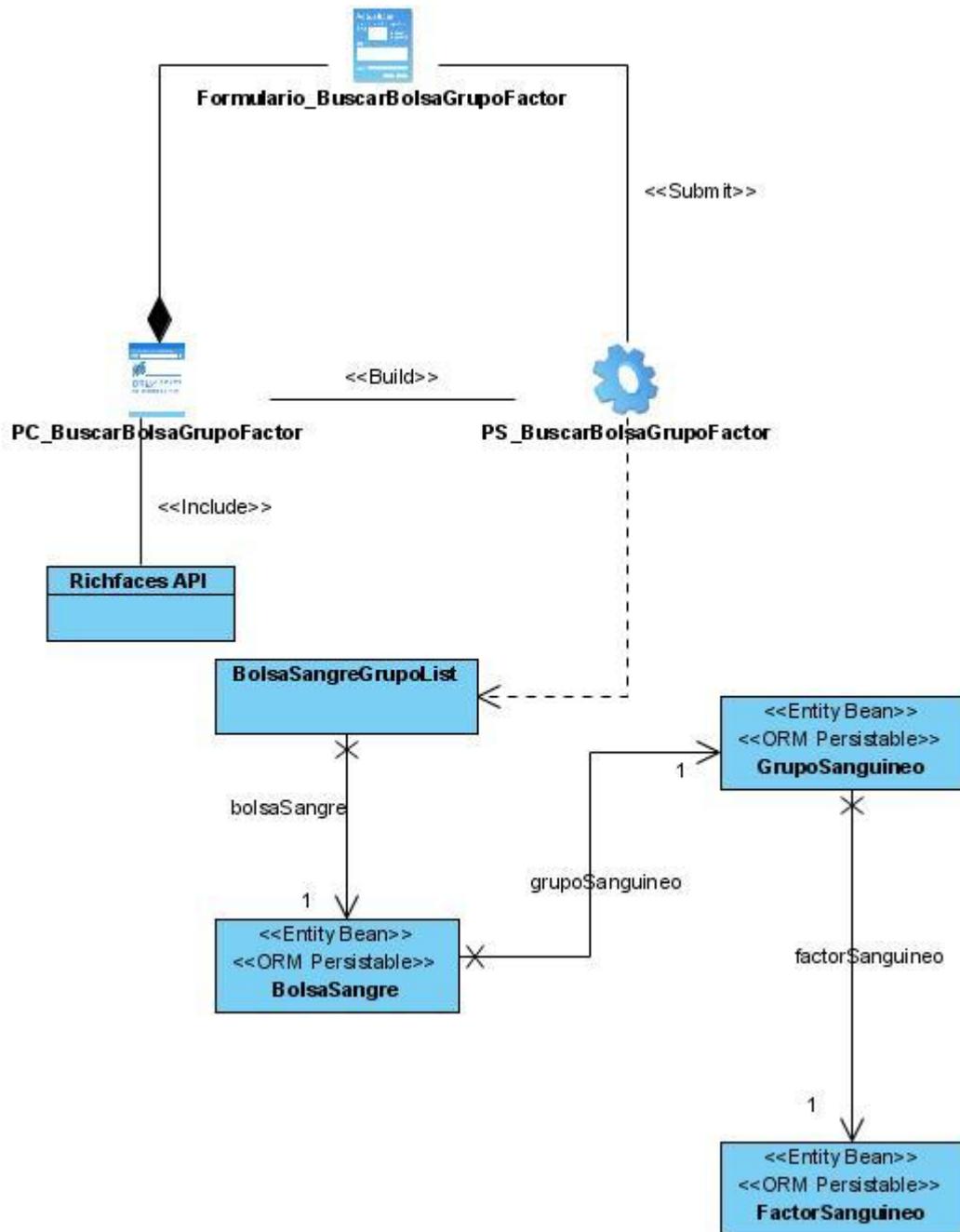


Figura 3.5. Diagrama de Clases del Diseño CUS15_ Buscar grupo y factor por bolsa de componentes sanguíneos.

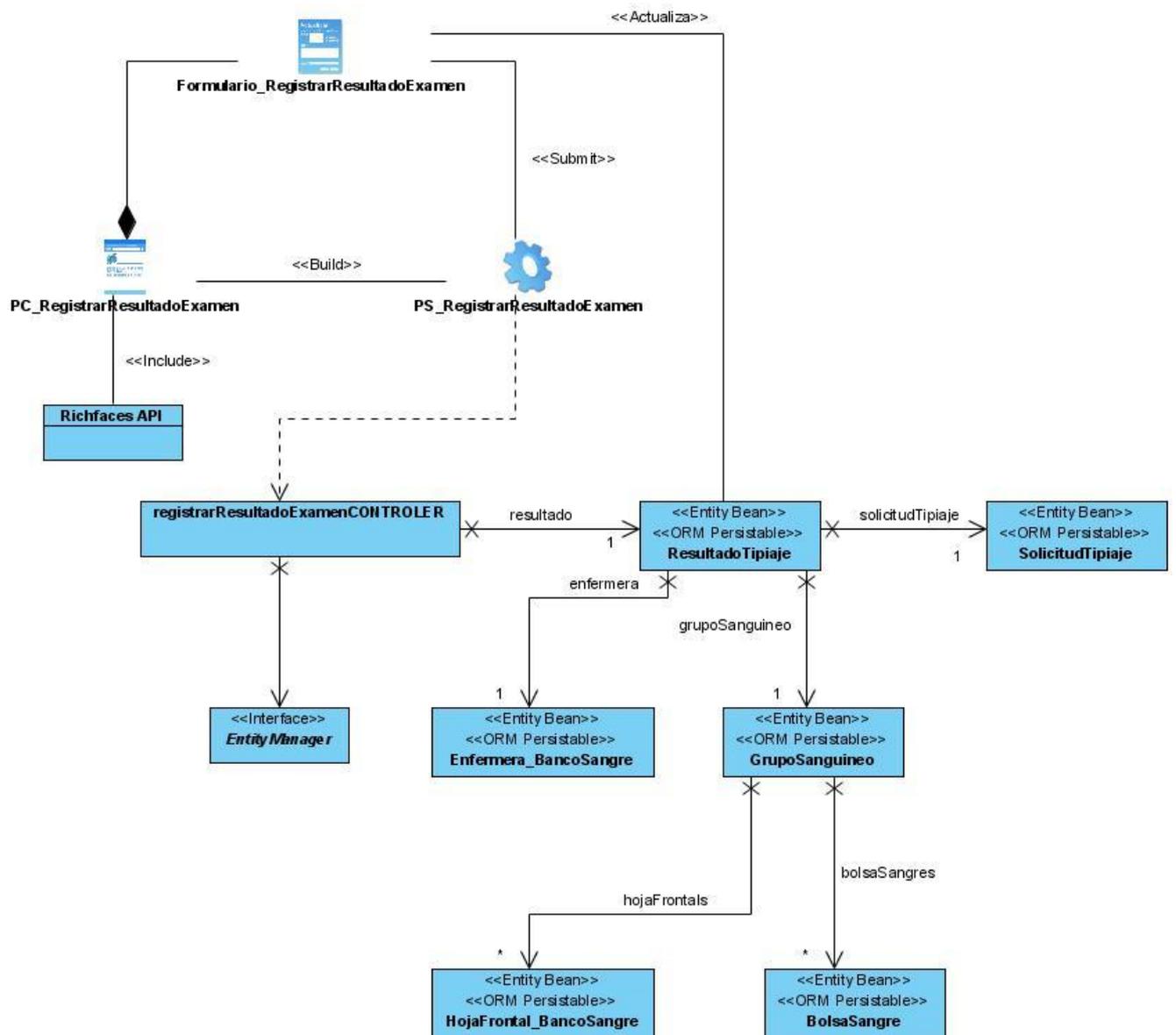


Figura 3.6. Diagrama de Clases del Diseño CUS14_ Registrar resultados de exámenes.

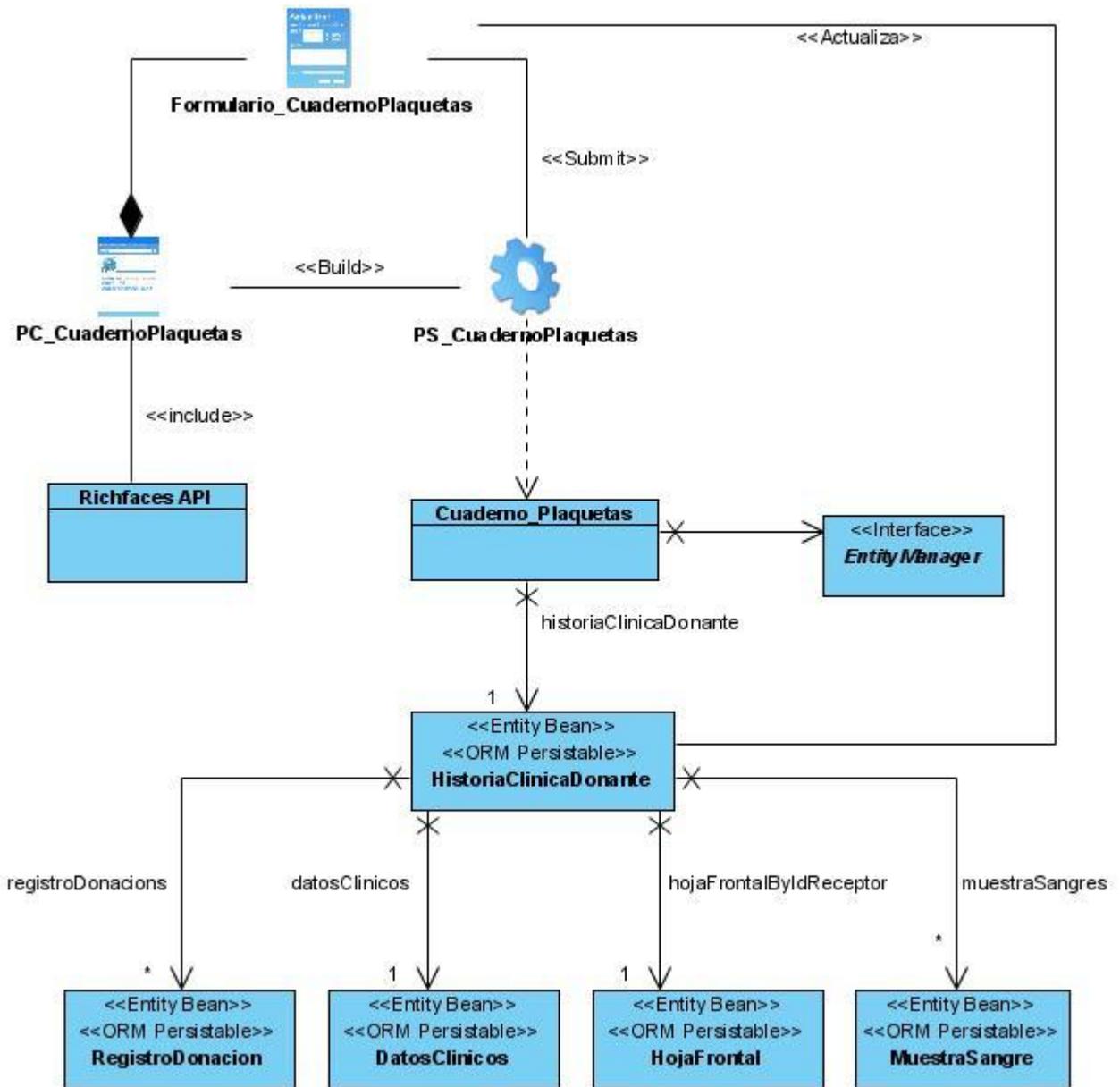


Figura 3.7. Diagrama de Clases del Diseño CUS6_ Actualizar cuaderno de plaquetoaféresis.

3.5. Diagramas de interacción

Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes.

Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control

particular de un caso de uso. Los diagramas de interacción no son sólo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa.

Los diagramas de interacción que se presentan en este capítulo representan los casos de uso arquitectónicamente significativos. Para consultar los restantes diagramas de iteración ir al Anexo 1.

3.6. Descripción de clases del diseño

3.6.1. Descripción de las páginas clientes

Nombre: PC_CrearFichaDonacion
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite a la Secretaria en banco de sangre crear una ficha de donación.

Tabla 3.1. Descripción página cliente PC_CrearFichaDonacion.

Nombre: PC_Registrar_Examen_Fisico
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al hemoterapista registrar los datos clínicos de los donantes.

Tabla 3.2. Descripción página cliente PC_Registrar_Examen_Fisico.

Nombre: PC_AsignarSerial
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite a la Secretaria en banco de sangre asignarle un serial a una historia clínica del donante determinada.

Tabla 3.3. Descripción página cliente PC_AsignarSerial.

Nombre: PC_BuscarBolsaGrupoFactor
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al hemoterapista realizar una búsqueda en las bolsas de componentes por grupo y factor sanguíneo.

Tabla 3.4. Descripción página cliente PC_BuscarBolsaGrupoFactor.

Nombre: PC_RegistrarResultadoExamen
Tipo de Clase: Página Cliente
Descripción General: Es una página web que se ejecuta del lado del cliente, le permite al hemoterapista registrar los resultados de exámenes que fueron enviados a hacer con anterioridad.

Tabla 3.5. Descripción página cliente PC_RegistrarResultadoExamen.

Nombre: PC_CuadernoPlaquetas
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al hemoterapista registrar en el cuaderno de plaquetoaféresis cuando la donación realizada sea de este tipo.

Tabla 3.6. Descripción página cliente PC_CuadernoPlaquetas.

3.6.2. Descripción de las clases controladoras.

Nombre: CrearFichaDonacionControler	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
public void inicio()	Permite realizar una búsqueda en varias entidades y muestra los resultados.

Tabla 3.7. Descripción clase controladora CrearFichaDonacionControler.

Nombre: Registrar_Examen_Fisico	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
public void RegistrarCampos()	Inicializa todos los atributos necesarios para registrar los datos clínicos en la historia clínica del donante
public void save()	Persiste los datos en la historia clínica del donante

Tabla 3.8. Descripción clase controladora Registrar_Examen_Fisico.

Nombre: Asignar_Serial	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
public string BuscarAntecedentes()	Busca si existe algún antecedente anterior a este.
public void asignarSerial()	Asigna el serial a la bolsas de componentes

Tabla 3.9. Descripción clase controladora Asignar_Serial.

Nombre: BolsaSangreGrupoList	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
public void etiquetar(grupo, factor)	Permite realizar una búsqueda sobre las bolsas de sangre atendiendo a los criterios de grupo y factor dados y muestra esta información al usuario

Tabla 3.10. Descripción clase controladora BolsaSangreGrupoList.

Nombre: RegistrarResultadoExamenCONTROLER	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
public void iniciarProceso(id: int)	Inicializa todos los atributos necesarios para registrar los resultados de los exámenes dado el id de la solicitud de examen.
public void save()	Persiste en la base de datos los resultados de los exámenes

Tabla 3.11. Descripción clase controladora RegistrarResultadoExamenCONTROLER.

Nombre: Cuaderno_Plaquetas	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
public void iniciarProceso(id: int)	Inicializa todos los atributos necesarios para actualizar el cuaderno de plaquetoaféresis en el cuaderno.
public void save()	Persiste en la base la plaquetoaféresis.

Tabla 3.12. Descripción clase controladora Cuaderno_Plaquetas.

Como resultado del estudio realizado en este capítulo, correspondiente a la etapa de diseño del sistema, se describió detalladamente la arquitectura, se modelaron los diagramas de clases para cada caso de uso así como sus diagramas de interacción correspondiente, y se describieron las clases del diseño.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

En el presente capítulo se muestra la organización y estructuración de los datos que almacenará la aplicación. Se presentan las relaciones entre los mismos en el modelo de datos y describiendo las entidades a partir de las tablas que estas representan dentro de la base de datos. Se presenta la comunicación entre los diferentes dispositivos para el total funcionamiento del sistema en el correspondiente modelo de despliegue. El conjunto de componentes existente dentro del modelo vista_controlador y su interrelación formando un todo funcional se representa mediante el diagrama de componentes. Además, se tratarán la seguridad del sistema y las políticas para el tratamiento de los errores y los estándares de codificación definidos para la implementación.

4.1. Modelo de datos

El modelo de datos es el enfoque utilizado para la representación de las entidades y sus características dentro de la base de datos. (41)

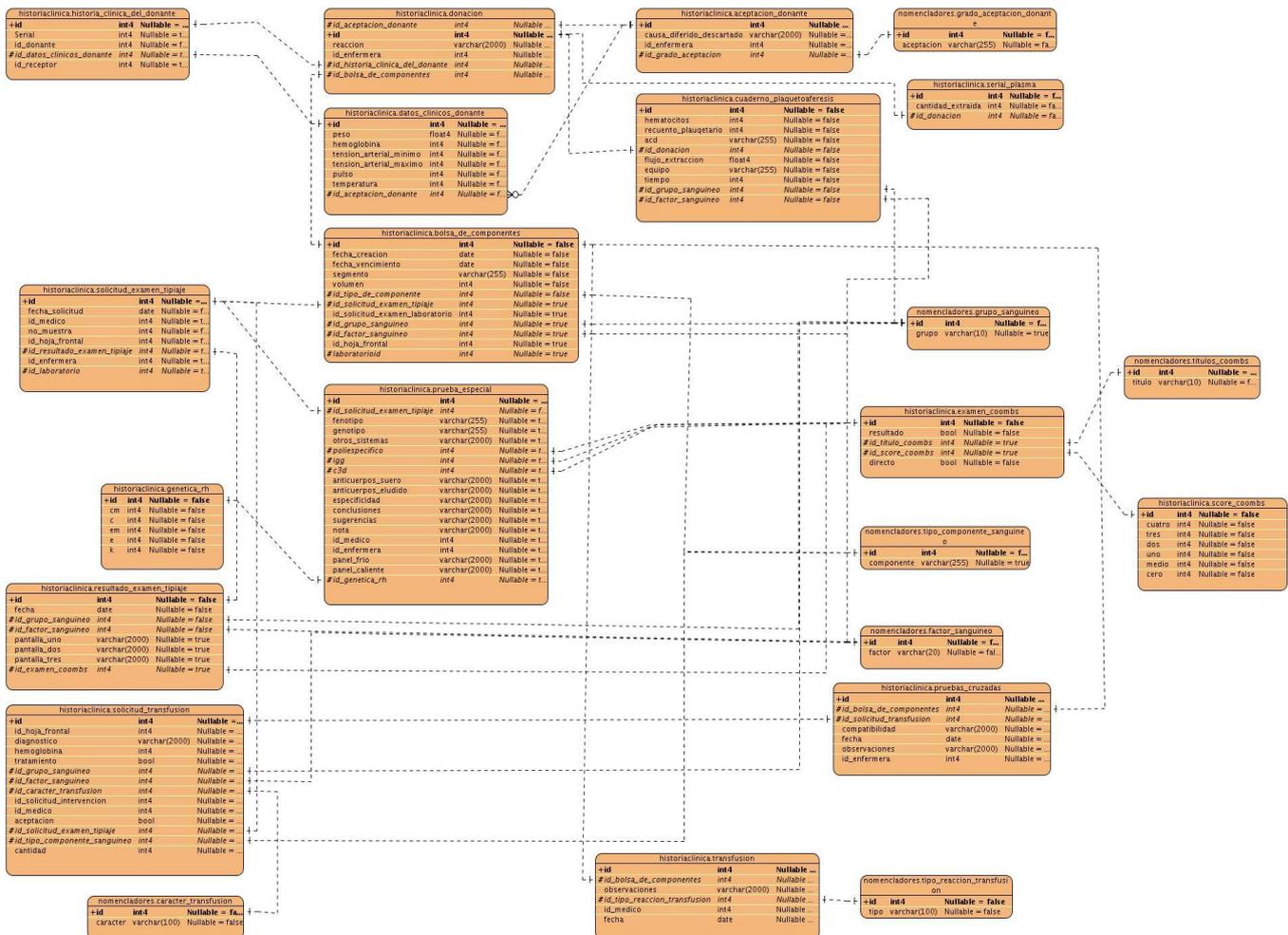


Figura 4.1. Modelo de datos.

4.2. Descripción de las tablas

Nombre: historiaclinica.historia_clinica_del_donante		
Descripción: Almacena datos relacionados con la historia clínica del donante, tales como identificadores de sus datos clínicos y de la hoja frontal relacionada al donante que contiene otra serie de datos personales del mismo. Además contiene los identificadores de aceptación y del tipo de donación que se desea efectuar. También se pueden encontrar las causas de diferido_descartado así como la fecha y el estado de la historia clínica, eliminada o no.		
Atributo	Tipo	Descripción
id	Integer	Identificador de la historia clínica.
motivo_dif_des	Varchar	Motivo del posible rechazo del donante.
id_aceptacion	Integer	Permite acceder a los datos de la aceptación.
id_datos_clinicos	Integer	Permite acceder a los datos clínicos, tales como el peso, hemoglobina, tensión arterial, temperatura.
id_hoja_frontal	Integer	Permite acceder a los datos contenidos en la hoja frontal, en su mayoría información personal del paciente.
procedencia	varchar(100)	Comentario sobre la procedencia del donante.
serial	varchar(20)	Código para identificar la historia clínica.
Id_tipo_donacion	integer	Permite acceder al valor del tipo de donación que se realizara por el donante.
version	integer	Número de la versión de la historia clínica.
eliminado	boolean	Estado de la historia clínica , eliminada o no.
cid	integer	Identificador de la última conversación relacionada a la tabla.
fecha	date	Fecha de la creación de la historia clínica.
id_receptor	integer	Permite el acceso a los datos del encargado de seleccionar al donante.

Tabla 4.1. Descripción de la tabla historia_clinica_donante.

Nombre: historiaclinica.donacion		
Descripción: Contiene datos relacionados con las donaciones. Tiene el identificador de la hoja frontal que almacena los datos del donante. Así mismo contiene la fecha de la donación, el serial y el tipo de la misma. Se pueden encontrar también la serología y la aceptación y un identificador del posible examen de laboratorio a realizarse con relación a la donación. Además contiene el número de versión de la donación y su estado, eliminada o no.		
Atributo	Tipo	Descripción
Id	integer	Identificador de la donación.
id_hoja_frontal	integer	Permite el acceso a los datos almacenados en la hoja frontal, en su mayoría relacionados con el donante.
Fecha	date	Fecha en la que se realiza la donación.
Serial	varchar(100)	Código asignado a la donación.
tipo_donacion	varchar(100)	Enuncia el tipo de donación.
Aceptación	varchar	Comentario sobre la aceptación
Serología	varchar	Serología de la donación.
id_solicitud_exam_laboratorio	integer	Permite acceder a los datos del examen de laboratorio cuando este se realice.
Versión	integer	Número de versión de la donación.
Eliminado	boolean	Estado de la donación, eliminada o no
cid	integer	Identificador de la última conversación relacionada con la tabla.

Tabla 4.2. Descripción de la tabla hc_donaciones.

Nombre: nomencladores.grado_aceptacion_donante		
Descripción: Contiene datos relacionados con el grado de aceptación del donante.		
Atributo	Tipo	Descripción
id	integer	Identificador del grado de aceptación del donante.
valor	varchar	Valor del grado de aceptación.
version	integer	Número de versión del grado de aceptación del donante.
eliminado	boolean	Estado del grado de aceptación, eliminado o no.

Tabla 4.3. Descripción de la tabla grado_aceptacion_donante.

Nombre: historiaclinica.datos_clinicos_donante		
Descripción: Contiene datos relacionados a características clínicas como el peso, la hemoglobina, el pulso, la temperatura corporal y la presión arterial.		
Atributo	Tipo	Descripción
id	Integer	Identificador de los datos clínicos.
id_enfermera	Integer	Permite acceder a los datos de la enfermera relacionada con la recogida de los datos clínicos.
peso	Integer	Peso corporal del donante.
hemoglobina	Integer	Valor de la hemoglobina.
presion_dias	Integer	Valor de la presión diastólica (más baja).
presion_sist	Integer	Valor de la presión sistólica (más alta).
pulso	Integer	Valor del pulso.
id_caract_pulso	Integer	Permite acceder al carácter del pulso.
temperatura	double precision	Temperatura corporal.
id_aceptacion	Integer	Permite acceder a la aceptación.
motivo_dif_des	Varchar	Comentario sobre el motivo del rechazo del donante.
version	Integer	Número de versión de los datos clínicos.
eliminado	Boolean	Estado de los datos clínicos, eliminados o no.
cid	Integer	Identificador de la última conversación relacionada con la tablas.

Tabla 4.4. Descripción de la tabla datos_clinicos.

Nombre: historiaclinica.bolsa_de_componentes		
Descripción: Contiene datos relacionados con las características de una bolsa de sangre entre las cuales están los identificadores del grupo y componente sanguíneos. Las fechas de creación y vencimiento de la bolsa. Además contempla si esta etiquetada o no y si contiene el resultado de la serología o no.		
Atributo	Tipo	Descripción
id	Integer	Identificador de la bolsa de sangre.
serial	varchar(15)	Código identificativo de la donación relacionada.
segmento	varchar(30)	Código identificativo de la bolsa de sangre (coincide con su código de barra).
creada	Date	Fecha de creación de la bolsa.
vence	Date	Fecha de vencimiento de la bolsa.
id_grupo	Integer	Permite acceder al grupo sanguíneo.
id_componente	Integer	Permite acceder al tipo de componente de la bolsa.
etiqueteada	Boolean	Dice si esta etiquetada o no la bolsa.
serologias	Boolean	Dice si posee o no serologías.
version	Integer	Número de versión de la bolsa.
eliminado	Boolean	Estado de la bolsa, eliminada o no.
cid	integer	Identificador de la última conversación relacionada con la tabla.

Tabla 4.5. Descripción de la tabla bolsa_sangre.

Nombre: historiaclinica.cuaderno_plaquetoaféresis		
Descripción: Recoge los datos referentes al cuaderno de plaquetoaféresis. Contiene un identificador de la donación para acceder a los datos que contiene la misma. Recoge el resultado del hematocrito. Además almacena el valor del recuento plaquetario, el anticoagulante utilizado, el tiempo empleado y el equipo utilizado en el procedimiento.		
Atributo	Tipo	Descripción
id	integer	Identificador del cuaderno.
id_donacion	integer	Permite el acceso a la donación y al conjunto de datos contenidos en la misma.
hematocrito	varchar(100)	Comentario sobre el resultado del hematocrito.
recuento_plaquetario	integer	Resultado del recuento plaquetario.
acd	varchar(20)	Solución anticoagulante utilizada.
flujo_extracion	double precision	Valor del flujo de extracción.
tiempo	integer	Tiempo empleado en el proceso.
equipo	varchar(20)	Nombre del equipo utilizado en el proceso.
version	integer	Número de versión del cuaderno.
eliminado	boolean	Estado del cuaderno, eliminado o no.
cid	integer	Identificador de la última conversación relacionada con la tabla.

Tabla 4.6. Descripción de la tabla cuaderno_de_plaquetas.

Nombre: historiaclinica.solicitud_examen_tipiaje		
Descripción: Contiene datos necesarios para efectuar un examen tales como el número de la muestra relacionada con el donante a examinar, el identificador de la historia clínica del mismo, el número de versión del examen y el identificador de la solicitud de transfusión asociada al examen y el estado del mismo, eliminado o no.		
Atributo	Tipo	Descripción
id	integer	Identificador del examen.
Número_muestra	integer	Número de la muestra asociada al examen.
id_hcd	integer	Permite el acceso a la historia clínica del donante, o sea a todos sus datos.
version	integer	Número de la versión del examen.
eliminado	boolean	Estado del examen, eliminado o no.
cid	integer	Identificador de la última conversación relacionada con la tabla.
id_solicitud_transfucion	integer	Permite el acceso a la solicitud asociada al examen.

Tabla 4.7. Descripción de la tabla solicitud_tipiaje.

Nombre: historiaclinica.resultado_examen_tipiaje		
Descripción: Almacena los datos relacionados con el resultado de tipiaje, entre otros los identificadores del grupo sanguíneo, de la solicitud y de la enfermera encargada del examen. Las pantallas (1,2,3) y los coombs directo e indirecto.		
Atributo	Tipo	Descripción
id	Integer	Identificador del resultado.
id_grupo	integer	Permite acceder al grupo sanguíneo.
pantalla1	varchar	Resultado de examen de anticuerpos presentes en la sangre
pantalla2	varchar	Resultado de examen de anticuerpos presentes en la sangre
pantalla3	varchar	Resultado de examen de anticuerpos presentes en la sangre
coombs_directo	varchar	Resultado del examen coombs directo.
Id_enfermera	integer	Permite el acceso a los datos de la enfermera.
Id_solicitud	integer	Permite el acceso a los datos de la solicitud.
coombs_indirecto	varchar	Resultado del examen coombs indirecto.
observaciones	varchar	Observaciones sobre los resultados del examen
version	integer	Número de versión del resultado de examen.
eliminado	boolean	Estado del examen, eliminado o no.
cid	integer	Identificador de la última conversación relacionada a la tabla.

Tabla 4.8. Descripción de la tabla resultado_tipiaje.

Nombre: historiaclinica.serial_plasma		
Descripción: Contiene datos relacionados con la donación de sangre mediante su identificador. Además almacena la cantidad de plasma extraído y el estado del serial de plasma, eliminado o no.		
Atributo	Tipo	Descripción
id	integer	Identificador
id_donacion	integer	Permite el acceso a los datos de la donación.
cantidad_plasma	integer	Cantidad de plasma extraído en la donación.
version	integer	Versión del serial de plasma.
eliminado	boolean	Estado del serial de plasma, eliminado o no.
cid	integer	Identificador de la última conversación relacionada con la tabla.

Tabla 4.9. Descripción de la tabla serial_plasma.

Nombre: historiaclinica.transfusion		
Descripción: Contiene datos relacionados con las transfusiones, entre ellos la fecha de la transfusión, el componente sanguíneo así como el solicitante de la transfusión.		
Atributo	Tipo	Descripción
id	intger	Identificador de la transfusión.
fecha	date	Fecha de la transfusión.
componente	varchar(200)	Componente sanguíneo asociado a la transfusión.
solicitante	varchar	Solicitante de la transfusión.
version	integer	Número de versión de la transfusión.
eliminado	boolean	Estado de la transfusión, eliminada o no.
cid	integer	Identificador de la última conversación relacionada a la tabla.

Tabla 4.10. Descripción de la tabla hc_transfusiones.

Nombre: historiaclinica.solicitud_transfusion		
Descripción: Almacena los identificadores del paciente necesitado de la transfusión y del médico que la solicita así como el carácter de la solicitud, la fecha y el motivo de la misma. Además el identificador del componente sanguíneo, la hemoglobina del paciente y el estado de la solicitud, aceptada o eliminada.		
Atributo	Tipo	Descripción
id	integer	Identifica la solicitud.
Id_caracter	integer	Identifica el carácter de la transfusión, mediante este se puede acceder al valor del mismo.
Id_hoja_frontal	integer	Es el identificador de la hoja frontal, en la cual se almacenan los datos del paciente.
Id_medico	integer	Es el identificador del médico, con este puede acceder a los datos del médico solicitante.
Id_componente	integer	Identificador del componente sanguíneo, permite acceder al valor del componente sanguíneo que necesita el paciente.
cantidad	integer	Representa la cantidad de bolsas necesarias.
aceptada	boolean	Dice si la solicitud fue aceptada o no
motivo_rechazo	varchar	Comentario sobre la causa del posible rechazo de la solicitud.
version	integer	Enumera la versión de la solicitud.
eliminado	boolean	
cid	integer	Identifica la última conversación relacionada con la tabla
Fecha	date	Fecha de la solicitud.
hb	double precision	Valor de la hemoglobina.
motivo_solicitud	varchar	Descripción del porque de la solicitud.

Tabla 4.11. Descripción de la tabla solicitud_transfusion.

Nombre: nomencladores.factor_sanguineo		
Descripción: Almacena el valor del factor sanguíneo.		
Atributo	Tipo	Descripción
id	integer	Identificador del tipo de factor sanguíneo.
valor	varchar	Valor del factor sanguíneo.
version	integer	Número de versión del factor sanguíneo.
eliminado	boolean	Estado del factor, eliminado o no
cid	integer	Identificador de la última conversación relacionada con la tabla.

Tabla 4.12. Descripción de la tabla prueba_compatibilidad.

Nombre: historiaclinica.prueba_especial		
Descripción: Contiene datos presentes en las pruebas especiales, entre los cuales se encuentran anticuerpos presentes en sueros, anticuerpos presentes en eludidos, genotipo probable, fenotipo, sistema rh, pantallas (1,2,3) y los diferentes identificadores de exámenes coombs.		
Atributo	Tipo	Descripción
id	integer	Identificador de la prueba.
id_enfermera	integer	Permite acceder a los datos de la enfermera relacionada con la prueba.
id_solicitud	integer	Permite acceder a los datos de la solicitud de transfusión asociada a la prueba.
version	integer	Número de versión de la prueba.
eliminado	boolean	Estado de la prueba, eliminada o no.
cid	integer	Identificador de la última conversación asociada a la tabla.
id_coombs_poliespecifico	integer	Permite el acceso a los datos almacenados en coombs específicamente a los poliespecíficos.
id_directo_c3d	integer	Permite el acceso a los datos almacenados en coombs específicamente a los directo c3d.
Id_directo_igg	integer	Permite el acceso a los datos almacenados en coombs específicamente a los directo igg.
anticuerpos_presentes_sueros	varchar	Nombrar anticuerpos.
anticuerpos_presentes_eludidos	varchar	Nombrar anticuerpos.
especificidad	varchar	Comentarios específicos.
sugerencia	varchar	Sugerencias acerca de la prueba.
nota	varchar	Anotaciones
fenotipo	varchar	Fenotipo sanguíneo del paciente.
genotipo_probable	varchar	Posible genotipo sanguíneo.
otros_sistemas	varchar	Otros sistemas presente en la sangre.
sistemas_rh	integer	Factor sanguíneo.
pantalla1	varchar	Resultado de examen de anticuerpos presentes en la sangre
pantalla2	varchar	Resultado de examen de anticuerpos presentes en la sangre
pantalla3	varchar	Resultado de examen de anticuerpos presentes en la sangre
conclusiones	varchar	Conclusiones

Tabla 4.13. Descripción de la tabla pruebas_especiales.

Nombre: historiaclinica.examen_coombs		
Descripción: Almacena datos relacionados con el examen coombs, contiene un identificador del título del examen y varios valores score (4,3,2,1,0).		
Atributo	Tipo	Descripción
id	integer	Identificador del examen coombs.
coombs	varchar	Nombre del examen coombs.
titulo	integer	Permite acceder al título del examen.
score4	varchar(100)	Cantidad de anticuerpos del score en cuestión.
score3	varchar(100)	Cantidad de anticuerpos del score en cuestión.
score2	varchar(100)	Cantidad de anticuerpos del score en cuestión.
score1	varchar(100)	Cantidad de anticuerpos del score en cuestión.
score1/2	varchar(100)	Cantidad de anticuerpos del score en cuestión.
score0	varchar(100)	Cantidad de anticuerpos del score en cuestión.

Tabla 4.14. Descripción de la tabla coombs.

Nombre: nomencladores.tipo_reaccion_transfucion		
Descripción: Almacena el valor de la reacción de la transfusión.		
Atributo	Tipo	Descripción
id	integer	Identificador del tipo de reacción de transfusión.
valor	varchar	Valor de reacción de transfusión.
version	integer	Número de versión de la reacción de transfusión.
eliminado	boolean	Estado de la reacción, eliminada o no
cid	integer	Identificador de la última conversación relacionada con la tabla.

Tabla 4.15. Descripción de la tabla tipo_reaccion_transfucion.

4.3. Diagrama de despliegue

Los Diagramas de Despliegue 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. (42)

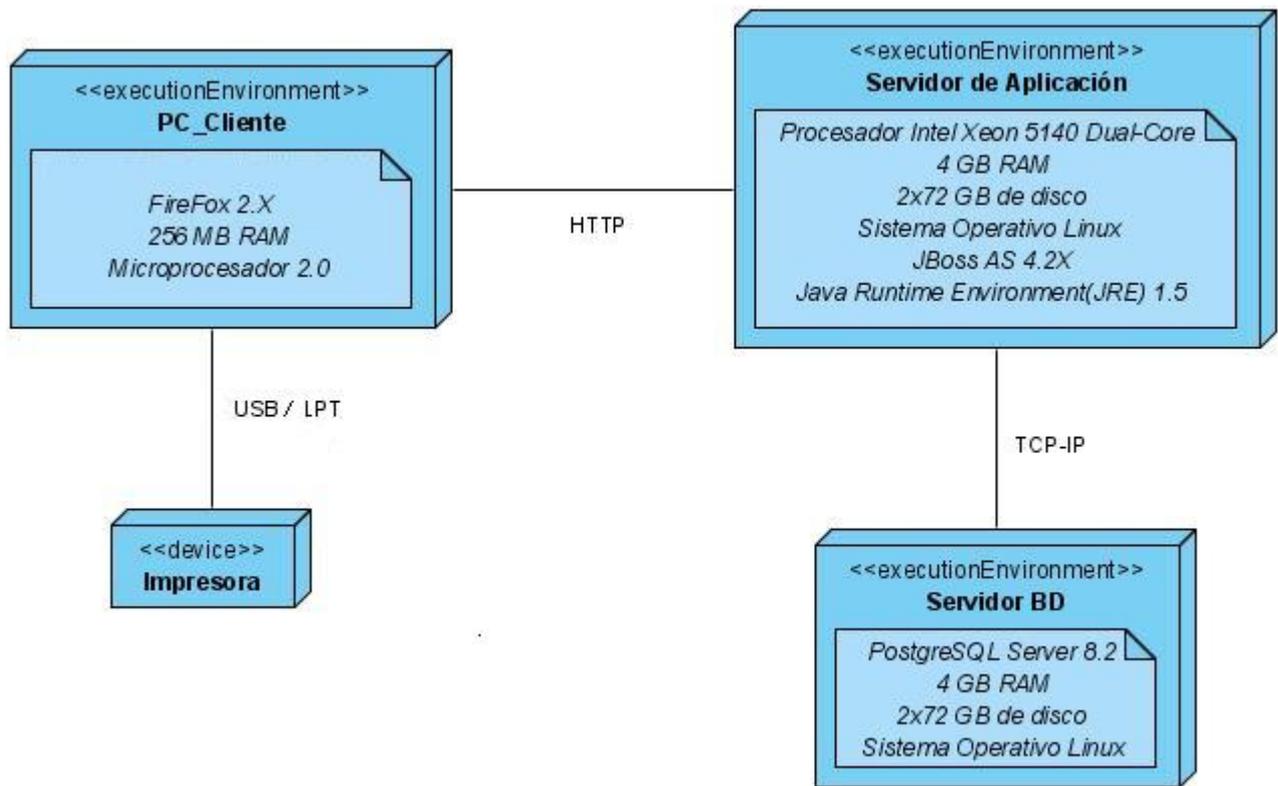


Figura 4.2. Diagrama de despliegue.

4.4. Vista de componentes

El modelo de componentes ilustra los componentes de software que se usarán para construir el sistema. Los componentes son agregaciones de alto nivel de las piezas de software más pequeñas y proveen un enfoque de construcción de bloques para la elaboración del software.

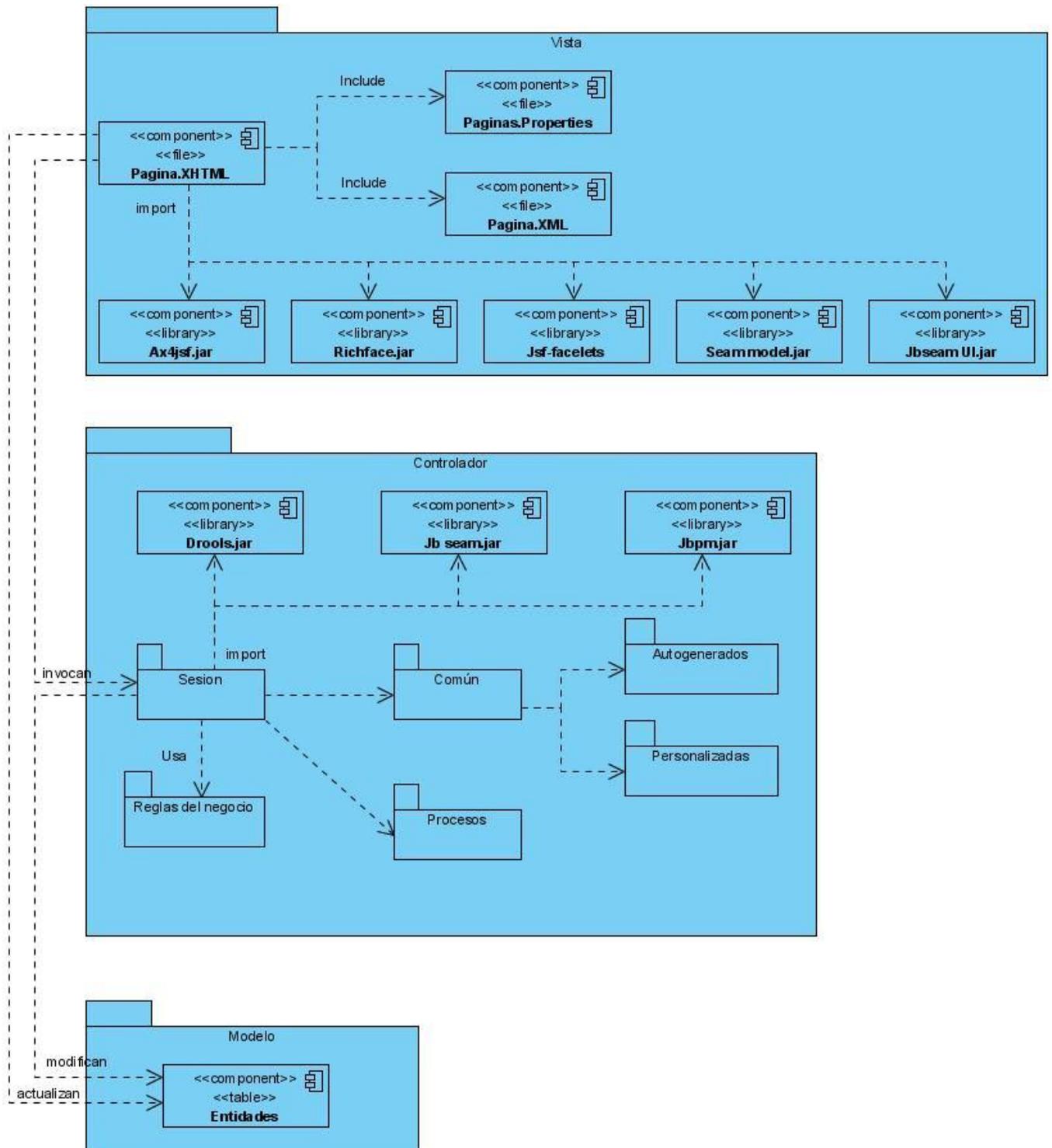


Figura 4.3. Diagrama de componentes.

4.5. Tratamiento de errores

Las excepciones son aquellas situaciones excepcionales que requieren un tratamiento especial en el sistema. Estas pueden ocurrir dentro del programa durante su ejecución, y de ser así, podrían interrumpir su correcto funcionamiento. Para lograr obtener un sistema estable, confiable y libre de errores, existen una serie de técnicas que permiten formas normalizadas de manejar estos errores, permitiendo anticiparse a los problemas potenciales previstos e imprevistos, así como permitir al programador reconocerlos, fijar su ubicación y corregirlos. A este proceso se le llama tratamiento de excepciones.

En la solución propuesta el proceso de tratamiento de excepciones se realiza mediante la estructura sintáctica-semántica try catch para la captura y tratamiento de excepciones: la sentencia try para generar la excepción en caso de que alguna de las sentencias que se encuentran dentro del bloque lo requiera y catch para manejar las mismas.

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

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

4.6. Seguridad

Para que un sistema se pueda definir como seguro debe tener las siguientes características:

- **Integridad:** La información sólo puede ser modificada por quien está autorizado y de manera controlada.
- **Confidencialidad:** La información sólo debe ser legible para los autorizados.
- **Disponibilidad:** Debe estar disponible cuando se necesita.

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

La seguridad del sistema se manifiesta de la siguiente manera.

Acceso al sistema: El acceso al sistema estará restringido para cada usuario en dependencia de los permisos establecidos teniendo en cuenta el rol que desempeña en la institución hospitalaria. Para validar esto, cada usuario del sistema tiene un nombre de usuario y contraseña que el sistema solicita siempre que alguien quiere acceder a la aplicación y luego verifica que los datos introducidos sean válidos, si es así, el sistema muestra al usuario las funcionalidades a las que tiene acceso.

Registro de trazas: El sistema lleva un registro de trazas para poder tener control de qué hace cada usuario. Una traza se registra por cada acción que realiza un usuario sobre el sistema.

Administrar seguridad: El administrador del sistema es quien asigna y deniega permiso a roles y usuarios en las funcionalidades de los módulos.

Configurar funcionalidades: El administrador del sistema es el encargado de configurar funcionalidades, algunas de estas funcionalidades son administrar roles, usuarios y seguridad.

4.7. Estrategias de codificación. Estándares y estilos a utilizar

Un estándar de codificación es “Un conjunto de reglas de notación y nomenclatura, específicas de cada lenguaje de programación, que se usan y se siguen durante la fase de implementación (codificación) de una aplicación y reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores que no son detectados por los compiladores, reduciendo el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismos.”

Las convenciones de nombres hacen los programas entendibles y fáciles de leer. Para el desarrollo del sistema propuesto se usan las convenciones de codificación en Java de Sun (Sun Code Conventions). Algunas de las convenciones más importantes de este estándar son las siguientes:

Indentación		
Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.	
Aspectos Generales	El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla. Los inicios ({) y cierre (}) de ámbito deber estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.	
Comentarios, separadores, líneas, espacios en blanco y márgenes.		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.
Líneas en blanco	Se emplean antes y después de métodos, clases y estructuras.	Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto
Aspectos generales	Sobre el comentario	Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción
	Sobre los espacios en blanco	No se debe usar espacio en blanco: Después del corchete abierto y antes del cerrado de un ar Después del paréntesis abierto y antes del cerrado. Antes de un punto y coma.
Variables y constantes		
Apariencia de variables	Las variables tendrán un prefijo para el tipo de datos en minúscula.	El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, la cual identificara el tipo de datos al que se refiere (ver tabla 1.1), en caso de que sea un nombre compuesto se empleará notación CamellCasing**. Ejemplo: sNombrePaciente
Apariencia de constantes	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
Aspectos generales	Nombres de las variables y constantes	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.
Clases y Objeto		

Apariencia de clases y objetos	Primera letra en mayúscula	Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: MiClase(). Para el caso de las instancias se comenzará con un prefijo que identificara el tipo de dato, este se escribirá en minúscula.
Apariencia de atributos	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación CamellCasing**.
Apariencia de las funciones	Primera letra en mayúscula	Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Se empleará notación PascalCasing*. Ejemplo: function BuscarUnidad(). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set
Declaración de parámetro en funciones	Agrupados por tipos Poner los string 1 numéricos 2, además, agrupar según valores por defecto.	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen, especificando el tipo de datos (<i>ver tabla 1.1</i>).
Aspectos generales	Sobre las clases, los objetos, los atributos y las funciones.	El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.
Controles		
Apariencia de los controles.	Los controles tendrán un prefijo para el tipo de datos en minúscula.	El nombre que se le da a los controles deben comenzar con las primeras letras en minúscula, las cuales identificarán el tipo de datos al que se refiere (<i>ver tabla 1.2</i>), en caso de que sea un nombre compuesto se empleará notación CamellCasing**. Ejemplo: btnAceptar

Tabla 4.16. Convenciones del estándar de codificación de SUN Microsystem.

(*) Notación PascalCasing: Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula. Ejemplo: NotacionPascalCasing.

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

Tipo Datos	Prefijo	Ejemplo
Int	i	iCantPacientes
flota	f	fPesoPaciente
double	d	dPesoCarro
bool	b	bPacienteActivo
string	s	sNombrePaciente
char	c	cLetra
De tipo enum	e	eSexo
byte	b	bCantDiasPaciente
sbyte	sb	sbEdadPaciente
short	sh	shVariableShort
ushort	us	usVariableUshort
uint	ui	uiVariableUint
long	l	lVariableLong
ulong	ul	ulVariableUlong
decimal	dc	dcVariableDecimal
Objetos	o	oPacienteHistorico
Objetos de tipo Struct	st	stUnaStruct

Tabla 4.17. Especificaciones de los tipos de datos.

Control	Prefijo	Ejemplo
Botón	btn	btnAceptar
Etiqueta	lbl	lblNombre
Lista/Menú	mn	mnPrincipal
Campo de Texto	txt	txtFecha
Botón de Opción	bpt	optSexo
Casilla de Verificación	chx	chxBorrar
Casilla de Selección	cbx	cbxSexo

Tabla 4.18. Especificaciones de los controles.

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

Con la culminación del capítulo quedaron totalmente descritas las tablas modeladas como entidades dentro del modelo de datos. Se agruparon los componentes según su contenido y nivel en el cual es utilizado dentro del modelo, la vista o el controlador. Se relacionaron los nodos físicos que componen del diagrama de despliegue. Además quedó expuesta la política para el tratamiento de errores y se

enumeraron y mostraron en forma de tablas los estándares de codificación para la implementación. Además se describieron las normas de seguridad del sistema.

CONCLUSIONES

Con la elaboración del presente trabajo se arriba a las siguientes conclusiones:

- Con el análisis de los procesos de negocio se modeló un sistema para dar solución a las principales necesidades existentes en el área de banco de sangre.
- Los Sistemas de Información Hospitalaria que fueron analizados no responden a todas las necesidades requeridas.
- La arquitectura definida por el Área Temática Gestión Hospitalaria contiene los requisitos necesarios para el desarrollo de una aplicación con las características esperadas.
- La obtención de los flujos de trabajo de “Modelado de Negocio”, “Gestión de Requerimientos” y “Diseño” es relevante para una correcta implementación del sistema.
- Con la implementación de los procesos que se llevan a cabo en los bancos de sangre se espera contribuir a la automatización de forma eficiente de la gestión de información en esta área.

De esta forma se da cumplimiento al objetivo y tareas planteadas; se logró desarrollar el módulo Banco de Sangre, del Sistema de Información Hospitalaria alas HIS, que facilita la gestión de información en esta área de las instituciones hospitalarias.

RECOMENDACIONES

Por la experiencia adquirida en el desarrollo de a aplicación se recomienda:

- Incluir funcionalidades para la gestión de bolsas de componentes e intercambios con otras instituciones.
- Brindar capacitación al personal que va a interactuar con el sistema.
- Incluir sistema de Ayuda en línea para los usuarios que interactúan con el sistema.
- Obtener con la ejecución del sistema por el cliente los cambios y mejoras necesarias con el objetivo de lograr un sistema que cumpla con todas las necesidades que puedan existir.
- Extender el sistema para que pueda ser utilizado no solo en un banco de sangre perteneciente a un hospital sino que pueda ser desplegado en cualquier institución de otro tipo.

REFERENCIAS BIBLIOGRÁFICAS

1. [Online] [Cited: octubre 28, 2008.] <http://www.sociotecnologia.com/Fotos/ProyectoIntegracionSI.doc>.
2. [Online] [Cited: octubre 18, 2008.] <http://www.facmed.unam.mx/emc/computo/ssa/HIS/hisindex.htm>.
3. [Online] [Cited: octubre 18, 2008.]
http://www.doyma.es/revistas/ctl_servlet?_f=7014&articuloid=13032592.
4. **ALBET**. Conceptos del Módulo Banco de Sangre alas HIS. 2008.
5. **Página oficial de Virtus Group, empresa que realiza soluciones informáticas**. [Online] [Cited: noviembre 22, 2009.] <http://www.virtus.com.mx/xblood>.
6. **Página oficial de TIMSA, empresa que ofrece soluciones para la administración del Sector Salud**. [Online] <http://www.timsa.com.mx/Paginas/10timhemob.htm>.
7. **Página oficial de STARTNET, empresa que se dedica principalmente al desarrollo informático**. [Online] <http://www.stnt.com.ar/sistema-gestion-bancos-sangre.asp>.
8. **Softel**. *Propuesta para la Gestión de los Donantes de Sangre y Distribución de Sangre y Componentes Sanguíneos*.
9. **López Mastrapa, Henry Yordán**. Propuesta de arquitectura del sistema de registro de visitantes en la UCI. [Online] julio 2008. <http://biblioteca.uci.cu/sbd/biuci/index.html>.
10. [Online] [Cited: enero 19, 2009.] <http://www.csae.map.es/csi/silice/Global71.html>.
11. [Online] http://www.docirs.cl/arquitectura_tres_capas.htm.
12. [Online] <http://www.di-mare.com/adolfo/cursos/1998-2/3-tier.htm>.
13. [Online] [Cited: febrero 02, 2009.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
14. [Online] [Cited: febrero 02, 2009.] <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>.
15. [Online] [Cited: febrero 02, 2009.] <http://www.programacionweb.net/articulos/articulo/?num=505>.
16. **Sun Microsystems**. [Online] [Cited: enero 29, 2009.] <http://java.sun.com/javaee/javaserverfaces/>.
17. **Desarrollo Web**. [Online] [Cited: enero 29, 2009.]
<http://www.desarrolloweb.com/articulos/2380.php>.
18. **ALBET**. Documento de Arquitectura de Software alas HIS. 2008.
19. **Maestros del Web**. [Online] [Cited: febrero 22, 2009.]
<http://www.maestrosdelweb.com/editorial/ajax/>.
20. Adictos al trabajo. [En línea] [Citado el: 23 de febrero de 2009.]
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
21. Desarrollo en Web. [Online] [Cited: febrero 23, 2009.]
<http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.
22. La web del programador. [Online] [Cited: febrero 26, 2009.]
<http://www.lawebdelprogramador.com/diccionario/mostrar.php?letra=X>.
23. **ALBET**. Documento de Arquitectura alas HIS. p. 11.
24. RedHat. [Online] [Cited: marzo 3, 2009.] http://www.redhat.com/docs/manuals/jboss/jboss-eap-4.2/doc/seam/Seam_Reference_Guide/A_complete_Seam_application_the_Hotel_Booking_example-The_Seam_UI_control_library.html.
25. [Online] [Cited: marzo 04, 2009.] <http://blog.wixel.es/2008/07/desarrollos-con-jsf-y-jboss-seam-i-introduccion/>.
26. **Java en Castellano**. [Online] [Cited: marzo 25, 2009.]
<http://www.programacion.com/java/noticia/1342/>.

27. **JBOSS**. [Online] [Cited: marzo 25, 2009.] <http://www.jboss.com/products/jbpm/>.
28. [Online] [Cited: marzo 29, 2009.] <http://salaboy.wordpress.com/2008/03/02/algunos-conceptos-de-jboss-jbpm/>.
29. **Bauer, Christian and King, Gavin**. *Java Persistence with Hibernate*. p. 67 : s.n., 2007 .
30. **Sun Microsystems**. [Online] [Cited: abril 03, 2009.] <http://java.sun.com/products/ejb/>.
31. [Online] [Cited: abril 03, 2009.] <http://www.elholgazan.com/2007/08/jpa-java-persistence-api.html>.
32. [Online] http://java.com/es/download/whatis_java.jsp.
33. [Online] <http://www.osalt.com/es/jboss>.
34. Soluciones y propuestas Rational. [Online] Argentina. <http://www.rational.com.ar/herramientas/rup.html>.
35. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James**. *El Proceso Unificado del Desarrollo de Software*.
36. **BOOCH, GRADY, RUMBAUGH, JAMES and JACOBSON, IVAR**. *El lenguaje Unificado de Modelado*. 2000.
37. [Online] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
38. [Online] <http://ictnet.es/2007/sobre-la-notacion-bpmn-business-process-modeling-notation-en-la-suite-auraportal-bpms>.
39. [Online] <http://www.slideshare.net/david.motta/modelo-del-negocio-con-rup-y-uml-parte-1>.
40. [Online] <http://data2max.com/ingenieria/658us672pmbwi.pdf>.

BIBLIOGRAFÍA

Adictos al trabajo. Citado el: 23 de febrero de 2009.]

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.

ALBET. Conceptos del Módulo Banco de Sangre alas HIS. 2008.

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

Allen, Dan. Seam in Action. Publicado: September, 2008 ISBN: 1933988401.

Bauer, Christian. Java Persistence with Hibernate. Publicado: Diciembre 2006 ISBN: 1-932394-88-5

BOOCH, GRADY, RUMBAUGH, JAMES and JACOBSON, IVAR. *El lenguaje Unificado de Modelado.* 2000.

Cited: abril 19, 2009 <http://www.di-mare.com/adolfo/cursos/1998-2/3-tier.htm>.

Cited: enero 19, 2009. <http://www.csae.map.es/csi/silice/Global71.html>.

Cited: febrero 02, 2009. <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>.

Cited: febrero 02, 2009. <http://www.proactiva-calidad.com/java/patrones/mvc.html>.

Cited: febrero 02, 2009. <http://www.programacionweb.net/articulos/articulo/?num=505>.

Cited: marzo 04, 2009. <http://blog.wixel.es/2008/07/desarrollos-con-jsf-y-jboss-seam-i-introduccion/>.

Cited: marzo 19, 2009 http://www.docirs.cl/arquitectura_tres_capas.htm.

Cited: marzo 29, 2009. <http://salaboy.wordpress.com/2008/03/02/algunos-conceptos-de-jboss-jbpm/>.

Cited: abril 03, 2009. <http://www.elholgazan.com/2007/08/jpa-java-persistence-api.html>.

Cited: mayo 03, 2009. http://java.com/es/download/whatis_java.jsp.

Cited: mayo 09, 2009. <http://www.osalt.com/es/jboss>.

Cited: mayo 08, 2009 <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.

Cited: mayo 09, 2009 <http://ictnet.es/2007/sobre-la-notacion-bpmn-business-process-modeling-notation-en-la-suite-auraportal-bpms>.

Cited: mayo 09, 2009 <http://www.slideshare.net/david.motta/modelo-del-negocio-con-rup-y-uml-parte-1>.

Cited: octubre 18, 2008. http://www.doyma.es/revistas/ctl_servlet?_f=7014&articuloid=13032592.

Cited: octubre 18, 2008. <http://www.facmed.unam.mx/emc/computo/ssa/HIS/hisindex.htm>.

Cited: octubre 28, 2008. <http://www.sociotecnologia.com/Fotos/ProyectoIntegracionSI.doc>.

Cumberlidge, Matt. Business Process Management with JBoss jBPM.

Desarrollo en Web Cited: febrero 23, 2009. <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>.

Desarrollo Web. Cited: enero 29, 2009. <http://www.desarrolloweb.com/articulos/2380.php>.

Forman, Ira and Forman, Nate. Java Reflection in Action. Publicado: 2004 ISBN: 1932394184.

Jacobson, Ivar, Booch, Grady and Rumbaugh, James. *El Proceso Unificado del Desarrollo de Software*.

Java en Castellano. Cited: marzo 25, 2009. <http://www.programacion.com/java/noticia/1342/>.

JBOSS. Cited: marzo 25, 2009. <http://www.jboss.com/products/jbpm/>.

JBoss Server Manager Reference Guide Publicado: April 2008.

jBPM Tools Reference Guide Publicado: April 2008.

Katz, Max. Practical richfaces. Publicado: Diciembre, 2008 ISBN10: 1-4302-1055-9 ISBN13: 978-1-4302-1055-9

La web del programador. Cited: febrero 26, 2009.

<http://www.lawebdelprogramador.com/diccionario/mostrar.php?letra=X>.

López Mastrapa, Henry Yordán. Propuesta de arquitectura del sistema de registro de visitantes en la UCI. julio 2008. <http://biblioteca.uci.cu/sbd/biuci/index.html>.

Maestros del Web. Cited: febrero 22, 2009. [<http://www.maestrosdelweb.com/editorial/ajax/>].

Página oficial de STARTNET, empresa que se dedica principalmente al desarrollo informático.

<http://www.stnt.com.ar/sistema-gestion-bancos-sangre.asp>.

Página oficial de TIMSA, empresa que ofrece soluciones para la administración del Sector Salud. [Online] <http://www.timsa.com.mx/Paginas/10timhemob.htm>.

Página oficial de Virtus Group, empresa que realiza soluciones informáticas. [Online] [Cited: noviembre 22, 2009.] <http://www.virtus.com.mx/xblood>.

RedHat. Cited: marzo 3, 2009. http://www.redhat.com/docs/manuals/jboss/jboss-eap-4.2/doc/seam/Seam_Reference_Guide/A_complete_Seam_application_the_Hotel_Booking_example-The_Seam_UI_control_library.html.

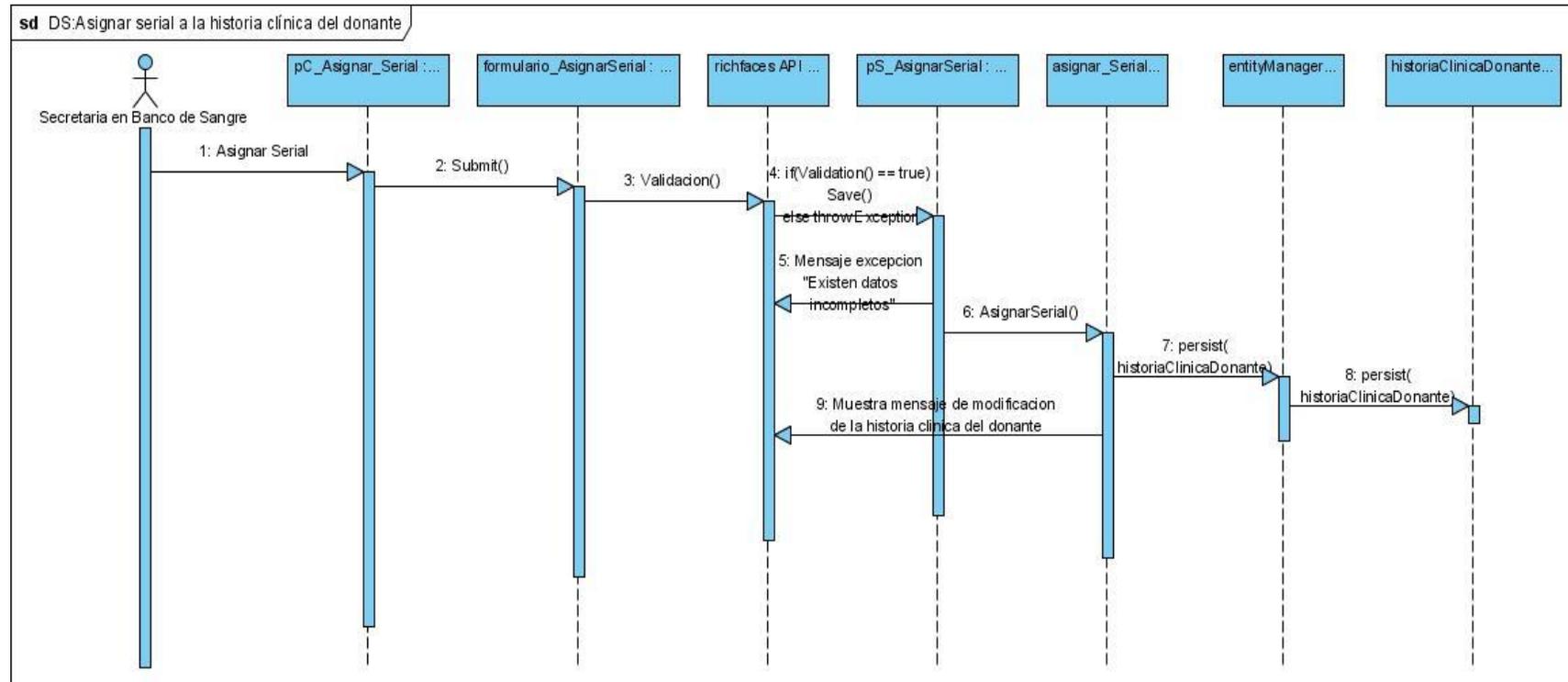
Softel. *Propuesta para la Gestión de los Donantes de Sangre y Distribución de Sangre y Componentes Sanguíneos.*

Soluciones y propuestas Rational. Argentina. <http://www.rational.com.ar/herramientas/rup.html>.

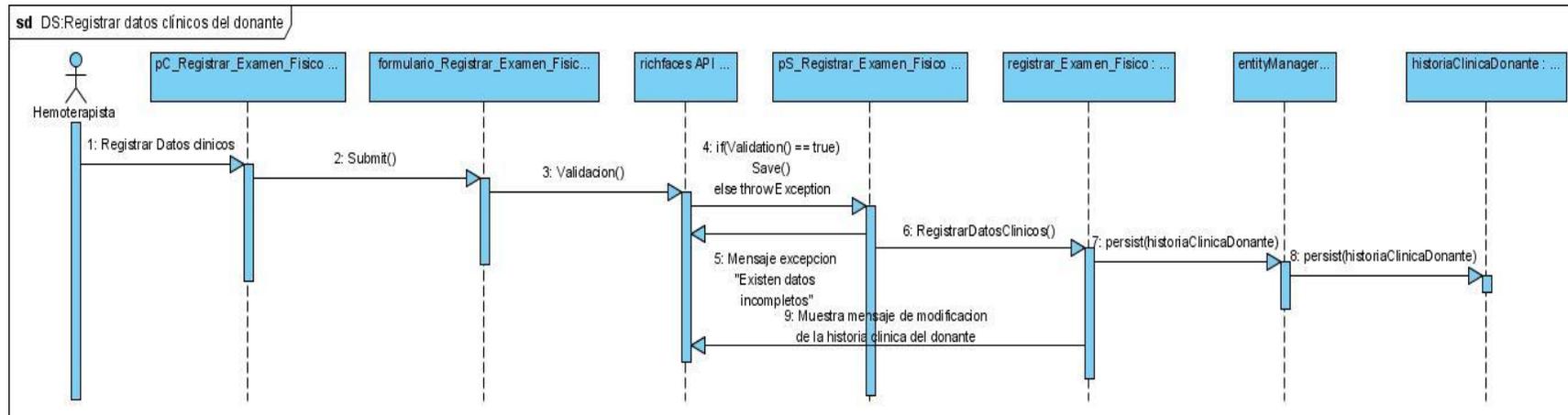
Sun Microsystems. Cited: enero 29, 2009. <http://java.sun.com/javaee/javaserverfaces/>.

ANEXOS

1. Diagramas de secuencia.



Anexo 1.1. Diagrama de secuencia CU Asignar serial a la historia clínica del donante.



Anexo 1.2. Diagrama de secuencia CU Registrar datos clínicos del donante.

2. Pantallas del sistema.

Seleccionar paciente Q Buscar...

Criterios de búsqueda <<

Cédula:

Listado de pacientes

Foto	Nombre	Primer apellido	Segundo apellido	Cédula	
	Andy	Diaz	Allende	87031202268	 Sele
	Manolo	Lama	Hernandez	90101025874	
	Ernesto	Garcia	Orduñez	92030127867	

⏪ ⏩

Anexo 2.1. Seleccionar paciente.

Crear Historia Clínica del donante Q Buscar...

Datos generales del donante No.H.C.: 8979216

 Nombre: Andy Cédula: 87031202268
Primer apellido: Diaz Fecha de nacimiento: 1987-03-12
Segundo apellido: Allende Sexo: Masculino

Otros datos del donante

Antecedentes: Sin inconvenientes Edad: 22
Dirección: Procedencia:
Estado:

[Seleccionar donante](#)

Anexo 2.2. Crear Historia clínica del donante.

Registrar datos clínicos del donante

Hemoterapista

Nombre: Gerardo

Área: Banco de Sangre

Datos generales del donante

No.H.C.: 591689

Nombre: Manolo Cédula: 90101025874

Primer apellido: Lama Fecha de nacimiento: 1990-10-10

Segundo apellido: Hernandez Sexo: Masculino

Otros datos del donante

Antecedentes: Sin inconvenientes Edad: 19

Dirección: Las Mercedes Estado: Vargas

Seleccionar donante

Examen físico

Peso: kg Hemoglobina: g/dL

Presión arterial

Diastólica: mmHg Sistólica: mmHg

Pulso

Valor: ppm Características:

Temperatura: °C Aceptación:

Motivo de Diferido o Descartado:

Anexo 2.3. Registrar datos clínicos

Asignar serial a la Historia Clínica del donante Q Buscar...

Datos generales del donante No.H.C.: 14785223


 Nombre: Manolo Cédula: 90101025874
 Primer apellido: Lama Fecha de nacimiento: 1990-10-10
 Segundo apellido: Hernandez Sexo: Masculino

Otros datos del donante

Antecedentes: Sin inconvenientes Edad: 19

Datos generales del paciente No.H.C.: 12345678


 Nombre: María Emilia Cédula: 93718246
 Primer apellido: García Fecha de nacimiento: 1980-05-20
 Segundo apellido: Cruz Sexo: Femenino

Otros datos del paciente

Ubicación: Hospitalización 2 Edad: 29

[Seleccionar paciente](#)

Datos de la donación

Serial: Tipo de Donación:

Anexo 2.4. Asignar serial a la Historia clínica del donante.

Registrar donación Q Buscar...

Datos generales del donante **No.H.C.: 14785223**

Nombre: Manolo Cédula: 90101025874

Primer apellido: Lama Fecha de nacimiento: 1990-10-10

Segundo apellido: Hernandez Sexo: Masculino

Otros datos del donante

Antecedentes: Sin inconvenientes Edad: 19

Dirección: Las Mercedes Estado: gehos.bancosangre.entity.auto.Estado_bancosangre@87ef26

[Seleccionar donante](#)

Datos de las bolsas de componentes

Bolsas de Componentes	
Segmento	
111	
112	

Segmento:

Anexo 2.5. Registrar donación.

GLOSARIO

Aféresis

La aféresis es la técnica mediante la cual se separan los componentes de la sangre, siendo seleccionados los necesarios para su aplicación en medicina y devueltos al torrente sanguíneo el resto de componentes. La finalidad de la aféresis es la extracción de un componente sanguíneo destinado a la transfusión o para el tratamiento de algunas enfermedades que precisen la eliminación de un componente patológico de la sangre.

API

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

Bean

Es un componente software que tiene la particularidad de ser reutilizable

Desglosar

Se refiere a la acción de retirarle a un paciente una bolsa de componente sanguíneo anteriormente asignado a él.

Framework

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Hemoterapista

Es el Profesional de la Salud que realiza la obtención, estudio inmunohematológico y serológico, procesamiento manual o mecánico, conservación y transfusión de la sangre humana, componentes, derivados y productos recombinantes de acuerdo con las técnicas al más alto nivel nacional e internacional.

HTML

(HyperText Markup Language), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP

(HyperText Transfer Protocol), protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la Red Global Mundial.

Java Script

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.

Java Server Pages (JSP)

Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

LPT

Puerto paralelo para conectar periféricos a una computadora.

Plain Old Java Object (pojo)

Enfatiza el uso de clases simples y que no dependen de un framework en especial.

Plasma

Parte líquida de la sangre en la que se encuentran las células. Su peso específico es de 1,026 gr. La composición: agua (cerca del 95%), proteínas, lípidos, glúcidos, potasio, sodio, calcio, bicarbonato, hormonas. En el aspecto transfusional el plasma obtenido después de una centrifugación es muy útil porque puede ser usado como tal o fraccionado en sus constituyentes esenciales: albúmina, fibrinógeno, globulinas o inmunoglobulinas, factor VIII y PPSB, todos inyectables a gran concentración en enfermos con déficits de uno o varios de los componentes mencionados.

Plasma Fresco Congelado

Es el plasma separado de la sangre de un donante por centrifugación o aféresis y congelado antes de las seis horas de la extracción. La congelación debe realizarse a una temperatura inferior a -30°C y debe estar completada en un plazo inferior a seis horas.

Plasmaferésis

Consiste en la obtención del componente sanguíneo plasma a través del proceso de la aféresis.

El fenotipo está determinado fundamentalmente por el genotipo, o por la identidad de los alelos, los cuales, individualmente, cargan una o más posiciones en los cromosomas. Algunos fenotipos están determinados por los múltiples genes, y además influenciados por factores del medio. De esta manera, la identidad de uno, o de unos pocos alelos conocidos, no siempre permite una predicción del fenotipo.

Pruebas de Compatibilidad

Pruebas realizadas para verificar la compatibilidad de un componente sanguíneo con un paciente determinado.

TCP/IP

Es un conjunto de protocolos de red que permiten la transmisión de datos entre redes de computadoras.

USB

El Universal Serial Bus (USB), es un puerto que sirve para conectar periféricos a una computadora

Web

Web, (World Wide Web) o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet.

XML

Extensible **Markup Language** o lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium.