

Universidad de las Ciencias Informáticas
Facultad 7



TÍTULO: Desarrollo de los procesos para la gestión de la información del trasplante de riñón para el Sistema de Trasplante.

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

Autores:

Dailín Reyes Perdomo
Juan Miguel Moreno de León

Tutoras:

Ing. Claudia Fuentes Escobar
Ing. Angélica de la Caridad Vázquez Rúa

Ciudad de la Habana, junio 2013

“Año 55 de la Revolución”



"La base de un desarrollo impetuoso en los años futuros debe basarse en una ciencia cada vez más desarrollada".

Ernesto Guevara de la Serna.

Declaración de Auditoría

Declaración de autoría:

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 7 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____ .

Firma del autor:

Dailín Reyes Perdomo

Firma del autor:

Juan Miguel Moreno de León

Firma de la Tutora:

Ing. Claudia Fuentes Escobar

Firma de la Tutora:

Ing. Angélica de la Caridad Vázquez Rúa

Agradecimientos

De Dailín

A mis padres por ayudarme y apoyarme cada vez que lo necesito.

A mi hermana, por ser siempre muy especial conmigo y por aconsejarme en todo momento de mi vida.

A mi hermanito Pedrito que me quiere mucho.

A mi tía Alba por confiar en mí y llevarme por el camino correcto.

A mi tía Ada por ayudarme cuando más lo necesito.

A mi compañero de tesis, Juan Miguel, por luchar junto a mí para poder realizar nuestra tesis.

A mis tutoras Claudia Fuentes Escobar y Angélica de la Caridad Vázquez Rúa por ser siempre muy atentas y ayudarme en el transcurso de la tesis.

A mi novio Luis Miguel por ser tan comprensivo conmigo, apoyarme en todo momento y estar siempre a mi lado.

A mi suegra Magalís Milián por quererme como soy y por ayudarme siempre.

Juan Miguel

De Dailín

A mis padres que siempre quieren lo mejor para mí.

A mi hermana que me quiere mucho, me apoya y me aconseja.

A mi tía Alba que siempre está al tanto de mis cosas.

A mi novio que siempre me consiente y me da tanto amor y comprensión.

De Juan Miguel

Datos de contacto

Ing. Claudia Fuentes Escobar

Especialista, graduada en el año 2010 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Pertenece al departamento Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM), donde se desempeña como Programadora.

Correo electrónico: cfuentes@uci.cu

Ing. Angélica de la Caridad Vázquez Rúa

Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2011. Actualmente posee la categoría de Recién Graduada en Adiestramiento y se desempeña en el Departamento de Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM) como programadora del proyecto Nefrología.

Correo electrónico: alvarez@uci.cu

Resumen

El proceso de gestión de la información del trasplante de riñón en las instituciones hospitalarias, requiere del procesamiento de grandes volúmenes de información. Las actividades relacionadas con dicha gestión son realizadas por los especialistas de forma manual. Esto implica que la información pueda encontrarse duplicada o redundante, expensa al extravío y/o deterioro y que no se pueda obtener en tiempo real ni con la exactitud requerida, disminuyendo la calidad de la atención médica a los pacientes. A raíz de estos problemas se evidencia la necesidad de informatizar este proceso.

El presente trabajo tiene como objetivo desarrollar una solución de software que agilice el proceso de selección donante – receptor del trasplante renal, a partir de un donante cadáver, en las instituciones hospitalarias cubanas. Para el desarrollo del sistema se utilizó como metodología de desarrollo el Proceso Unificado de Desarrollo (RUP), como lenguaje de modelado el Lenguaje Unificado de Modelado (UML), como framework para desarrollo de aplicaciones web Java Server Faces (JSF), como sistema gestor de Base de Datos PostgreSQL versión 8.4, como servidor de aplicación JBoss Server versión 4.2.2 GA, como herramienta de desarrollo Eclipse, que es una plataforma de desarrollo extensible y adaptable a cualquier lenguaje, Visual Paradigm para el análisis y diseño y como lenguaje de programación Java, que es robusto, seguro y multiplataforma.

Palabras Clave

Gestión, Sistema, Trasplante.

Tabla de contenidos

Introducción	1
Capítulo 1: Fundamentación Teórica.....	6
1.1. Conceptos básicos asociados al dominio del problema	6
1.2. Sistemas de gestión de trasplante a nivel internacional y nacional	8
1.2.1. CARLA	8
1.2.2. SINTRA.....	9
1.2.3. SIUL.....	10
1.2.4. SURUS.....	11
1.2.5. BIOTCEL.....	12
1.2.6. Sistema Nacional de Trasplante Renal.....	12
1.3. Descripción de tecnologías, herramientas y metodología a utilizar	13
1.3.1. Metodología de desarrollo	15
1.3.2. Tecnologías utilizadas en el proceso de desarrollo	16
1.3.2.1. Lenguaje de Programación.....	19
1.3.2.2. Lenguaje Unificado de Modelado.....	20
1.3.3. Tecnologías Horizontales.....	21
1.3.4. Herramientas de desarrollo.....	21
Capítulo 2: Características del Sistema	25
2.1. Flujo actual de procesos involucrados en el campo de acción.....	25
2.1.1. Gestionar receptor.....	25
2.1.2. Gestionar donante.....	25
2.1.3. Gestionar reporte estadístico.....	25
2.2. Información que se maneja.....	26
2.3. Objeto de automatización	26
2.4. Modelación de Negocio.....	27
2.4.1. Actores y trabajadores del negocio.....	27
2.4.2. Diagramas de procesos del negocio	28

Tabla de contenidos

2.5. Especificación de los requisitos del software	35
2.5.1. Requisitos funcionales	36
2.5.2. Requisitos no funcionales	37
2.6. Modelo de casos de uso del sistema	39
2.6.1. Diagrama de casos de uso	40
2.6.2. Descripción Textual de los Casos de Uso	42
Capítulo 3: Diseño del Sistema	51
3.1. Descripción de la arquitectura	51
3.2. Modelo de diseño	52
3.2.1. Diagrama de clases del diseño	52
3.3. Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados	59
Capítulo 4: Implementación	60
4.1. Modelo de datos	60
4.1.1. Descripción de las tablas	61
4.2. Implementación	67
4.2.1. Diagrama de componentes	68
4.2.2. Diagrama de despliegue	70
4.3. Tratamiento de errores	71
4.4. Seguridad	72
4.5. Estrategias de codificación. Estándares y estilos a utilizar	73
Conclusiones	77
Recomendaciones	78
Referencias Bibliográficas	79
Bibliografía	84
Anexos	87

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) son un conjunto de disciplinas científicas, tecnológicas y de ingeniería, utilizadas en el manejo y procesamiento de la información mediante el uso de computadoras y aplicaciones. Estas facilitan la interacción de los hombres con las máquinas, con el objetivo de transformar, almacenar, gestionar, proteger y difundir los datos necesarios para cualquier actividad humana. Por esta razón, la informatización de la sociedad se hace cada vez más necesaria en todo el mundo y llevarla a cabo, es una tarea que requiere de grandes esfuerzos.

Cuba está consciente de que para mejorar su economía, lograr una sociedad más eficaz, eficiente y competitiva, debe aplicar las TIC en todos sus ámbitos. Este proceso en la esfera de la salud es hoy un hecho. El Ministerio de Salud Pública (MINSAP), es el encargado de dirigir, ejecutar y controlar la aplicación de la política de informatización en esta rama de la sociedad, priorizando los diferentes niveles de atención: primaria, secundaria y terciaria.

La Atención Terciaria cuenta con los servicios intrahospitalarios de alta especificación, los cuales están destinados a la ejecución de intervenciones quirúrgicas y atenciones de mayor complejidad, que requieran hospitalización. Dentro de los procedimientos quirúrgicos llevados a cabo en este nivel de atención se encuentran los trasplantes de órganos, cuyo fin es erradicar o retrasar el progreso de una enfermedad. Actualmente los trasplantes más realizados pertenecen al servicio de Nefrología, especialidad que se encarga del estudio de la estructura y función renal, tanto en la salud como en la enfermedad, que incluye la prevención y tratamiento de las enfermedades que afectan al riñón. (1)

Desde el inicio de 1970 hasta la fecha, Cuba ha realizado más de 5 100 trasplantes renales, con resultados de sobrevida similares a países del primer mundo. El enfermo con insuficiencia renal puede recibir el riñón de un donante vivo (generalmente un familiar) o de un donante cadáver recién fallecido. Este riñón único suple todo el trabajo que los dos riñones insuficientes no pueden realizar. En el país el 94 por ciento de estas operaciones se hacen con donantes cadáver. Gracias a un exitoso programa de obtención de órganos, la isla posee la tasa de donación más elevada de Centroamérica y el Caribe y es la cuarta en Latinoamérica después de Uruguay, Brasil y Argentina. Además, está ubicada entre los de más baja tasa de rechazo.

La nación tiene 2 700 pacientes con tratamiento de hemodiálisis y cada año crece en un 10 por ciento los enfermos que lo necesitan. Se ha está demostrado que el trasplante renal es tres veces más económico que mantener al enfermo en diálisis o con riñón artificial. El doctor Alexander Mármol Sóñora, especialista de segundo grado en Nefrología y funcionario de la Organización Nacional de Trasplantes (ONT) del MINSAP, explicó que según estudios realizados cada paciente le cuesta al país anualmente 20 mil dólares mantenerlo en diálisis, proceder que se hace generalmente en días alternos. Para ese fin existen 47 hospitales que prestan el servicio, incluso en zonas montañosas. (2)

El trasplante renal no siempre resulta una solución definitiva, pues existe la posibilidad de que el organismo rechace el nuevo riñón, aunque exista un alto grado de compatibilidad. Por lo general, del 75 al 80 por ciento de los trasplantes procedentes de donantes cadáveres, siguen funcionando al año de la operación. Los trasplantes procedentes de donantes vivos familiares del paciente suelen funcionar mejor que los trasplantes de donantes cadáveres, debido normalmente a un mayor grado de compatibilidad. El paciente trasplantado debe ser medicado de por vida con fármacos inmunosupresores que contribuyen a evitar el rechazo del órgano. Estos conllevan a efectos secundarios y el más importante consiste en el debilitamiento del sistema inmunitario del trasplantado favoreciendo así el desarrollo de infecciones. En ocasiones, este tipo de medicación no logra evitar el rechazo y el paciente debe volver a someterse a algún tipo de diálisis y a esperar, probablemente, otro trasplante. (3)

Para llevar a cabo el trasplante de riñón las unidades de hemodiálisis, de diálisis peritoneal y las consultas de infección renal crónica (IRC) del país envían mensualmente al Centro Coordinador Nacional de Trasplante los datos de los receptores aptos mediante correo electrónico o vía telefónica. Dicho centro ubica a estos pacientes en la lista nacional de pacientes a trasplantar teniendo en cuenta el por ciento de hipersensibilidad, si es niño y la prioridad establecida por el especialista. Una vez recibido un donante cadáver se procede a buscar la compatibilidad de este con los receptores, teniendo en cuenta el Grupo y Factor, edad, peso, sexo, tiempo en diálisis y el tipaje de Antígenos Leucocitarios Humanos (HLA); a cada uno de estos aspectos se le atribuye una puntuación, siendo el receptor idóneo el que mayor puntaje acumule. Después de obtener los receptores potenciales se procede a realizar la prueba Cross-Match con el suero extraído de la sangre de dichos pacientes, seleccionando los que posean resultado negativo.

Anualmente, los especialistas generan reportes que reflejan la cantidad de receptores por hospital de

procedencia y la cantidad de donantes por centro. Las actividades mencionadas anteriormente, son realizadas por los especialistas de forma manual. Esto implica que resulte más engorrosa la búsqueda del receptor idóneo para un donante. Además, la información que se gestiona puede encontrarse duplicada o redundante, expensa al extravío y/o deterioro y puede que no se obtenga en tiempo real ni con la exactitud requerida, disminuyendo la calidad de la atención médica a los pacientes. A raíz de estos problemas se evidencia la necesidad de informatizar este proceso.

Por su parte, la Universidad de las Ciencias Informáticas (UCI) tiene entre sus principales objetivos producir aplicaciones y servicios informáticos que sirvan de soporte a la industria cubana de la informática, contribuyendo así a la informatización de su sociedad. Para ello cuenta con varios centros productores de software¹, dentro de los que se destaca el Centro de Informática Médica (CESIM). Este centro cuenta con el departamento Sistemas Especializados en Salud (SES), que desarrolla una línea de investigación denominada Sistema de Trasplante. Esta aplicación tiene como fin automatizar los procesos de gestión de la información generados durante los diferentes tipos de trasplantes desarrollados en el país. Actualmente, este sistema no gestiona lo referente al trasplante renal, siendo uno de los servicios más priorizados en el país.

Partiendo de lo antes planteado se identifica como **problema a resolver**: ¿Cómo agilizar el proceso de selección donante – receptor del trasplante renal en las instituciones hospitalarias cubanas?

Se define como **objeto de estudio** de la investigación: El proceso de selección donante – receptor del trasplante renal en las instituciones hospitalarias.

El **campo de acción** se centra en: El proceso de selección donante – receptor del trasplante renal en las instituciones hospitalarias cubanas, a partir de un donante cadáver.

Para darle solución al problema identificado, se define como **objetivo general**: Desarrollar una solución de software que agilice el proceso de selección donante – receptor del trasplante renal, a partir de un donante cadáver, en las instituciones hospitalarias cubanas.

¹ Software: Es un conjunto de programas, documentos, procedimientos, y rutinas asociados con la operación de un sistema de cómputo.

Para dar cumplimiento al objetivo trazado se prevee la realización de las siguientes **tareas de la investigación**:

1. Realización de un análisis crítico y valorativo de los sistemas de trasplante que gestionen información referente al trasplante de riñón a nivel internacional y nacional.
2. Caracterización de la metodología, las técnicas, tecnologías y herramientas a utilizar para la solución del problema.
3. Elaboración de los artefactos propuestos por la metodología seleccionada.
4. Implementación de los procesos para la gestión de la información del trasplante de riñón, aplicando las pautas de diseño y siguiendo las necesidades de funcionamiento establecidas en la especificación de requisitos de software.

En el transcurso y desarrollo de esta investigación científica se utilizaron los siguientes métodos teóricos y empíricos:

Teóricos:

- ✓ Histórico – Lógico: para conocer las investigaciones relacionadas con el trasplante de riñón así como su evolución desde su surgimiento hasta la actualidad.
- ✓ Inductivo – deductivo: para el análisis de los sistemas existentes que gestionan la información referente al trasplante de riñón a nivel internacional y nacional.
- ✓ Análisis – Síntesis: para el análisis de la documentación relacionada con el trasplante de riñón y se determinaron los conceptos relacionados con el objeto de estudio.
- ✓ Modelación: para la realización de los diagramas correspondientes a las fases: modelación del negocio, requisitos, diseño e implementación de la solución propuesta.

Empíricos:

- ✓ Entrevista: a los especialistas de trasplante de riñón del Instituto Nacional de Nefrología, con el fin de lograr un mayor entendimiento del negocio en las instituciones hospitalarias cubanas. Ver Anexos para una mejor comprensión de la misma.

El desarrollo de la solución propuesta proporcionará los siguientes beneficios:

- ✓ Permitirá a los especialistas una mejor gestión de la información referente a los pacientes y los donantes.
- ✓ Facilitará el proceso de selección Donante-Receptor.
- ✓ Servirá de soporte a la toma de decisiones de los especialistas, proporcionándole reportes estadísticos.
- ✓ Posibilitará a los especialistas un mayor control del estado del paciente antes del trasplante.
- ✓ Aumentará la calidad de la atención al paciente.

El presente trabajo está estructurado en cuatro capítulos que se describen a continuación:

Capítulo 1: Fundamentación Teórica: Investigación sobre el estado del arte de los sistemas de gestión de trasplante renal a nivel internacional y nacional relacionado con el campo de acción. Descripción de las tecnologías y herramientas a utilizar.

Capítulo 2: Características del sistema: Descripción de las funcionalidades del sistema, tomando como punto de partida los procesos del negocio.

Capítulo 3: Diseño del sistema: Modelación y construcción de la estructura de la aplicación.

Capítulo 4: 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 procesos hospitalarios asociados al área en cuestión.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se exponen los conceptos básicos referentes al campo de acción que servirán de apoyo para un mejor entendimiento del mismo. Además, se realiza un análisis crítico y valorativo de los sistemas de gestión de trasplante renal a nivel internacional y nacional y se describe la metodología, las tecnologías, plataformas y herramientas de software asimiladas para el desarrollo del sistema.

1.1. Conceptos básicos asociados al dominio del problema

El cuerpo humano está compuesto por 21 **órganos**, los cuales son estructuras corporales de tamaño y forma característicos. Están formados por distintas clases de tejidos como pueden ser músculos o membranas. Los órganos generalmente funcionan dentro de sistemas o aparatos, compenetrados con otros órganos para la realización de una función. Así por ejemplo en el Sistema Excretor o Urinario se encuentran los riñones. (4)

Los **riñones** son considerados los principales químicos del organismo. Son dos glándulas de color rojo oscuro colocadas simétricamente a los lados de la columna vertebral, en la región lumbar. Miden 10 cm de largo y pesan unos 150 gr cada uno. Están envueltos por una capa de grasa, la cápsula adiposa renal, que está cubierta por delante y por detrás por una capa de tejido conjuntivo: la fascia de Gerota. A pesar de ser tan pequeños, procesan y purifican toda la sangre cada 50 minutos; alrededor de 1 700 litros de sangre pasan por ellos al cabo del día. Son los encargados de eliminar de la sangre los productos nocivos y, junto al resto del sistema urinario, eliminarlos del cuerpo mediante la orina. También elaboran sustancias que ayudan al control de la presión arterial y que regulan la formación de los glóbulos rojos. Cuando los riñones fallan, es necesario un tratamiento para sustituir el trabajo que ellos hacían. A menos que se realice un trasplante de riñón, el paciente debe someterse a un tratamiento de diálisis.

Existen dos tipos principales de **diálisis**: hemodiálisis y diálisis peritoneal. Ambos tipos filtran la sangre para eliminar los desechos peligrosos del cuerpo y el exceso de sal y agua. La hemodiálisis se logra con un aparato. La diálisis peritoneal usa la membrana que recubre el abdomen, llamada membrana peritoneal, para filtrar la sangre. Cada tipo tiene riesgos y beneficios. También requiere que siga una dieta especial. El médico puede ayudarlo a decidir el mejor tipo de diálisis para su caso. Este tratamiento no es curativo y es de por vida, ya que no modifica la lesión que se produjo en los riñones. Además, permite

Capítulo 1: Fundamentación Teórica

mejorar los síntomas y vivir por más tiempo, ofreciendo la posibilidad futura de realizar un trasplante renal. (5)

Un **trasplante renal** es una intervención quirúrgica, que se realiza en un centro específicamente autorizado, por un equipo con experiencia, mediante el cual se implanta un riñón de otra persona en su cuerpo. Es la mejor opción de tratamiento para aquellos pacientes que precisan un tratamiento sustitutivo de la función renal de por vida y el que más se asemeja a una función renal normal. El riñón puede provenir de un donante fallecido o de uno vivo. Los familiares o las personas que no estén relacionadas pero que cumplan con los requisitos de compatibilidad pueden donar uno de sus riñones. Este tipo de trasplante se denomina trasplante de donante vivo. Las personas que donan un riñón pueden vivir vidas sanas con el riñón restante. Cuando una persona se somete a esta clase de trasplante, suele recibir sólo un riñón; sin embargo, existen casos aislados en los que es posible que reciba ambos riñones de un donante fallecido. En la mayoría de los casos, los órganos enfermos se mantienen en su lugar durante el procedimiento. El riñón trasplantado se implanta en la parte inferior del abdomen en la región anterior del cuerpo. (6)

Un aspecto a tener en cuenta al realizar la lista nacional de pacientes a trasplantar en el Centro Coordinador Nacional de Trasplante es el **por ciento de hipersensibilidad**. El médico responsable le otorga un valor a cada paciente según la hipersensibilidad que este posea. Esto sucede porque muchos de los pacientes no les sirven cualquier órgano, ya que son hipersensibles ante la mayoría de los donantes posibles y pueden pasar años hasta encontrar el idóneo. Son personas que o bien ya han sido trasplantadas y necesitan de un nuevo implante o han recibido muchas transfusiones que les han generado esa mayor susceptibilidad al rechazo. Aquellas personas que han recibido muchas transfusiones o las mujeres que han tenido varios embarazos y que su cuerpo ha creado anticuerpos frente a los antígenos del feto (heredados del padre), también presentan dificultad para encontrar un órgano compatible. (7)

Para llevar a cabo el trasplante renal se realiza una prueba llamada **Cross_Match**. El Cross-Match es la búsqueda de anticuerpos preformados contra los linfocitos de un posible donante en el suero de un paciente. El receptor potencial de un trasplante alogénico puede estar "sensibilizado" contra su donante potencial, es decir, puede tener en su suero anticuerpos contra las células de éste. Estos anticuerpos

Capítulo 1: Fundamentación Teórica

suelen ser consecuencia de la respuesta del receptor a transfusiones, embarazos o trasplantes previos. Pueden estar específicamente dirigidos contra los antígenos HLA o contra cualquier otro antígeno del donante que éste no comparta con el receptor. La presencia de dichos anticuerpos en el receptor potencial, especialmente los dirigidos contra los antígenos HLA del donante, contraindica el trasplante de ciertos órganos o tejidos provenientes de ese donante. En el caso de trasplante de riñón, la presencia de anticuerpos preformados en el receptor contra los HLA del donante produce un rechazo hiperagudo con pérdida del órgano en forma inmediata. En un estudio de Cross-Match se enfrenta el suero del receptor con los linfocitos del potencial donante; para detectar la presencia de posible auto anticuerpos, se enfrenta también con los linfocitos propios del receptor. (8) Para la realización de la prueba Cross_Match los especialistas se auxilian de la Placa Terasaki, pues en sus pocillos se incuban el suero del receptor con los linfocitos del donante.

1.2. Sistemas de gestión de trasplante a nivel internacional y nacional

A continuación se presentan los sistemas de gestión de trasplante a nivel internacional que gestionan los procesos de trasplantes para el órgano riñón.

1.2.1. CARLA

El sistema CARLA es un proyecto español conjunto de la Organización Nacional de Trasplantes (ONT) y la Fundación Vodafone, desarrollado por la empresa especializada en aplicaciones tecnológicas OESIA, CARLA recoge información del sistema central de la ONT y, a través de la red 'Blackberry', los móviles con sistema 'Android', los 'iPhone' y los 'iPad', por correo electrónico o sms, la distribuye entre todos los profesionales que intervienen en el proceso de donación y trasplantes. Los coordinadores de trasplantes pueden enviar en tiempo real, datos por escrito del donante, que quedan registrados a través de una aplicación informática. En toda la cadena del trasplante los médicos disponen de todos los datos necesarios para valorar al donante y la idoneidad de los órganos. Pueden verificar el grupo sanguíneo del donante, un elemento fundamental en los trasplantes, y agilizar la búsqueda del receptor adecuado para cada órgano disponible. (9)

CARLA permite mejorar la logística y todo lo relativo a la movilización de los equipos encargados de la extracción de órganos. También prevé la posibilidad de presentar la información resumida y con datos estadísticos. Los coordinadores pueden ver esta información en la pantalla de su móvil, lo que les permite

Capítulo 1: Fundamentación Teórica

conocer fácilmente la actividad que desarrollan y compartirla con el resto de profesionales de la ONT. Además, alerta a los coordinadores de trasplantes de la existencia de un nuevo donante y de la disponibilidad de órganos, facilitando sus datos personales y sobre cómo se podría realizar el transporte del órgano.

Las nuevas aplicaciones de CARLA en su segunda fase permitirán a los profesionales contar, a las 24 horas de la donación, con un resumen sobre el destino de cada uno de los órganos de sus donantes. Los profesionales de guardia en la sede central de la ONT podrán enviar informes logísticos a los hospitales implicados en un proceso de donación y trasplantes. Además, se incorpora un registro histórico de los donantes de los últimos siete días. (10)

1.2.2. SINTRA

El Sistema Nacional de Información de Procuración y Trasplante de la República Argentina (SINTRA) es el sistema informático a cargo de la administración, gestión y fiscalización de las actividades de procuración y trasplante de órganos, tejidos y células en este país. Es un proyecto impulsado, coordinado y financiado por el INCUCAI (Instituto Nacional Central Único Coordinador de Ablación e Implante), que comenzó hacia finales de 2002 y cuyo desarrollo e implementación total concluyó a finales del 2005.

Es un Sistema Nacional de Información con escalones jurisdiccionales y regionales, integrado en todo el país, que permite el registro en tiempo real de la actividad de procuración, la gestión de pacientes con insuficiencia renal crónica terminal, listas de espera y asignación de órganos, tejidos y células con fines de implante y los resultados de los trasplantes en el país, con el fin de permitir el monitoreo y evaluación permanente.

El proyecto implementa una aplicación web la cual está organizada en 6 módulos que en su conjunto representan toda la información referida al trasplante en la República Argentina:

1. Registro Nacional de Insuficiencia Renal Crónica Terminal (IRCT).
2. Listas de pacientes en espera de órganos y tejidos
3. Registro Nacional de Procuración.
4. Registro Nacional de Trasplante.

Capítulo 1: Fundamentación Teórica

5. Registro Nacional de donantes de órganos y tejidos.
6. Registro Nacional de donantes de células progenitoras hematopoyéticas (CPH).

Cada módulo tiene su propio grupo de usuarios y su funcionalidad particular, pero todos están integrados e interrelacionados internamente. El SINTRA fue desarrollado con tecnologías web de gran confiabilidad y flexibilidad en todos sus niveles (Linux, Oracle, J2EE). El hardware ²y software utilizado fue seleccionado y adquirido para uso exclusivo y se encuentra en la sala de servidores del INCUCAI. Su diseño prioriza la confiabilidad y seguridad de la información, y se utilizan todas las medidas de necesarias para lograrlo, como la autenticación de acceso mediante cuentas de usuario y la implementación de sesión segura (la información que se intercambia entre la computadora que se conecta al sistema y los servidores del INCUCAI se codifica para evitar su alteración o inspección). Su utilización es posible desde cualquier computadora que tenga acceso a Internet, sin la necesidad de instalar o configurar nada localmente. Sin embargo, el servicio de conexión a Internet se convierte en un factor fundamental en el rendimiento y el uso del sistema. La arquitectura del sistema SINTRA fue diseñada para funcionar las 24 horas del día los 365 días del año. También tiene a su disposición una línea telefónica gratuita 0800 555 4628 con el objetivo de evacuar las dudas existentes por los usuarios. (11)

SIUL, SURUS, BIOTCEL: Estos tres sistemas son creados por la ONT de España, desarrollados con aplicaciones web. Su proceso de donación y trasplante es similar al del sistema a desarrollar en la presente investigación.

1.2.3. SIUL

El Sistema Integrado de Donación y Trasplantes es un sistema de información, integrado para toda la información generada en el desarrollo de la actividad de donación y trasplante, siendo una herramienta de soporte a la gestión diaria de los procesos. Está dividida en tres módulos o niveles como son: Central, Autonómico y Hospitalario.

² Hardware: Conjunto de los componentes que conforman la parte material (física) de una computadora.

Capítulo 1: Fundamentación Teórica

El SIUL tiene como objetivos: Constituir un registro completo de las actividades de donación y trasplante, manteniendo en todo momento la trazabilidad del proceso; disponer de una herramienta funcionalmente integradora, ya que involucra al personal multidisciplinar (coordinación, quirúrgico, administrativo, etc.), constituyendo una herramienta de uso común por todos ellos; gestionar las listas de espera de manera independiente para cada programa de trasplante; generar toda la documentación necesaria para el apoyo al proceso (listados, informes, estudios estadísticos) que contribuye a reducir los tiempos de trabajo y la carga administrativa; establecer canales y protocolos de comunicación que permitan realizar estas actividades de manera segura, siempre respetando las políticas propias de seguridad de cada una de las Comunidades Autónomas.

El SIUL cuenta con los siguientes bloques de información:

1. Donantes: Gestión de donantes facilita el control y seguimiento de todo el proceso de la donación, desde el alta en la aplicación como posible donante hasta el momento de la extracción.
2. Lista de espera: Gestión independiente de la lista de espera de cada órgano.
3. Receptores: Se recogen datos generales de los pacientes así como los asociados a los diferentes estudios que se les realicen.
4. Trasplantes: Registro de todas las intervenciones de trasplantes de órganos realizadas.
5. Gestión de ofertas: La aplicación incluirá la posibilidad de hacer el seguimiento de las ofertas de los órganos a los centros trasplantadores. (12)

1.2.4. SURUS

Sistemas de registro de trasplantes y sus correspondientes seguimientos es una aplicación web, diseñada para la recogida de información, seguimiento de donantes y los receptores de diferentes órganos: pulmón, riñón, páncreas, hígado entre otros. Consta de un Módulo diferente para cada uno de los órganos que se recoge información. SURUS cuenta con los siguientes bloques de información: Gestión de personas (donantes y receptores), gestión de trasplantes, gestión de episodios clínicos e informes.

El estudio de seguimiento lo realiza a dos niveles como son: donantes de órganos para trasplante y pacientes trasplantados: renales, hepáticos, cardiacos, pulmonares y de páncreas. Su entorno tecnológico se basa en una plataforma: Jdk 1.4, con una arquitectura multicapa y su base de datos en Oracle. (13)

Capítulo 1: Fundamentación Teórica

1.2.5. BIOTCEL

El Sistema de gestión y control del tratamiento de células y tejidos destinados a trasplantes es una aplicación web que permite la implantación de una mejora en el sistema de gestión y control del tratamiento de células y tejidos destinados a trasplantes, pudiendo, tener la traza o rastro de células y tejidos en cualquier momento de su vida útil así como su situación y disponibilidad. Además, dispone de sistemas de alarma (biovigilancia), con el fin de poder realizar un seguimiento de las incidencias detectadas en cualquier proceso de la vida útil de las células y tejidos, con posibilidad de aviso a todos los participantes del proceso y seguimiento hasta su resolución. (14)

A continuación se presenta el sistema de gestión de trasplante a nivel nacional que gestiona los procesos de trasplantes para el órgano riñón.

1.2.6. Sistema Nacional de Trasplante Renal

Este sistema permite centralizar toda la información referente al trasplante renal en el país, ayudando a la toma de decisión en cuanto a la prioridad del órgano y su posible receptor, así como la posibilidad del mantenimiento del banco de receptores y la obtención de informes estadísticos.

Los objetivos del sistema son: permitir la actualización del banco de receptores para trasplante y realizar la selección de la pareja donante receptor entre todos los pacientes aptos del país; revisar la información de los pacientes del banco de receptores en función a los parámetros de selección; facilitar una selección rápida por orden de prioridad de los receptores potenciales para recibir trasplante renal, teniendo en cuenta el análisis de parámetros por orden de importancia; configurar los parámetros de selección, posibilitando con ello la perdurabilidad del sistema; reportar información requerida y preservar esta.

El proceso de selección se basa en un filtro de 5 parámetros clínicos y geográficos:

- ✓ Enfermedades virales.
- ✓ Tipo de sangre.
- ✓ Compatibilidad.
- ✓ Programa médico.
- ✓ Grupo (zona geográfica).

Capítulo 1: Fundamentación Teórica

Luego de realizar el análisis de los sistemas se decide descartarlos: CARLA, debido a que su mayor funcionamiento radica en que cada profesional que utilice el sistema tenga un teléfono móvil con una aplicación para recibir correo electrónico, por lo que resulta muy costoso; SINTRA porque se hace indispensable un total acceso a internet; SIUL, SURUS y BIOTCEL, su arquitectura es similar a la del sistema a desarrollar, pero son sistemas que no están integrados y su adquisición sería muy costosa y el Sistema Nacional de Trasplante, porque la tecnología usada no es compatible con el Sistema de Trasplante desarrollado en el departamento.

Sin embargo, se identificaron un conjunto de requisitos comunes entre todos estos sistemas que servirán de guía para el desarrollo de la solución propuesta, dentro de los que se encuentran: listados de pacientes a trasplantar, listados de donantes, asignación de un donante a un receptor y obtención de reportes estadísticos.

1.3. Descripción de tecnologías, herramientas y metodología a utilizar

La solución propuesta es un módulo del Sistema de Trasplante y debido a que su arquitectura ya está definida, se realizó un estudio de la metodología, las herramientas y tecnologías que la conforman. A continuación se presentan las mismas.

Modelo de Capacidad y Madurez (CMMI)

CMMI, por sus siglas en inglés (Capability Maturity Model Integration), es un modelo elaborado por el Software Engineering Institute (SEI) que permite obtener un diagnóstico preciso de la madurez de los procesos relacionados con las tecnologías de la información de una organización, y describen las tareas que se tienen que llevar a cabo para mejorar esos procesos. (15) Es un conjunto de buenas prácticas que abarcan el ciclo de vida de un producto desde la concepción hasta la entrega y mantenimiento. Su propósito es ayudar a las organizaciones a mejorar sus procesos de desarrollo y el mantenimiento de sus productos.

Define diferentes procesos y actividades a ejecutar por los ingenieros de software con el objetivo de desarrollar productos con un elevado nivel de calidad. Este modelo en su representación escalonada, cuenta con cinco niveles de madurez (ver Figura 1), cada área de proceso se asocia a uno de estos niveles y a medida que la organización cumple con los procesos definidos para cada uno, alcanza el nivel

Capítulo 1: Fundamentación Teórica

de madurez de referencia. Una organización alcanza un nivel de madurez determinado cuando ha puesto en práctica todas y cada una de las áreas de proceso aplicables a ese nivel y a los niveles inferiores.

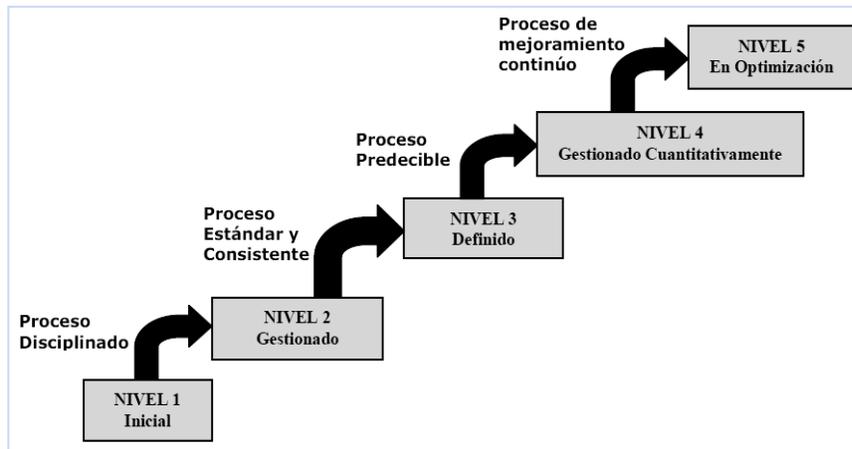


Figura 1: Niveles de CMMI.

CMMI plantea que toda organización se encuentra inicialmente en nivel 1 porque de una forma u otra ejecuta sus procesos, aunque no de una manera organizada. La UCI con la necesidad de organizar los mismos inició en el 2008 un programa de mejora de la calidad del software, adoptando para su desarrollo este modelo. Esta es una estrategia para identificar, desarrollar, implantar y mejorar sus procesos a mediano y largo plazo, brindando beneficios en cuanto a la predictibilidad de la planificación, el presupuesto, tiempo de desarrollo, productividad, satisfacción del cliente y satisfacción de los empleados. El 14 de julio del 2011 se obtuvo la certificación del nivel 2 de CMMI en 3 centros productivos. Teniendo en cuenta los resultados obtenidos, se decidió institucionalizar los procesos definidos para el nivel 2 en el resto de los 14 centros, añadiendo además las 11 áreas de proceso que propone CMMI en el nivel 3. Con la ejecución del nivel 2 se trabajó en la implementación de algunas de las prácticas que propone el nivel 3. (16)

El programa de mejora que se desarrolla actualmente en la UCI tiene como objetivo la definición e implementación de los procesos necesarios con el fin de cubrir las 7 áreas de procesos en la representación escalonada del nivel 2 del modelo: Planeación del Proyecto (PP), Monitoreo y Control de Proyecto (PMC), Administración de Requisitos (REQM), Aseguramiento de la Calidad para Procesos y Productos (PPQA), Medición y Análisis (MA), Administración de la Configuración (CM) y Administración de

Capítulo 1: Fundamentación Teórica

Acuerdos con Proveedores (SAM), se busca con ello la reducción de varios de los problemas que están vigentes en la producción de software en la Universidad. (17)

CMMI plantea un conjunto de políticas a cumplir por cada área de proceso. Por esta razón en la UCI se creó un documento llamado “0523_Políticas”, donde se mencionan las mismas, así como las que complementan la realización de estas áreas de proceso, dentro de las cuales se encuentra “10.6 Definir configuración de la metodología a utilizar” donde el proyecto debe definir y documentar la configuración de la metodología a utilizar: fases, actividades, roles y artefactos. Para dar cumplimiento a esta, se definió que la metodología a utilizar sería RUP, pues es la más utilizada en nuestros proyectos y se tiene una vasta experiencia en su uso.

1.3.1. Metodología de desarrollo

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para su creación. Se podrían clasificar en dos grandes grupos: metodologías pesadas y metodologías ligeras o ágiles. Las metodologías pesadas son las más tradicionales. Se centran en la definición detallada de los procesos, tareas y herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodología es más eficaz y necesaria cuando el proyecto que se pretende realizar requiere de mucho tiempo, recursos y gran organización. (18)

El **Proceso Unificado de Desarrollo** (por sus siglas en inglés RUP) es una metodología pesada que define quién debe hacer qué, cuándo y cómo debe hacerlo. El ciclo de vida de RUP posee tres características que lo identifican:

Dirigido por casos de uso: Los casos de uso reflejan lo que el cliente necesita, lo cual se capta al modelar el negocio y se representa a través de los requerimientos. Luego los casos de uso guían el proceso de desarrollo, ya que como resultado de los diferentes flujos de trabajo, los modelos que se obtienen, representan la realización de los casos de uso, es decir, estos se van haciendo por cada caso

Capítulo 1: Fundamentación Teórica

de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema, RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.

Iterativo e Incremental: Cada fase se desarrolla en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos, donde cada uno de ellos es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos al crecimiento del producto. (19)

1.3.2. Tecnologías utilizadas en el proceso de desarrollo

Se presentan a continuación las tecnologías a utilizar para el proceso de desarrollo.

Java Server Faces (JSF) es un framework que define un modelo de componentes de interfaz de usuario y de eventos. Permite manejar el estado de los componentes de usuario, manejar sus eventos, la validación y conversión del lado del servidor y centralizar la navegabilidad de las páginas de la aplicación. JSF es el marco estándar que proporciona Java para construir aplicaciones web, sigue el patrón Modelo Vista Controlador (MVC), proporcionando una manera de validar datos, llamar a reglas de negocio, y devolver los resultados al cliente. (20)

RichFaces es una librería de componentes web enriquecidos, de código abierto y basada en el estándar JSF. Con RichFaces se puede integrar fácilmente las capacidades JavaScript asíncrono y XML (AJAX), utilizando para ello el framework Ajax4jsf, dentro del desarrollo de aplicaciones de negocio de nivel empresarial. RichFaces incluye ciclo de vida, validaciones, conversiones y la gestión de recursos estáticos y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax y un alto grado de personalización del —look-and-feel- que puede ser fácilmente incorporado dentro de las aplicaciones JSF. (21)

Asynchronous JavaScript and XML (AJAX) es un conjunto de tecnologías (HTML-Java Script-CSS-DHTML-PHP/ASP.NET/JSP-XML) que permiten hacer páginas web más interactivas. Su característica

Capítulo 1: Fundamentación Teórica

fundamental es permitir actualizar parte de una página con información que se encuentra en el servidor, sin tener que refrescar completamente la página. De modo similar se puede enviar información al servidor. (22)

Facelets es un framework simplificado de presentación, en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. (23)

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, controlar cualquier evento de usuario, entre otras. (24)

Lenguaje Extensible de Marcado de Hipertexto (por sus siglas en inglés XHTML) es una versión más limpia y estricta de HTML, esto con el objetivo de que como hoy en día lanzan al mercado nuevos dispositivos capaces de interpretar dicho lenguaje, es una forma de crear un estándar de interpretación de código web independientemente del dispositivo. (25)

Hojas de estilo en cascada (por sus siglas en inglés CSS) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, y es imprescindible para crear páginas web complejas. (26)

JBoss Seam versión 2.1.1 GA es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB), funcionando, según versa su significado en español, como una “costura” entre estos componentes. Seam también se integra perfectamente con otros frameworks como: RichFaces, ICEFaces, MyFaces, Hibernate y Spring. (27)

Interfaz de Usuario Seam (por sus siglas en inglés Seam UI) es una serie de controles JSF altamente integrables con JBoss Seam. Están dirigidos a complementar los controles JSF incorporados y los controles de otras bibliotecas externas. La Seam UI brinda a la Interfaz de Usuario JSF (JSF UI) componentes

Capítulo 1: Fundamentación Teórica

Seam para manejar información en tiempo de ejecución, así como un conjunto de componentes para solucionar fácilmente la representación de los problemas del negocio en la aplicación a desarrollar. Entre las ventajas que brinda la misma se encuentran las validaciones, el manejo de las conversaciones y el manejo de los procesos de negocio. (28)

JBoss Server o JBoss AS versión 4.2.2 GA es un Servidor de Aplicaciones Java EE de Software Libre implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. JBoss puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia. Por este motivo, es la plataforma más popular de middleware para desarrolladores, vendedores independientes de software y, también, para grandes empresas. (29)

Hibernate es una herramienta de mapeo objeto relacional. Utiliza para esto archivos declarativos (XML) que permiten establecer estas relaciones. Es una tecnología de software libre distribuida bajo los términos de la licencia GNU LGPL. Como todas las herramientas de su tipo, busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Le permite a la aplicación manipular los datos de la base de datos operando sobre objetos, con todas las características de la programación orientada a objetos, Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL.

Esta herramienta genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias. Logra mantener la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución. Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente conocida como "Criteria". (30)

Enterprise JavaBeans (EJB) versión 3 es un componente utilizado en Java que permite agrupar funcionalidades para formar parte de una aplicación, esto puede ser: un "Java Bean" agrupando información personal, datos sobre un pedimento, requerimientos de órdenes, entre otros. (31)

Java Persistence API (JPA) proporciona un modelo de persistencia basado en la programación orientada a objeto para mapear bases de datos relacionales en Java. El Java Persistence API fue desarrollado por

Capítulo 1: Fundamentación Teórica

el grupo de expertos de EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes; incluso, fuera de la plataforma Java EE, por ejemplo, en aplicaciones Java SE. (32)

JasperReports es una librería de clases de Java de código abierto diseñada para facilitar el agregar capacidades de reporte a las aplicaciones Java. No es una herramienta por sí sola, por lo que no se puede instalar. Para utilizar JasperReports es necesario añadirlo a las aplicaciones Java por medio de la inclusión de su librería. Esta librería permite la creación de informes Java libre que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML. Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico. Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible. JasperReports se usa comúnmente con iReport, una herramienta de código abierto para la edición de informes. (33)

PostgreSQL versión 8.4 es un Sistema Gestor de Base de Datos (por sus siglas en inglés SGBD) relacional de código abierto, muy poderoso, con una arquitectura probada. Puede ser ejecutado sobre la mayoría de los sistemas operativos que existen hoy en día. Posee protección de transacciones u operaciones de Atomicidad, Consistencia, Aislamiento, Durabilidad (ACID, por sus siglas en inglés). Es un gestor de base de datos empresarial, que posee características sofisticadas como Control de Concurrencia Multi - Versión (por sus siglas en inglés MVCC), replicación asíncrona, transacciones anidadas, realización de respaldo de datos en línea, optimizador o planificador de consultas, soporta internacionalización. Es altamente escalable en cuanto a la cantidad de información que puede manejar y al número de usuarios concurrentes que puede alojar. (34)

1.3.2.1. Lenguaje de Programación

El lenguaje de programación utilizado es **Java**. Fue desarrollado por la compañía Sun Microsystems, y utiliza el paradigma de la Programación Orientada a Objetos (POO). Es un lenguaje robusto, pues no permite el manejo directo del hardware ni de la memoria. Dentro de sus principales ventajas se encuentra la de ser multiplataforma. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Con Java se pueden programar aplicaciones web dinámicas, con acceso a bases de datos, utilizando XML, con cualquier tipo

Capítulo 1: Fundamentación Teórica

de conexión de red entre cualquier sistema. Este lenguaje es utilizado de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación. (35)

Permite organizar el código en entidades como las clases compuestas de datos y funciones, y a través de la característica de la herencia se pueden organizar las clases en jerarquías. La programación en Java admite, tanto el desarrollo de aplicaciones bajo el esquema de Cliente-Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más ordenadores, ejecutando tareas simultáneamente y de esta forma consigue distribuir el trabajo a cumplir.

Para el modelado de los artefactos se utilizará el Lenguaje Unificado de Modelado (por sus siglas en inglés UML), definido a continuación.

1.3.2.2. Lenguaje Unificado de Modelado

Lenguaje Unificado de Modelado versión 2.0 (por sus siglas en inglés UML) es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. Se centra en la representación gráfica de un sistema. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso usar.

Los elementos de UML se clasifican en: Estructurales (clases, interfaces, colaboraciones, casos de uso, clases activas, componentes y nodos), de comportamiento (interacciones y máquinas de estado), de agrupación (paquetes), de anotación (notas). A su vez, hay cuatro tipos de relaciones: de dependencia, de asociación, de agrupación y de realización.

Para construir un plano de software que tenga sentido, se combinan los elementos estructurales con sus respectivas relaciones, según sea el caso, obteniendo como resultado uno de los nueve diagramas que existen en UML: de clases, de objetos, de casos de uso, de secuencia, de colaboración, de estados, de actividades, de componentes y de despliegue.

Las funciones de UML son:

- ✓ Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ Especificar: Permite especificar cuáles son las características de un sistema antes de su

Capítulo 1: Fundamentación Teórica

construcción.

- ✓ Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- ✓ Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- ✓ Relaciones: Relacionan los elementos entre sí.
- ✓ Diagramas: Son colecciones de elementos con sus relaciones. (36)

1.3.3. Tecnologías Horizontales

Existen un conjunto de tecnologías que sirven de soporte a las antes mencionadas. Dichas tecnologías se describen a continuación.

Java Platform Enterprise Edition (JavaEE) versión 5 es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java, preferiblemente del lado del servidor. Se caracteriza por tener una arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares, ejecutándose sobre un servidor de aplicaciones. Esta plataforma incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, servicios web, ML, etc. Y define cómo coordinarlos. (37)

Java Runtime Environment (JRE) versión 1.6.0_24 es la máquina virtual de Java más básica. Sin ella, ningún programa Java lograría arrancar, y lo mismo se aplica a los applets de las páginas web. Es un componente tan imprescindible como .NET o las librerías DirectX. Su instalador se encarga de integrar Java con el sistema y los principales navegadores (como Internet Explorer o Firefox). (38)

1.3.4. Herramientas de desarrollo.

Se presentan a continuación las herramientas a utilizar para el proceso de desarrollo de la solución propuesta.

Eclipse es un entorno de desarrollo integrado de código abierto, portable y multiplataforma. Este fue

Capítulo 1: Fundamentación Teórica

diseñado originalmente por la empresa IBM y actualmente, es desarrollado por la Fundación Eclipse, una organización independiente, sin ánimo de lucro que fomenta una comunidad de código abierto. (39) Se basa en el uso de módulos (plugins), lo cual hace posible el trabajo en múltiples lenguajes de programación como son Java, C++, PHP, Perl y que se le puedan añadir otras funcionalidades. Mediante el SDE Enterprise edition permite la integración con la herramienta Visual Paradigm, propiciando un mejor entendimiento de todas las partes involucradas en el desarrollo del sistema. Cuenta además, con un sistema de control de versiones, el cual usando una combinación de vistas y editores que muestran los diversos aspectos de los recursos del proyecto organizados por el rol o la tarea del desarrollador, hace más fácil y eficiente el trabajo en equipo.

pgAdmin III es la plataforma de desarrollo de PostgreSQL y la más avanzada base de datos de código abierto en el mundo. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir simples consultas SQL hasta la elaboración de complejas bases de datos. La interfaz gráfica soporta todas las características de PostgreSQL y hace fácil la administración. La conexión con el servidor puede hacerse utilizando el protocolo TCP-IP y puede ser encriptado SSL para la seguridad. No se requieren drivers adicionales para comunicarse con el servidor de base de datos. Es desarrollado por una comunidad de expertos de PostgreSQL de todo el mundo y está disponible en más de una docena de idiomas. (40)

Visual Paradigm versión 6.4 es una herramienta que sirve para realizar modelado UML siguiendo el estándar UML 2.1. Esta herramienta tiene unas características gráficas muy cómodas que facilitan la realización de los diagramas de modelado UML. (41)

Características:

- ✓ Centrado en Casos de Uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Permanece sincronizado en todo el ciclo de desarrollo el modelo y el código.
- ✓ Admite tanto la ingeniería directa como inversa, soportada en varios lenguajes de programación.
- ✓ Uso de un lenguaje estándar común para todo el equipo de desarrollo facilitando la comunicación.
- ✓ Permite la disponibilidad de múltiples versiones para cada necesidad, capaz de integrarse con los principales Ambientes Integrados del Desarrollo (IDEs).

Capítulo 1: Fundamentación Teórica

- ✓ Disponible en múltiples plataformas y soporta el análisis y diseño orientado a objetos.
- ✓ Genera Bases de Datos basado en diagramas.

JBoss Tools es una herramienta compuesta por un conjunto de plugins con características enfocadas al desarrollo y soporte de aplicaciones Web. Permite vincular tecnologías como J2EE, JBoss, Seam, Hibernate, JSF, JSP, servidores de aplicaciones, HTML, CSS, etc., brindando un soporte muy completo para el desarrollador. (42)

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

- ✓ Escrito en el lenguaje Java.
- ✓ Maneja el 98 por ciento de las etiquetas de JasperReports.
- ✓ Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreportes.
- ✓ Soporta internacionalización nativamente.
- ✓ Recopilador y exportador integrados.
- ✓ Soporta JDBC y JavaBeans como orígenes de datos (estos deben implementar la interface JRDataSource).
- ✓ Incluye asistentes para crear automáticamente informes, generar los subreportes y las plantillas.
- ✓ Facilita la instalación. (43)

El análisis realizado permite arribar a las siguientes conclusiones parciales:

- ✓ La identificación de los conceptos y definiciones fundamentales asociados al problema a resolver, posibilitó una mejor comprensión para el desarrollo de la propuesta tecnológica.
- ✓ El estudio realizado de los numerosos sistemas de gestión de trasplante arrojó como resultado un conjunto de funcionalidades que servirán de soporte para el desarrollo de la solución propuesta.

Capítulo 1: Fundamentación Teórica

- ✓ La caracterización de la metodología, las técnicas, tecnologías y herramientas utilizadas facilitó el trabajo con las mismas y permitió la obtención de los artefactos correspondientes al desarrollo del sistema.

Capítulo 2: Características del Sistema

Capítulo 2: Características del Sistema

En este capítulo se realiza una descripción detallada de los procesos para un mejor entendimiento del mismo. Además, se identifican los actores y casos de uso del sistema, así como los requisitos funcionales y no funcionales que soportará.

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

Un proceso es un conjunto de actividades que se realizan para obtener un determinado resultado. En el área de trasplante de riñón de los centros hospitalarios se tienen tres procesos principales: Gestionar receptor, Gestionar donante y Gestionar estadísticas de trasplante de riñón.

2.1.1. Gestionar receptor

Mensualmente el Centro Coordinador Nacional de Trasplante recibe de las unidades de hemodiálisis, diálisis peritoneal y las consultas de IRC los datos de los pacientes con necesidad de trasplante que estén aptos para el mismo, esto se realiza por correo electrónico o vía telefónica. Dichos receptores son ubicados por el Coordinador nacional de trasplante en la lista nacional de trasplante, teniendo en cuenta varios parámetros: por ciento de hipersensibilidad, si es niño (menor de 16 años) y la prioridad establecida por el especialista.

2.1.2. Gestionar donante

Una vez recibido un donante cadáver el Especialista en trasplante recoge en la Hoja del donante la información referente a este y lo incluye en el listado de donantes. Luego se realizan pruebas para determinar la compatibilidad de este con los receptores, teniendo en cuenta: Grupo y Factor, edad, peso, sexo, tiempo en diálisis y el tipaje HLA, obteniéndose el listado de pacientes compatibles. El suero extraído de la sangre de estos pacientes se ubica en las posiciones de la placa Terasaki y se realiza la prueba Cross-Match, descartándose los que posean resultado positivo y seleccionando como receptor idóneo el que se encuentre en la primera posición de la lista.

2.1.3. Gestionar reporte estadístico

Anualmente los especialistas en trasplante tienen la posibilidad de generar reportes estadísticos. Estos tienen como objetivo determinar la cantidad de receptores por hospital de procedencia y la cantidad de

Capítulo 2: Características del Sistema

donantes por centro, para apoyar la toma de decisiones.

2.2. Información que se maneja

Los documentos que se exponen a continuación representan la información que se gestiona en el negocio actual.

Una vez recibidos los datos de un receptor, el Coordinador nacional de trasplante crea la **Hoja del receptor**. En ella se recogen los datos referentes al receptor como son: nombre(s) y apellidos, carnet de identidad, sexo, edad, grupo sanguíneo, método de diálisis, tipaje HLA, entre otros. Posteriormente se incluye a este receptor en el **listado nacional de pacientes a trasplantar**, teniendo en cuenta: por ciento de hipersensibilidad, si es niño y la prioridad establecida por el especialista.

La **Hoja del donante** es creada por el Especialista en trasplante cuando existe un donante cadáver apto para el trasplante. En ella se recogen los datos referentes al mismo tales como: nombre(s) y apellidos, carnet de identidad, sexo, edad, grupo sanguíneo, causa de muerte, hora de llamada, centro que dona, entre otros. Éste es incluido en el **listado de donantes**, el cual es utilizado para realizar las pruebas de compatibilidad.

El **listado de pacientes compatibles** se obtiene al realizar la prueba de compatibilidad con el donante y los receptores, donde se tienen: Grupo y Factor, edad, sexo, peso, tiempo en diálisis y tipaje HLA. Luego este listado es actualizado con los pacientes que posean resultado negativo en la prueba de Cross_Match.

2.3. Objeto de automatización

Para darle solución a la problemática, el departamento SES del centro CESIM de la UCI, propone desarrollar una herramienta informática que permita gestionar la información de los trasplantes de riñón para el Sistema de Trasplante. Para darle cumplimiento se desea automatizar los siguientes procesos: Gestionar receptor, pues en este se crea la Hoja del receptor y se incluye el mismo en el listado de pacientes a trasplantar; Gestionar donante, porque es donde se establece la prioridad en el listado de pacientes compatibles y se genera el listado de pacientes a trasplantar y el proceso Gestionar estadísticas de trasplante de riñón, que permite que anualmente los especialistas en trasplante generen reportes que reflejen la cantidad de receptores por hospital de procedencia y la cantidad de donantes por centro.

Capítulo 2: Características del Sistema

2.4. Modelación de Negocio

La Modelación de Negocio es la fase donde se realiza el estudio de la organización, se examina su estructura, el flujo de trabajo y los procesos principales de la misma. Además, se observan los roles y la relación que existe entre ellos. Para la descripción y modelado de negocio se utiliza la notación BPMN.

BPMN es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. La notación ha sido diseñada, específicamente, para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. Las cuatro categorías básicas de elementos son: objetos de flujo, objetos conectores, artefactos y swimlanes. (44)

Los objetos de flujo se conectan entre ellos en un diagrama para crear el esqueleto básico de la estructura de un proceso de negocio. Hay tres objetos conectores que hacen esta función: Flujo de secuencia, flujo de mensaje y asociación.

Swimlanes (calles): Permite organizar las actividades en categorías separadas visualmente para ilustrar diferentes capacidades funcionales o responsabilidades. BPMN soporta los swimlanes con dos constructores principales. Los dos tipos de objetos swimlanes son: pool y lane.

2.4.1. Actores y trabajadores del negocio

Un **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.

En la Tabla 1 se describe el actor involucrado en los procesos de negocio:

Tabla 1. Actor involucrado en el proceso de negocio.

Actor	Funciones
Paciente	Persona de cualquier edad con IRC.

Un **Trabajador del negocio** representa un rol que juega una persona (o grupo de personas), una máquina o un sistema automatizado; actuando en el negocio. Son los que realizan las actividades, interactuando con otros trabajadores del negocio y manipulando entidades. (45)

Capítulo 2: Características del Sistema

En la Tabla 2 se describen los trabajadores involucrados y representados en los procesos de negocio para una mejor comprensión de los mismos:

Tabla 2. Trabajadores involucrados en el proceso de negocio.

Trabajadores	Funciones
Coordinador nacional de trasplante	Es el encargado de crear la Hoja del Receptor e incluirlo en el listado de paciente a trasplantar.
Especialista en trasplante	Es el encargado de crear la Hoja de Donante e incluirlo en el listado de donantes, establecer la prioridad en el listado de pacientes compatibles, seleccionar el receptor idóneo y gestionar reportes estadísticos.

2.4.2. Diagramas de procesos del negocio

Los modelos de procesos del negocio son una representación abstracta de los procesos de una organización. Muestran cómo y quién efectúa las actividades que generan valor para la organización, visualizan todas las funciones que se realizan en la organización y permiten una visión más abarcadora de cómo se hace el trabajo.

En la Figura 3 se representa el diagrama de procesos del negocio “Gestionar receptor”.

Capítulo 2: Características del Sistema

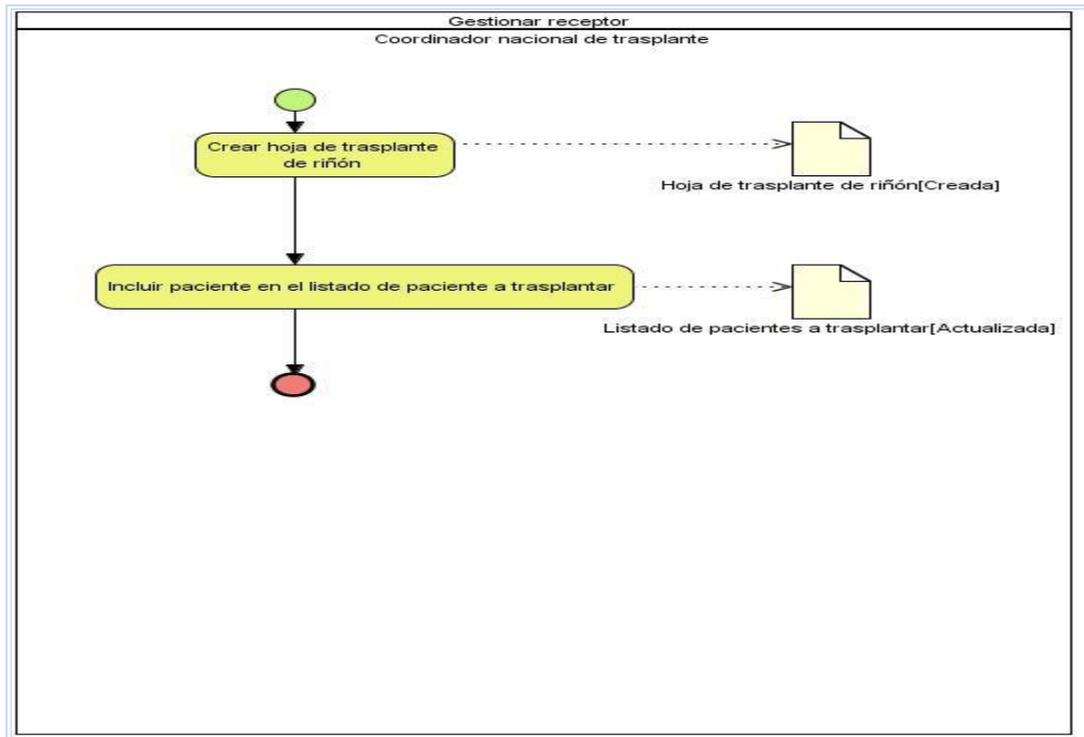


Figura 3. Gestionar receptor.

A continuación se muestra una descripción de dicho proceso.

Tabla 3. Gestionar receptor.

Nombre:	Gestionar receptor.
Objetivos:	Gestionar receptor.
Evento(s) que lo generan:	Cuando el paciente llega a la consulta.
Precondiciones:	El paciente debe de haber sido atendido por el Coordinador Nacional de trasplante.
Poscondiciones:	Se creó la Hoja de trasplante de riñón. Se actualizó el Listado de pacientes a trasplantar.
Reglas de Negocio:	No aplica.
Responsables:	Coordinador nacional de trasplante.

Capítulo 2: Características del Sistema

Clientes internos:	No aplica.	
Clientes externos:	Paciente.	
Entradas:	Hoja de trasplante de riñón.	
Salidas:	Hoja de trasplante de riñón. Listado de pacientes a trasplantar.	
Actividades:	Crear Hoja de trasplante de riñón. Incluir paciente en el listado de paciente a trasplantar.	
Actividad 1		
Crear Hoja de trasplante de riñón: La Hoja de trasplante de riñón se crea cuando el paciente está apto para ser trasplantado.		
Responsables	Entradas	Salidas
Coordinador nacional de trasplante.	No aplica.	Hoja de trasplante de riñón.
Actividad 2		
Incluir paciente en el listado de paciente a trasplantar: Actividad que realiza el Coordinador nacional de trasplante, con el objetivo de adicionar al paciente en el listado de pacientes a trasplantar.		
Responsables	Entradas	Salidas
Coordinador nacional de trasplante.	Hoja de trasplante de riñón.	Listado de pacientes a trasplantar.

En la Figura 4 se representa el diagrama de procesos del negocio “Gestionar donante”.

Capítulo 2: Características del Sistema

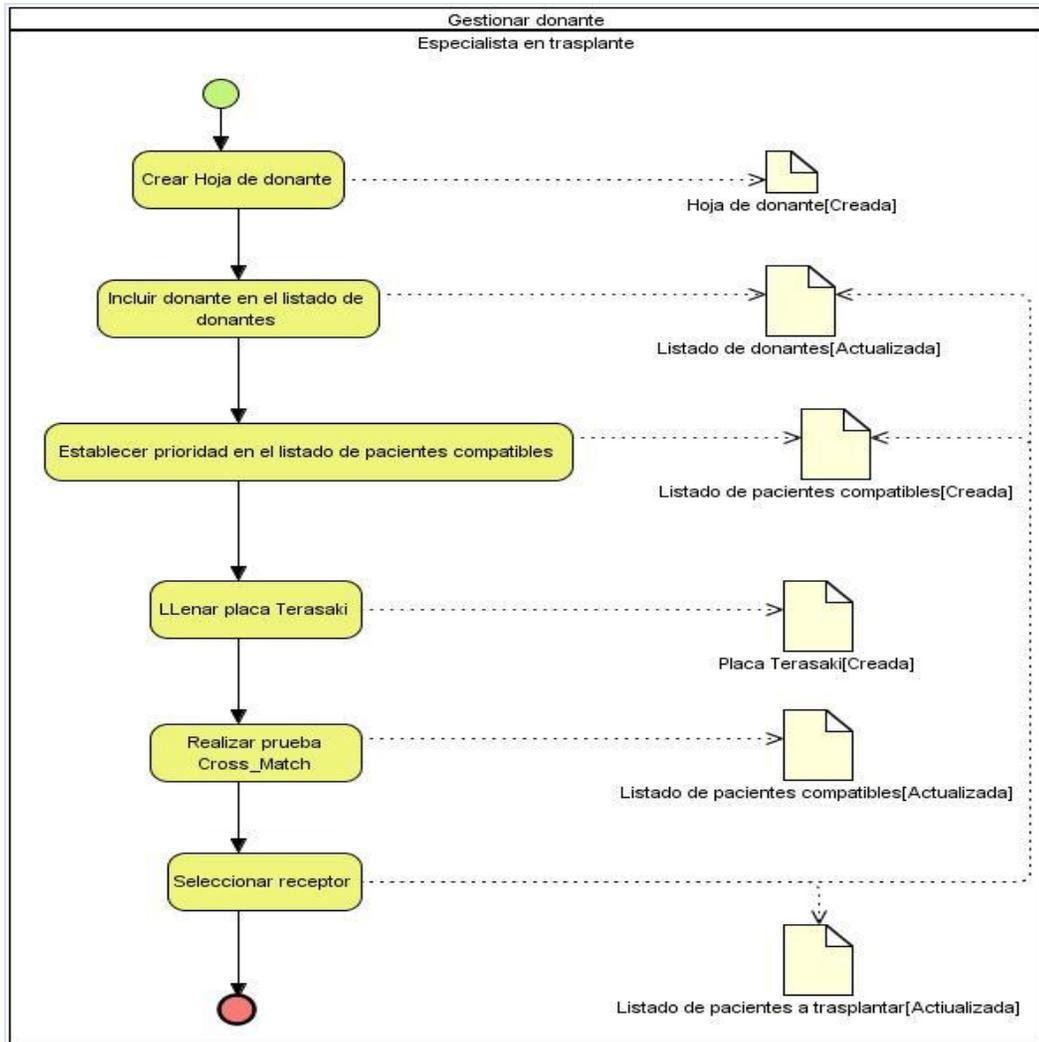


Figura 4. Gestionar donante.

A continuación se muestra una descripción de dicho proceso.

Tabla 4. Gestionar donante.

Nombre:	Gestionar donante.
Objetivos:	Seleccionar un receptor que sea compatible con el donante.
Evento(s) que lo generan:	Cuando llega un donante.

Capítulo 2: Características del Sistema

Precondiciones:	El Especialista en trasplante debe haberle realizado pruebas donante.	
Poscondiciones:	Se creó la Hoja de donante. Se actualizó el Listado de donantes. Se creó el Listado de pacientes compatibles. Se actualizó el Listado de pacientes compatibles. Se actualizó el Listado de pacientes a trasplantar.	
Reglas de Negocio:	No aplica.	
Responsables:	Especialista en trasplante.	
Clientes internos:	No aplica.	
Clientes externos:	Paciente.	
Entradas:	Hoja de donante.	
Salidas:	Hoja del Donante. Listado de donantes. Listado de pacientes compatibles. Placa Terasaki. Listado de pacientes a trasplantar.	
Actividades:	Crear Hoja de donante. Incluir donante en el Listado de donantes. Establecer prioridad en el Listado de pacientes compatibles. Llenar placa Terasaki. Realizar prueba Cross_Match. Seleccionar receptor.	
Actividad 1		
Crear Hoja de donante: La hoja del donante es creada cuando aparece un donante cadáver apto para el trasplante.		
Responsables	Entradas	Salidas
Especialista en trasplante.	No aplica.	Hoja del donante.
Actividad 2		

Capítulo 2: Características del Sistema

Incluir donante en el Listado de donantes: Una vez que el especialista en trasplante crea la Hoja de donante, incluye al mismo en el Listado de donantes.

Responsables	Entradas	Salidas
Especialista en trasplante.	Hoja del donante.	Listado de donantes.

Actividad 3

Establecer prioridad en el listado de pacientes compatibles: Una vez que el donante cadáver es incluido en el listado de donantes se procede a buscar la compatibilidad de éste con los receptores, teniendo en cuenta el Grupo y Factor, edad, peso, sexo, tiempo en diálisis y el tipaje de antígenos leucocitarios humanos (HLA); a cada uno de estos aspectos se le atribuye una puntuación, creando el listado de pacientes compatibles organizados de mayor a menor según dicha puntuación.

Responsables	Entradas	Salidas
Especialista en trasplante.	Hoja de donante.	Listado de pacientes compatibles.

Actividad 4

Llenar placa Terasaki: El especialista en trasplante llena la placa Terasaki.

Responsables	Entradas	Salidas
Especialista en trasplante.	Listado de pacientes a trasplantar.	Placa Terasaki.

Actividad 5

Realizar prueba Cross_Match: El especialista en trasplante realiza esta prueba para determinar los pacientes compatibles que no harán rechazo al riñón del donante.

Responsables	Entradas	Salidas
Especialista en trasplante.	Placa Terasaki.	Listado de pacientes compatibles.

Actividad 6

Seleccionar receptor: Luego de haber determinado los receptores que poseen resultado negativo se selecciona el que se desea trasplantar.

Responsables	Entradas	Salidas
Especialista en trasplante.	Listado de pacientes compatibles. Listado de pacientes a trasplantar. Listado de donantes.	Paciente seleccionado.

En la Figura 5 se representa el diagrama de procesos del negocio “Gestionar estadísticas de trasplante de

Capítulo 2: Características del Sistema

riñón”.

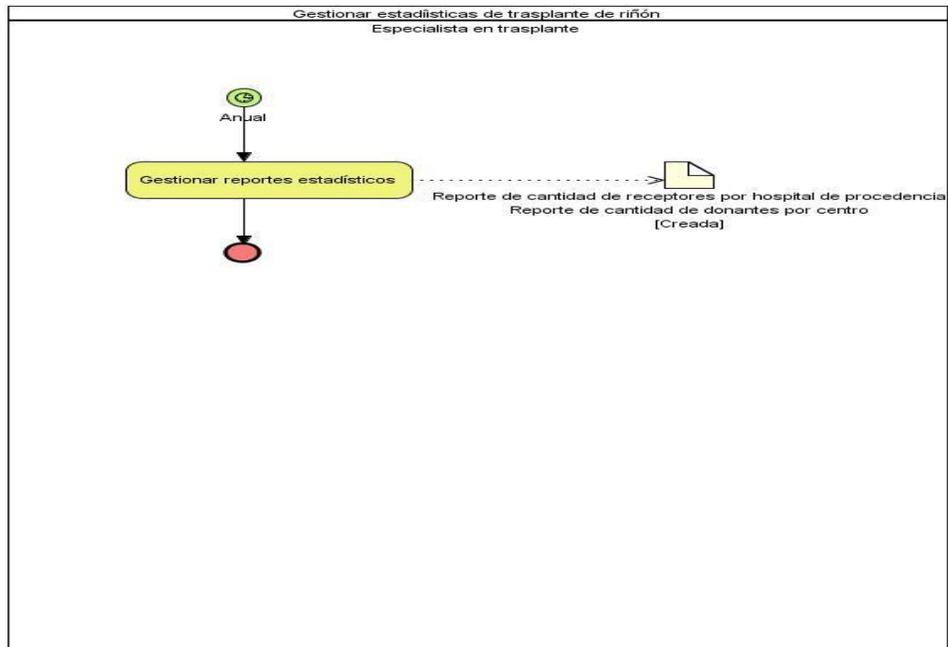


Figura 5. Gestionar estadísticas de trasplante de riñón.

A continuación se muestra una descripción de dicho proceso.

Tabla 5. Gestionar estadísticas de trasplante de riñón.

Nombre:	Gestionar estadísticas de trasplante de riñón.
Objetivos:	Realizar reportes estadísticos para determinar la cantidad de receptores por hospital de procedencia y la cantidad de donantes por centro.
Evento(s) que lo generan:	Cuando un especialista desea generar un reporte anualmente.
Precondiciones:	No aplica.
Poscondiciones:	Se creó un Reporte de la cantidad de donantes por centro. Se creó un Reporte de la cantidad de receptores por hospital

Capítulo 2: Características del Sistema

	de procedencia.	
Reglas de Negocio:	No aplica.	
Responsables:	Especialista en trasplante.	
Clientes internos:	No aplica.	
Clientes externos:	No aplica.	
Entradas:	Hoja de donante. Hoja de trasplante de riñón.	
Salidas:	Reporte de cantidad de receptores por hospital de procedencia. Reporte de cantidad de donantes por centro.	
Actividades:	Gestionar reportes estadísticos.	
Actividad 1		
Gestionar reportes estadísticos: A final de cada año el especialista debe generar reportes, para saber la cantidad de receptores por hospital de procedencia y la cantidad de donantes por centro.		
Responsables	Entradas	Salidas
Especialista en trasplante.	Hoja de donante. Hoja del receptor.	Reporte cantidad de receptores por hospital de procedencia. Reporte de cantidad de donantes por centro.

2.5. Especificación de los requisitos del software

La especificación de los requisitos de software (ERS) provee una imagen clara del producto software. La empresa desarrolladora y la empresa cliente deben comunicarse muy estrechamente para establecer los requisitos de la aplicación. Una buena ERS ayudará a la empresa cliente a describir qué es lo que quiere, y también servirá a los desarrolladores para comprender qué es, exactamente, lo que le están pidiendo. Por tanto, es muy importante que en ella se describa lo más detalladamente posible lo que se espera que haga dicho software. Los requisitos de sistemas de software se clasifican en funcionales y no funcionales.

Capítulo 2: Características del Sistema

2.5.1. Requisitos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. Además, describen lo que el sistema debe hacer y en algunos casos lo que no debe hacer.

A partir de los procesos de negocio estudiados se pueden definir los siguientes requisitos funcionales:

Tabla 6. Requisitos funcionales.

No	Requisitos Funcionales
RF1.	Listar pacientes a trasplantar.
RF2.	Seleccionar donante.
RF3.	Ver datos de Hoja de donante.
RF4.	Seleccionar paciente compatible.
RF5.	Buscar donantes.
RF6.	Modificar Hoja de donante.
RF7.	Eliminar Hoja de donante.
RF8.	Crear Hoja de donante.
RF9.	Ver detalles de Hoja de donante.
RF10.	Buscar paciente a trasplantar.
RF11.	Ver datos de Hoja de trasplante de riñón.
RF12.	Modificar Hoja de trasplante de riñón.
RF13.	Eliminar Hoja de trasplante de riñón.
RF14.	Seleccionar paciente a trasplantar.
RF15.	Visualizar Placa Terasaki.
RF16.	Insertar resultado Cross-Match.
RF17.	Ver detalles de resultado Cross-Match.
RF18.	Modificar resultado Cross-Match.
RF19.	Eliminar resultado Cross-Match.
RF20.	Buscar resultado Cross-Match.

Capítulo 2: Características del Sistema

RF21.	Crear Hoja de trasplante de riñón.
RF22.	Ver detalles de Hoja de trasplante de riñón.
RF23.	Listar pacientes trasplantados
RF24.	Generar reporte de cantidad de receptores por hospital de procedencia.
RF25.	Generar reporte de cantidad de donantes por centro.

2.5.2. Requisitos no funcionales

Una vez determinado lo que el sistema debe hacer se especifica cómo debe comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Los requisitos no funcionales responden a las condiciones que debe cumplir el sistema para satisfacer el contrato o especificación. Están regidos por las necesidades del usuario para poder resolver un problema o conseguir un beneficio determinado. Los requisitos no funcionales son de gran utilidad para la aceptación del software, debido a que representan las ventajas más visibles al usuario y repercuten en el óptimo funcionamiento y mantenimiento del sistema.

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

Usabilidad

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

- ✓ Usuarios de pocos conocimientos informáticos: 30 días.
- ✓ Usuarios de avanzados conocimientos informáticos: 20 días.

La organización de la información facilita al usuario un buen entendimiento del sistema. Un usuario con un mínimo conocimiento del sistema será capaz de alcanzar sus objetivos con el máximo resultado. El sistema será capaz de señalarle al usuario los errores cometidos de una forma clara y entendible.

Fiabilidad

Solo el propio usuario o el administrador del sistema podrán cambiar la contraseña. Los usuarios tendrán

Capítulo 2: Características del Sistema

permiso al sistema en dependencia de la función que realicen. Además, permitirá realizar salvos o respaldos de la base de datos. El paciente nunca se le dará baja del mismo, con el objetivo de dar posibles seguimientos futuros.

Seguridad

La seguridad del sistema tiene como objetivo garantizar la continuidad del mismo en caso de que se produzcan incidencias, fallos, pérdidas accidentales o desastres que afecten los datos almacenados. Por esta razón se garantizará la confidencialidad de la información, protegiéndola contra el acceso de usuarios no autorizados, utilizando mecanismos de validación que puedan garantizar el cumplimiento de: usuario, contraseña y nivel de acceso, de manera que, cada uno pueda tener disponible solamente las opciones relacionadas con su actividad o rol. De esta manera la información será modificada solo por las personas autorizadas, posibilitando la integridad de la misma.

Eficiencia

Es la capacidad para hacer un buen uso de los recursos que se manipulan describiendo el rendimiento de la aplicación y los recursos que ésta utiliza bajo condiciones específicas. El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.

Hardware

Las estaciones de trabajo deben tener 1GB de Memoria de Acceso Aleatorio (por sus siglas en inglés RAM) y un microprocesador Intel® Core-2/Duo o Intel® Dual-Core. Para los servidores la solución estará conformada, fundamentalmente, por 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: Procesador Intel® Xeon® 5140 Dual – Core 8GB de memoria. Servidores de Aplicaciones: Procesador Intel® Xeon® 5140 Dual – Core 16GB de memoria. En ambos casos, se debe garantizar el almacenamiento inicial para la instalación de aplicaciones de 1x72GB de disco y luego garantizar el almacenamiento a medida que aumente el volumen de los datos.

Software

Se especifica las condiciones o capacidades que el sistema debe cumplir. El software que se usará tanto

Capítulo 2: Características del Sistema

en la parte del servidor como en la del cliente. El sistema debe correr en sistemas operativos Windows y Linux, utiliza la plataforma JAVA Runtime Environment (JRE) versión 1.6.0_24, JBoss AS versión 4.2.2_GA y PostgreSQL versión 8.4. En la solución se incluyen estaciones de trabajo que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web: Firefox 3.6 y Google Chrome 14 o versiones superiores de estos.

Diseño

Se especifica o restringe la construcción de un sistema. Son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

Se utilizará el patrón de diseño MVC. 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 bajo la ejecución de varios usuarios. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

Interfaces de usuario

Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español.

2.6. Modelo de casos de uso del sistema

El Modelo de Casos de Uso del Sistema, es un artefacto de Ingeniería de Software que describe un acuerdo entre clientes y desarrolladores sobre los requisitos que debe cumplir el sistema. Este modelo está formado por actores, casos de usos y las relaciones que se establecen entre estos. Por tanto, representa de manera gráfica, los procesos que tendrán lugar en un sistema, así como la interacción con los actores.

En la Tabla 7 se describen los actores del sistema para una mejor comprensión de los mismos.

Tabla 7. Actores del sistema.

Capítulo 2: Características del Sistema

Actor	Objetivo
Coordinador nacional de trasplante	Es el encargado de crear la Hoja del Receptor e incluirlo en el listado de paciente a trasplantar.
Especialista en trasplante	Es el encargado de crear la Hoja de Donante e incluirlo en el listado de donantes, establecer la prioridad en el listado de pacientes compatibles, seleccionar el receptor idóneo y gestionar reportes estadísticos.

2.6.1. Diagrama de casos de uso

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

A continuación se muestra el diagrama de caso de uso “Gestionar trasplante”.

Capítulo 2: Características del Sistema

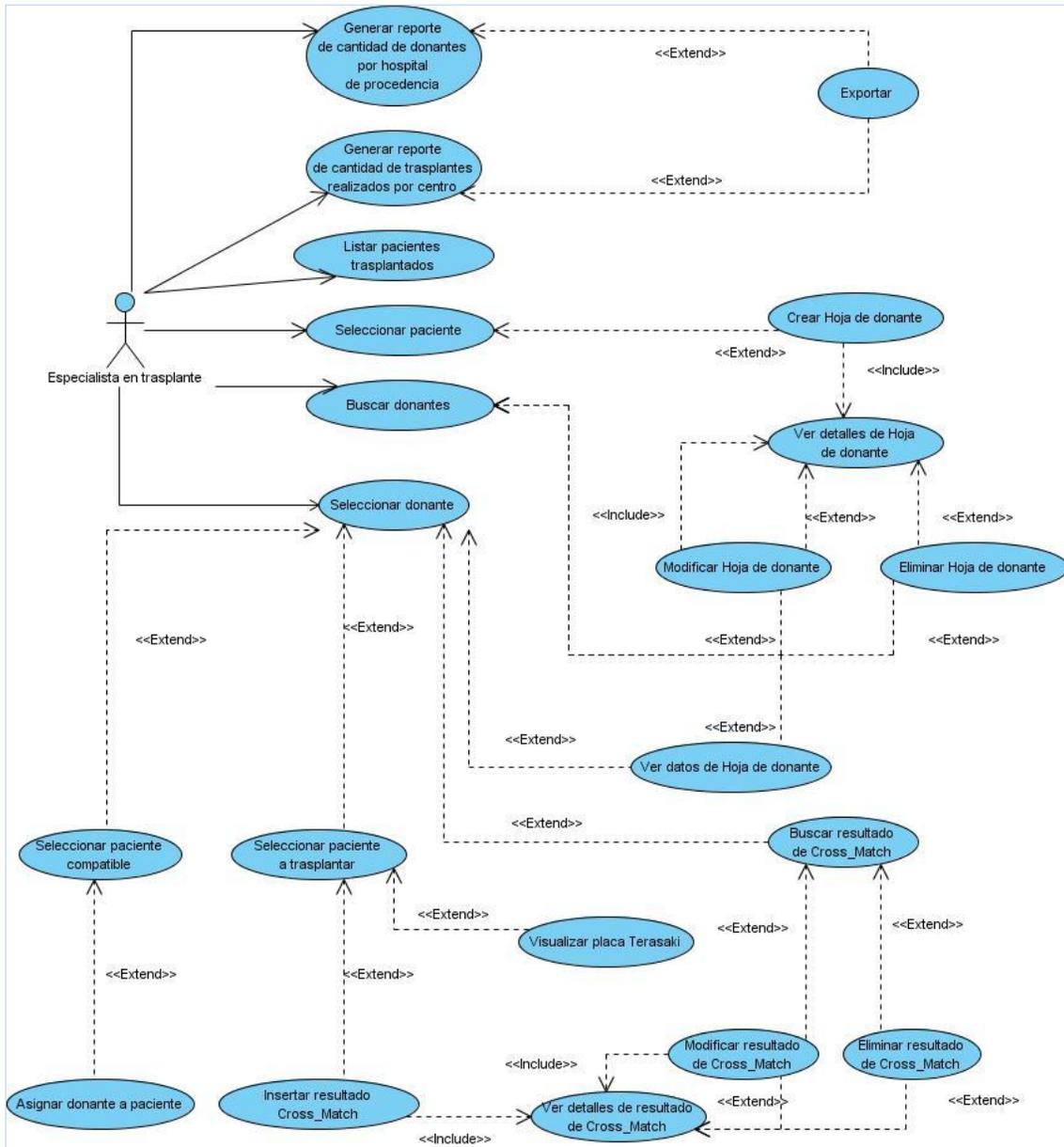


Figura 6. Gestionar trasplante.

Capítulo 2: Características del Sistema

2.6.2. Descripción Textual de los Casos de Uso

A continuación se describen algunos casos de uso presentes en el diagrama de caso de uso “Gestionar trasplante”. Ver Anexos para una mejor comprensión de otros casos de uso del sistema.

Tabla 8. Crear Hoja de donante.

Objetivo	Crear Hoja de donante.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor accede a la opción Crear Hoja de donante, el sistema brinda la posibilidad de introducir los datos para crear la Hoja de donante, el actor introduce los datos de la Hoja de donante, el sistema crea la Hoja de donante, el caso de uso termina.	
Complejidad	Alta.	
Prioridad	Alta.	
Precondiciones	Debe de haber sido seleccionado un paciente.	
Postcondiciones	Se creó una Hoja de donante.	
Flujo de eventos		
Flujo básico Crear horario		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Crear Hoja de donante.	
2.		Brinda la posibilidad de introducir los datos de la Hoja de donante : Y permite: <ul style="list-style-type: none"> • Aceptar Crear Hoja de donante. • Cancelar operación. Ver Sección 3: “Cancelar operación.”
3.	Introduce todos los datos de la Hoja de	

Capítulo 2: Características del Sistema

	donante.	
4.	Selecciona la opción de aceptar Crear Hoja de donante.	
5.		Valida los datos. Si hay datos incompletos, ver Sección 1 : “Existen datos incompletos.”. Si hay datos incorrectos, ver Sección 2 : “Existen datos incorrectos.”
6.		Crea Hoja de donante.
7.		Muestra los detalles de la Hoja de donante. Ver caso de uso: Ver detalles de Hoja de donante .
8.		El caso de uso termina
Flujos alternos		
Sección 1: “Existen datos incompletos”.		
	Actor	Sistema
1.		Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 1: “Existen datos incorrectos”.		
	Actor	Sistema
1		Muestra un indicador sobre los campos incorrectos.

Capítulo 2: Características del Sistema

2		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 3: “Cancelar la operación”		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción de Cancelar operación.	
2		Regresa a la vista anterior.
3.		El caso de uso termina.
Relaciones	CU Incluidos	Ver detalles de Hoja de donante; ver caso de uso: Ver detalles de Hoja de donante.
	CU Extendidos	No existe.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 9. Seleccionar paciente compatible.

Objetivo	Seleccionar paciente compatible.
Actores	Especialista en trasplante.
Resumen	El caso de uso inicia cuando el actor accede a la opción Seleccionar paciente compatible, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el paciente compatible deseado, el actor introduce los datos que considera como criterios para buscar paciente compatible, el sistema busca y muestra los paciente compatible que cumplen con los criterios de búsqueda, el actor selecciona el paciente compatible deseado, el sistema carga en la vista anterior cada una de los pacientes compatibles seleccionados, el caso de uso termina.

Capítulo 2: Características del Sistema

Complejidad	Media.	
Prioridad	Alta.	
Precondiciones	Debe haberse seleccionado un donante.	
Postcondiciones	Se seleccionó un paciente compatible dado criterios.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Seleccionar paciente compatible del Seleccionar donante.	
2.		<p>Muestra los datos del donante:</p> <ul style="list-style-type: none"> • Nombre (s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Muestra un listado de los pacientes a trasplantar.</p> <p>Ordenado teniendo en cuenta criterios establecidos, mostrando la cantidad de elementos configurada para mostrar por página, permitiendo navegar por el resultado.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Seleccionar un paciente compatible, para luego Asignar donante a paciente.

Capítulo 2: Características del Sistema

		<ul style="list-style-type: none"> • Cancelar operación. Ver Sección 1: “Cancelar operación.”
3.	Selecciona el paciente deseado.	
4.		Permite <ul style="list-style-type: none"> • Asignar donante a paciente. Ver Sección 2: “Asignar donante a paciente”.
5.		El caso de uso termina.

Sección 1: “Asignar donante a paciente”.

Flujo básico

	Actor	Sistema
1.	Selecciona el paciente deseado.	
2.		Permite: Asignar donante a paciente. Ver caso de uso : “Asignar donante a paciente”.
3.		El caso de uso termina.

Sección 2: “Cancelar operación”

Flujo básico

	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina

Capítulo 2: Características del Sistema

Relaciones	CU Incluidos	No existe.
	CU Extendidos	Asignar donante a paciente. Ver caso de uso: "Asignar donante a paciente".
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 10. Insertar resultado de Cross_Match.

Objetivo	Insertar resultado Cross_Match.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor accede a la opción Insertar resultado Cross_Match, el sistema brinda la posibilidad de introducir los datos para insertar resultado Cross_Match, el actor introduce los datos del resultado Cross_Match, el sistema inserta resultado Cross_Match, el caso de uso termina.	
Complejidad	Alta.	
Prioridad	Media.	
Precondiciones	Debe estar seleccionado el paciente a trasplantar.	
Postcondiciones	Se Insertó el resultado Cross_Match por el actor.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Seleccionar paciente a trasplantar del Seleccionar donante y luego accede a la opción Insertar resultado Cross_Match.	
2.		Muestra los datos del donante: <ul style="list-style-type: none"> • Nombre(s).

Capítulo 2: Características del Sistema

		<ul style="list-style-type: none"> • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Muestra los datos del paciente:</p> <ul style="list-style-type: none"> • Nombre(s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Y brinda la posibilidad de seleccionar si el resultado de Cross_Match es positivo o negativo.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar Insertar resultado Cross_Match. • Cancelar operación. Ver Sección 1: “Cancelar operación.”
3.	Selecciona si el resultado de Cross_Match es positivo o negativo.	
4.	Selecciona la opción Insertar resultado Cross_Match.	
5.		Valida los datos. Si hay datos incompletos, ver Sección 2: “Existen datos incompletos.”. Si hay datos incorrectos, ver Sección 3: “Existen datos incorrectos.”
6.		Inserta resultado Cross_Match.
7.		Muestra los datos de resultado Cross_Match. Ver

Capítulo 2: Características del Sistema

		caso de uso: “Ver detalles de resultado Cross_Match”.
8.		El caso de uso termina.
Sección 1: “Cancelar operación.”		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Sección 2: “Existen datos incompletos.”		
Flujo básico		
	Actor	Sistema
1.		Muestra un indicador sobre los campos incompletos.
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 3: “Existen datos incorrectos”.		
	Actor	Sistema
1.		Muestra un indicador sobre los campos incorrectos.
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Relaciones	CU Incluidos	Ver detalles de resultado Cross_Match. Ver caso de uso: “Ver detalles de resultado Cross_Match”.
	CU Extendidos	No existe.

Capítulo 2: Características del Sistema

Requisitos no funcionales	No existe.
Asuntos pendientes	No existe.

En calidad de conclusiones parciales de este capítulo se plantea que:

- ✓ La descripción de los procesos del negocio, permitió una mejor comprensión de cómo se realiza el trabajo en el Centro Coordinador Nacional de Trasplante, obteniendo las funcionalidades del sistema.
- ✓ La confección de los artefactos permitió dar comienzo al diseño del sistema propuesto, el cual requiere de un mayor esfuerzo por parte del equipo de desarrollo.

Capítulo 3: Diseño del Sistema

Este capítulo tiene como propósito fundamental definir la arquitectura y los patrones a utilizar. Para ello se realiza la traducción de los requisitos y se describe la implementación que tendrá el sistema. Además, se obtienen los diagramas del diseño y su correspondiente descripción. También se describen las principales clases del diseño, con el objetivo de lograr un mayor entendimiento de estas para dar paso a la implementación.

3.1. Descripción de la arquitectura

La arquitectura de software es de especial importancia, ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema (desempeño, seguridad, modificabilidad). El no crear este diseño desde etapas tempranas del desarrollo puede limitar severamente el que el producto final satisfaga las necesidades de los clientes. Es así que la arquitectura de software juega un papel fundamental dentro del desarrollo. (46)

El sistema propuesto presenta como parte de la línea base de su arquitectura el patrón Modelo Vista Controlador (MVC), el cual separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos: el modelo para la administración de los datos; la vista que muestra las interfaces de usuario y el controlador para la gestión de las acciones del usuario, invocando cambios tanto en la vista como en el modelo. Con el uso de este patrón se garantiza la no dependencia entre la vista y el modelo, por lo que un cambio en la vista trae consigo poco o ningún impacto en los datos o en la lógica de negocio.

La Vista está desarrollada básicamente con JSF, usando componentes Seam de Interfaz de Usuario, la librería de componentes Richfaces y Ajax4jsf, y se utiliza Facelets como motor de plantillas. Además, en el controlador se utiliza JBoss Seam como Framework de integración. En el modelo se utiliza Hibernate como herramienta de mapeo objeto relacional, que es la implementación de EJB versión 3.0 y JPA.

La integración de elementos existentes en la aplicación se logra mediante el framework JBoss Seam que permite la utilización de anotaciones con las que es posible acceder al modelo de datos directamente desde la vista, característica que permite un mejor funcionamiento del sistema.

3.2. Modelo de diseño

En el Modelo de diseño se estructura el sistema para que soporte tanto los requisitos funcionales como los no funcionales y las restricciones que se imponen, produciéndose un refinamiento del análisis y constituyendo una base para la implementación. Además, está constituido por los diagramas de clases del diseño. En este capítulo se especifican mejor los casos de uso mediante las clases.

Para lograr la optimización de esta actividad, se hace necesario su estructuración u organización en entidades que posibiliten un mayor grado de manipulación, por lo que se fragmenta en subsistemas o paquetes. La metodología RUP propone que el modelo de diseño contenga una breve descripción de paquetes y subsistemas de diseño que se representan en una jerarquía, diagramas de clases y de interacción, los cuales constan con clases, interfaces y relaciones.

3.2.1. Diagrama de clases del diseño

El diagrama de clases del diseño permite visualizar, especificar y documentar modelos estructurales. Está compuesto por un conjunto de clases, interfaces, entidades y relaciones, las cuales se utilizan para modelar la vista de diseño estática de un sistema. Además, permite construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Para realizar el diagrama de clase del diseño, se utilizó la extensión de UML, la cual facilita la utilización de los estereotipos web necesarios. Las vistas contienen a los formularios, los cuales son los encargados de hacer las peticiones al servidor. Estas utilizan un conjunto de librerías que permiten construir las mismas y están basadas en el Framework JSF.

Es importante resaltar que una página cliente es construida por una sola página servidora. Además, se utiliza JavaScript para incorporar dinamismo a las páginas y realizar validaciones del lado del cliente. El controlador de JSF (servlet), intercepta las peticiones enviadas por las vistas y realiza una secuencia de pasos que forman parte del ciclo de vida de JSF (validaciones de sintaxis, etc.) e interviene en la construcción de las vistas. Dicho controlador está contenido en el Framework JSF, el cual forma parte de la plataforma JEE versión 5. Además, este controlador invoca al controlador de SEAM (servlet), el cual enruta las peticiones para cada bean (páginas servidoras), en dependencia de la opción.

Capítulo 3: Diseño del Sistema

Cada bean contiene la lógica de programación que permite dar respuesta a las peticiones. Estas crean instancias de las entidades para persistir la información a través de la interfaz Entity Manager contenida en el motor de persistencia JPA. Utilizan el Framework EJB, el cual facilita el desarrollo de la lógica de negocio. Las entidades utilizan el Framework EJB y constituyen un tipo de bean, en este caso, bean de entidad. También utilizan el Framework Hibernate (ORM), para realizar el mapeo objeto/relacional. EL Framework Hibernate está regido por la especificación JPA que forma parte de la especificación EJB 3.0 y este a su vez de la plataforma JEE 5.

EL Framework SEAM permite la comunicación entre la presentación y la lógica de negocio y forma parte de la plataforma JEE 5. En algunos casos, las clases del modelo hacen uso de las interfaces de JPA: Entity Query y Entity Transaction a través de la interfaz Entity Manager para realizar consultas y encapsular las mismas de forma atómica.

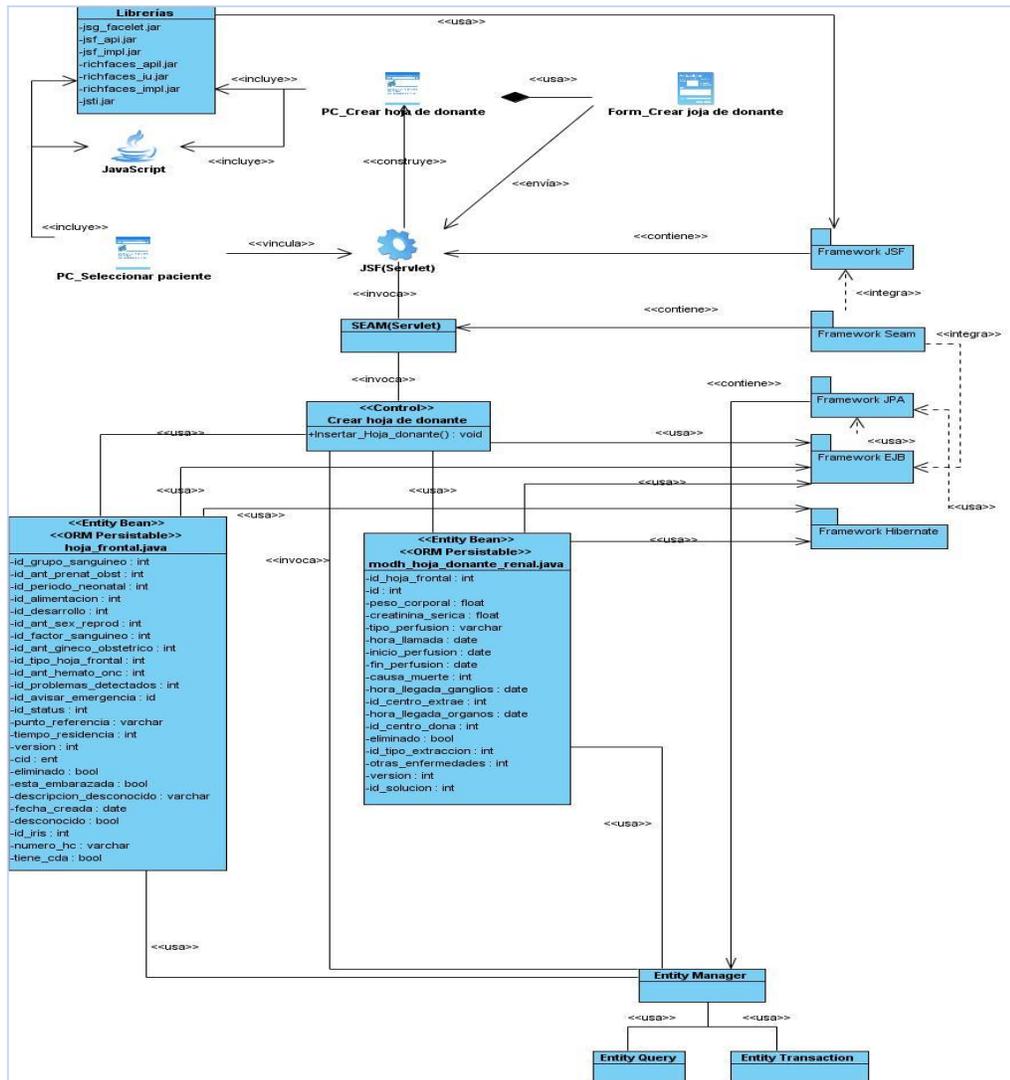
Para nombrar estas clases en el Diagrama de Clase se siguió la siguiente estructura: PC_<NombreClaseCliente>, Form_<NombreFormulario>, JSF (Servlet), para las páginas clientes, formularios, clases servidoras ubicadas en la Capa de Presentación y clases de la Lógica de Negocio respectivamente. En el diagrama también están presentes las clases entidades así como las relaciones que se establecen entre estas y las clases controladoras. Las mismas pueden conocerse mediante los estereotipos <<Entity Bean>> y <<ORM Persistable>>.

En el diseño de la aplicación se utiliza el patrón General Responsibility Assignment Software Patterns (GRASP), con el objetivo de describir los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Dentro de estos, los patrones Experto, Creador, Alta cohesión y Bajo acoplamiento. El Experto y Creador evidencian su uso mediante la asignación a cada clase de las tareas o responsabilidades que estas pueden realizar en dependencia de la información que contienen. Además, definen quién será el responsable de crear una instancia de una clase. 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 se encuentren aislados.

Capítulo 3: Diseño del Sistema

A continuación se presentan algunos ejemplos de diagramas de clases del diseño correspondientes al diagrama de caso de uso del sistema “Gestionar donante”. Ver Anexos para una mejor comprensión del resto de los de diagramas de clases del diseño.

Figura 7: Crear Hoja de Donante



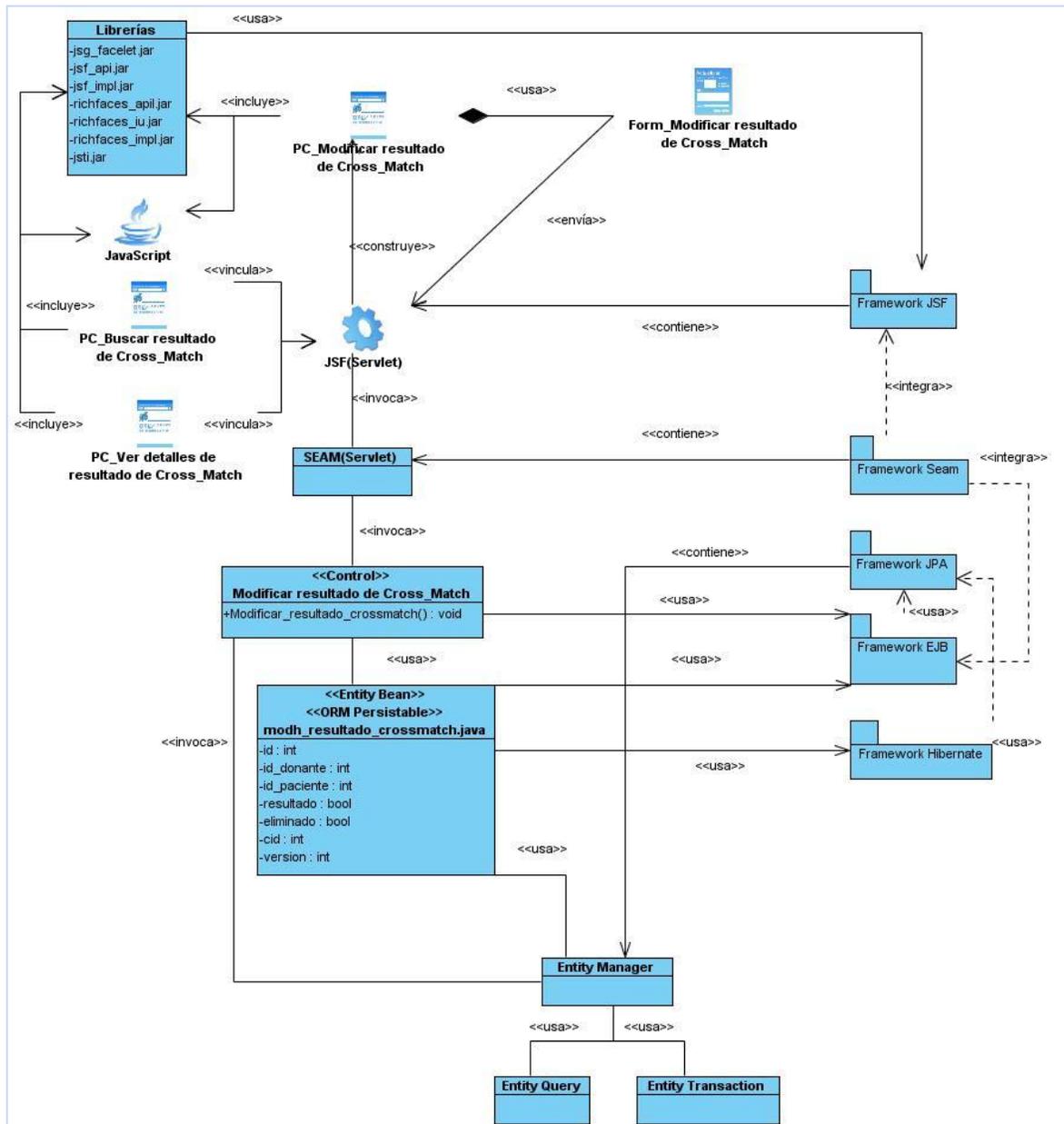


Figura 8. Modificar resultado de Cross_Match.

Capítulo 3: Diseño del Sistema

Descripción de clases del diseño.

A continuación, se explican algunas de las clases identificadas para su futura implementación. También serán descritas algunas de las responsabilidades que realizarán las páginas servidoras que responden a la lógica del negocio.

Descripción de páginas clientes.

Tabla 11. Descripción página cliente PC_Crear Hoja de donante.

Nombre	PC_Crear Hoja de donante.
Tipo de Clase	Página Cliente.
Descripción General	Página web que se ejecuta del lado del cliente, le permite al Especialista en trasplante introducir los datos para crear la hoja de donante.

Tabla 12. Descripción página cliente PC_Buscar donantes.

Nombre	PC_Buscar donantes.
Tipo de Clase	Página Cliente.
Descripción General	Página web que se ejecuta del lado del cliente, le permite al Especialista en trasplante introducir los datos para buscar donantes.

Tabla 13. Descripción página cliente PC_Ver datos de Hoja de donante.

Nombre	PC_Ver datos de Hoja de donante.
Tipo de Clase	Página Cliente.
Descripción General	Página web que se ejecuta del lado del cliente, le permite al Especialista en trasplante ver los datos de la hoja de donante.

Tabla 14. Descripción página cliente PC_Seleccionar paciente a trasplantar.

Capítulo 3: Diseño del Sistema

Nombre	PC_Seleccionar paciente a trasplantar.
Tipo de Clase	Página Cliente.
Descripción General	Página web que se ejecuta del lado del cliente, le permite al Especialista en trasplante seleccionar un paciente a trasplantar.

Tabla 15. Descripción página cliente PC_Generar reporte de cantidad de receptores por hospital de procedencia.

Nombre	PC_Generar reporte de cantidad de receptores por hospital de procedencia.
Tipo de Clase	Página Cliente.
Descripción General	Página web que se ejecuta del lado del cliente, le permite al Especialista en trasplante generar un reporte que refleja la cantidad de receptores por hospital de procedencia.

Descripción de páginas servidoras.

Tabla 16. Descripción página servidora JSF (Servlet) de Seleccionar paciente a trasplantar.

Nombre	JSF (Servlet) de Seleccionar paciente a trasplantar.
Tipo de Clase	Página Servidora.
Descripción General	La clase JSF (Servlet) es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Seleccionar paciente a trasplantar. Codifica la información y construye las estructuras que serán enviadas a la capa de negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 17. Descripción página servidora JSF (Servlet) de Insertar resultado de Cross_Match.

Nombre	JSF (Servlet) de Insertar resultado de Cross_Match.
---------------	---

Tipo de Clase	Página Servidora.
Descripción General	La clase JSF (Servlet) es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente PC_Seleccionar paciente a trasplantar. La servidora construye las estructuras que serán enviadas a la capa de negocio. Se invoca al método del negocio para mostrar los nuevos datos, y concluye la ejecución de sus responsabilidades.

Descripción de páginas controladoras.

Tabla 18. Descripción página controladora Ver datos de Hoja de donante.

Nombre	Ver datos de Hoja de donante.
Tipo de Clase	Controladora.
Descripción General	La clase Ver datos de Hoja de donante contiene el método Mostrar_datos () de tipo void y usa la entidad <<Entity Bean>> modh_hoja_donante_renal.java, la cual presenta disímiles de atributos, algunos de ellos son los siguientes: id_hoja_frontal: int, id_tipo_extraccion: int, id_solucion: int, causa_muerte: int. Además, invoca a la interface Entity manager.

Tabla 19. Descripción página controladora Seleccionar paciente a trasplantar.

Nombre	Seleccionar paciente a trasplantar.
Tipo de Clase	Controladora.
Descripción General	La clase seleccionar paciente a trasplantar contiene el método Buscar_paciente () y usa la entidad <<Entity Bean>> modh_hoja_trasplante_renal.java, la cual presenta los siguientes atributos: hipersensibilidad: int, prioridad: bool, fecha_inicio_tratamiento_sustitutivo: date, método_dialisis: int, suero: bool. Además, invoca a la interface Entity

manager.

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

La Programación Orientada a Objetos es uno de los paradigmas presentes en la ingeniería del software, el cual permite un aprovechamiento más eficiente del código ya construido, denominado reutilización del código. La reutilización de código presenta diversas mejoras para los desarrolladores de cualquier sistema, ya que permite la optimización y ahorro en la construcción de código, así como economizar el tiempo y reducir la redundancia generadas. Además, tiene varias ventajas como son: fiabilidad, eficiencia, reducción, de coste y consistencia. El proceso de abstracción es una práctica eficaz para la reutilización, ya que la misma permite mantener el código reusable en un solo lugar para luego ser llamado donde sea requerido. (47)

Para el desarrollo del sistema se utilizan componentes del Sistema de Información Hospitalaria alas HIS, entre ellos la clase Active Module, que brinda la información sobre qué módulo y entidad se encuentra el usuario que está utilizando el sistema; y la clase User, la cual, ofrece los datos del usuario que se encuentra registrado. Además, se utilizan los módulos Admisión y Configuración; y las funcionalidades “Seleccionar paciente” e “Exportar”. Del módulo Trasplante de hígado se utiliza la funcionalidad “Asignar donante a paciente”.

En calidad de conclusiones parciales se puede afirmar que:

- ✓ La elaboración de los diagramas de clases del diseño permitió visualizar e identificar las clases fundamentales que deben ser definidas para que el sistema funcione satisfactoriamente.
- ✓ La investigación realizada referente al patrón de arquitectura Modelo Vista Controlador, dio a conocer que es muy importante para llevar a cabo una implementación organizada y cuyo resultado sea un software robusto.
- ✓ El uso de los patrones de diseño incorporó buenas prácticas al resultado de la investigación.

Capítulo 4: Implementación

En el presente capítulo se describe lo relacionado con la implementación del sistema de acuerdo a la arquitectura utilizada en la investigación. Se representa el modelo de datos, se explican sus tablas y atributos, obteniéndose el diagrama de clases persistentes para generar a partir del mismo el scripts de la Base de Datos (BD). Además, se modelan los diagramas de componentes y de despliegue, lo que posibilita que quede conformado el modelo de implementación. También, se explica el tratamiento de errores, la seguridad del sistema y los estándares utilizar.

4.1. Modelo de datos

El modelo de datos es el artefacto más importante concebido por RUP dentro del proceso, que incluye generar la base de datos por esta metodología. Este modelo proporciona una representación visual y física de los datos persistentes del sistema, que en el futuro serán la base de datos. Se obtiene a partir del diagrama de clases persistentes y su forma se expresa mediante un diagrama de UML, siendo sus elementos esenciales del modelo las entidades, los atributos y las relaciones entre las entidades.

Las entidades son objetos que guardan información necesaria para el sistema. Su símbolo es un rectángulo. Los atributos son características de una entidad, se representan colocando su nombre dentro del rectángulo de la entidad. Los atributos se clasifican en: obligatorios, opcionales, claves foráneas y claves primarias (estas se dividen en simples y compuestas). Gráficamente la clave primaria se representa con un signo de suma y la foránea con un símbolo de número. Las relaciones muestran la asociación entre dos entidades, representadas por una línea que une a las entidades involucradas.

A continuación se muestra el modelo de datos del módulo Trasplante de riñón.

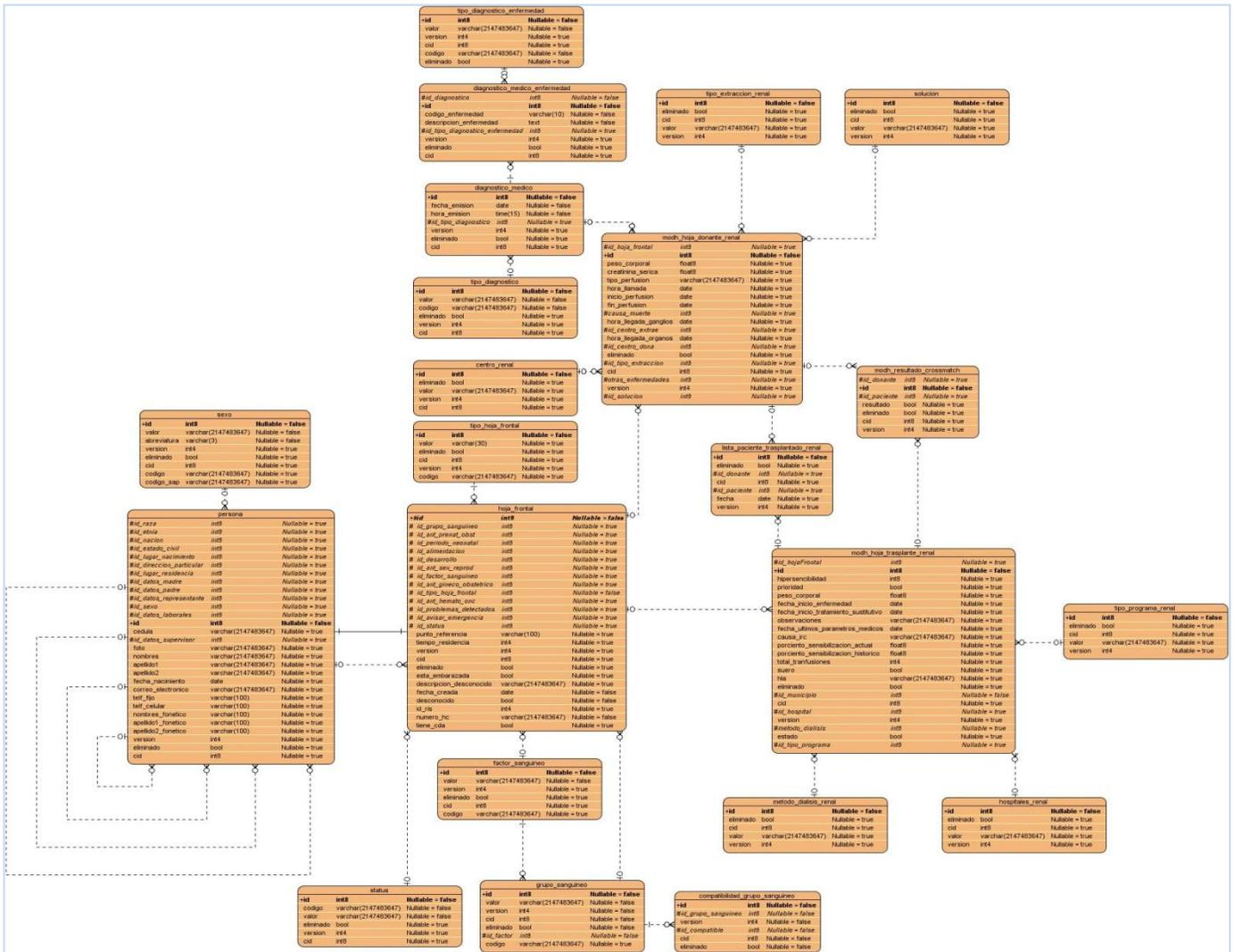


Figura 9. Modelo de datos.

4.1.1. Descripción de las tablas

Los siguientes atributos son comunes a todas las entidades, ya que fueron agregados con el objetivo de facilitar la implementación de algunas funcionalidades del sistema.

Tabla 20. Descripción de atributos comunes entre todas las entidades.

Atributo	Tipo	Descripción
versión	integer	Indica con qué versión de la entidad se está trabajando. Es usado para garantizar que se está trabajando con la versión de la entidad más actualizada que existe en la base de datos.
eliminado	bool	Permite la eliminación lógica con que cuenta el sistema, cuando está en verdadero indica que la entidad está eliminada.
cid	bigint	Permite identificar quién realiza alguna acción sobre la entidad.

Tabla 21. Descripción de la tabla modh_lista_donante.

modh_hoja_donante_renal		
Entidad que recoge la información relacionada con los datos de los donantes en el área de trasplante renal.		
Atributo	Tipo	Descripción
id_hoja_frontal	int	Esta es una llave foránea que contiene todos los datos de la tabla hoja_frontal de hc_local.
id	int	Es la llave primaria de esta tabla.
peso_corporal	float	Este es un atributo de tipo float que registra el peso del donante.
creatinina_serica	float	Este es un atributo de tipo float que registra la creatinina sérica del donante.
tipo_perfusión	varchar	Este es un atributo de tipo varchar que registra el tipo de perfusión del donante.
hora_llamada	date	Este es un atributo de tipo date que registra la hora de llamada.
inicio_perfusión	date	Este es un atributo de tipo date que registra la hora en que se inicia la perfusión.
fin_perfusión	date	Este es un atributo de tipo date que registra la hora en que concluye la perfusión.
causa_muerte	date	Esta es una llave foránea que contiene todos los datos de la tabla causa_muerte.
hora_llegada_ganglios	varchar	Este es un atributo de tipo date que registra la hora de llegada de

		los ganglios del donante.
id_centro_extrae	int	Esta es una llave foránea que contiene todos los datos de la tabla centro_extrae.
hora_llegada_organos	date	Este es un atributo de tipo date que registra la hora de llegada de los órganos del donante.
id_solucion	int	Esta es una llave foránea que contiene todos los datos de la tabla solución.
id_centro_dona	in	Esta es una llave foránea que contiene todos los datos de la tabla centro_dona.
otras_enfermedades	int	
id_tipo_extraccion	int	Esta es una llave foránea que contiene todos los datos de la tabla tipo_extraccion_renal.

Tabla 22. Descripción de la tabla modh_resultado_crossmatch

modh_resultado_crossmatch		
Entidad que recoge la información relacionada con los datos del resultado de Cross_Match.		
Atributo	Tipo	Descripción
id_donante	int	Esta es una llave foránea que contiene todos los datos de la tabla modh_hoja_donante_renal.
id	int	Este atributo es la llave primaria de esta tabla.
id_paciente	bigint	Esta es una llave foránea que contiene todos los datos de la tabla modh_hoja_trasplante_renal.
resultado	bool	Este es un atributo de tipo bool que registra el tipo de resultado.

Tabla 23. Descripción de la tabla modh_hoja_trasplante_renal.

modh_hoja_trasplante_renal		
Entidad que recoge la información relacionada con los datos del receptor.		
Atributo	Tipo	Descripción

Capítulo 4: Implementación

id_hojaFrontal	Int	Esta es una llave foránea que contiene todos los datos de la tabla hoja_frontal de hc_local.
id	Int	Este atributo es la llave primaria de esta tabla.
hipersensibilidad	Int	Este es un atributo de tipo int que almacena el porcentaje de hipersensibilidad del receptor.
prioridad	Bool	Este es un atributo de tipo bool que registra si el paciente tiene prioridad.
peso_corporal	Float	Este es un atributo de tipo float que registra el peso del receptor.
fecha_inicio_enfermedad	Date	Este es un atributo de tipo date que registra la fecha en que el receptor comenzó con IRC.
fecha_inicio_tratamiento_sustitutivo	Date	Este es un atributo de tipo date que registra la fecha en que el receptor comenzó a hacerse las diálisis.
observaciones	varchar	Este es un atributo de tipo varchar que registra las observaciones.
fecha_ultimos_parametros_medicos	Date	Este es un atributo de tipo date que registra la fecha de los últimos parámetros médicos.
causa_irc	varchar	Este es un atributo de tipo varchar que registra la causa que dio origen a la IRC del receptor.
por ciento_sensibilizacion_actual	Float	Este es un atributo de tipo float que registra el por ciento de hipersensibilidad actual del receptor.
por ciento_sensibilizacion_historico	Float	Este es un atributo de tipo float que registra el por ciento de hipersensibilidad histórica del receptor.
total_tranfusiones	Int	Este es un atributo de tipo int que registra el total de transfusiones que se le han hecho al receptor.
suero	Bool	Este es un atributo de tipo bool que registra receptor tiene suero.
hla	varchar	Este es un atributo de tipo varchar que registra el hla.
id_municipio	int	Esta es una llave foránea que contiene todos los da-

		tos de la tabla hospitales_renal.
id_hospital	Int	Esta es una llave foránea que contiene todos los datos de la tabla tipo_programa_renal.
método_dialisis	Int	Este es un atributo de tipo int que registra el método de diálisis del receptor.
estado:	Bool	Este es un atributo de tipo bool que registra el estado del receptor.
id_tipo_programa	In	Esta es una llave foránea que contiene todos los datos de la tabla tipo_programa_renal.

Tabla 24. Descripción de la tabla modh_lista_paciente_trasplantado.

modh_lista_paciente_trasplantado_renal		
Entidad que recoge la información relacionada con los datos de los pacientes trasplantados.		
Atributo	Tipo	Descripción
id	int	Este atributo es la llave primaria de esta tabla.
id_donante	int	Esta es una llave foránea que contiene todos los datos de la tabla modh_hoja_donante_renal.
fecha	date	Este es un atributo de tipo date que registra la fecha.
id_paciente	int	Esta es una llave foránea de la tabla modh_hoja_trasplante_renal.

Tabla 25. Descripción de la tabla solucion.

Solucion		
Entidad que recoge la información sobre la solución que se pueden utilizar para la cirugía del donante.		
Atributo	Tipo	Descripción
id	int	Este atributo es la llave primaria de esta tabla.
valor	varchar	Este es un atributo de tipo varchar que registra los tipos de soluciones que se utilizan para la cirugía del donante.

Tabla 26. Descripción de la tabla tipo_extraccion_renal.

tipo_extraccion_renal		
Entidad que recoge la información sobre los tipos de extracciones que se pueden utilizar para la cirugía del donante.		
Atributo	Tipo	Descripción
id	int	Este atributo es la llave primaria de esta tabla.
valor	varchar	Este es un atributo de tipo varchar que registra los tipos de extracciones que se pueden utilizar para la cirugía del donante.

Tabla 27. Descripción de la tabla método_dialisis_renal.

método_dialisis_renal		
Entidad que recoge la información sobre el tipo de diálisis renal que utiliza cada paciente.		
Atributo	Tipo	Descripción
id	int	Este atributo es la llave primaria de esta tabla.
valor	varchar	Este es un atributo de tipo varchar que registra los métodos de diálisis renal que existen

Tabla 28. Descripción de la tabla grupo_sanguineo.

grupo_sanguineo		
Entidad que recoge la información sobre los tipos de grupos sanguíneos que existen.		
Atributo	Tipo	Descripción
id	int	Este atributo es la llave primaria de esta tabla.
id_factor	varchar	Esta es una llave foránea que contiene todos los datos de la tabla factor_sanguineo.
codigo	varchar	Este es un atributo de tipo varchar que registra el código.

Tabla 29. Descripción de la tabla factor_sanguineo.

factor_sanguineo		
Entidad que recoge la información sobre el factor sanguíneo.		
Atributo	Tipo	Descripción
id	int	Este atributo es la llave primaria de esta tabla.
valor	varchar	Este es un atributo de tipo varchar que registra los valores de los factores sanguíneos.
codigo	varchar	Este es un atributo de tipo varchar que registra el código.

Tabla 30. Descripción de la tabla centro_renal.

centro_renal		
Entidad que recoge la información sobre los centros renales.		
Atributo	Tipo	Descripción
id	int	Este atributo es la llave primaria de esta tabla.
valor	varchar	Este es un atributo de tipo varchar que registra los nombres de los hospitales donde se realizan diálisis.

4.2. Implementación

La implementación comienza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El flujo de trabajo de diseño se propone crear un plano del modelo de implementación, por lo que sus últimas actividades están vinculadas a la creación del modelo de despliegue.

El modelo de implementación del módulo Trasplante de riñón expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los mismos contienen componentes y sus relaciones, dividiendo el sistema en partes más manejables. Esto posibilita la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio (ver Figura 10).

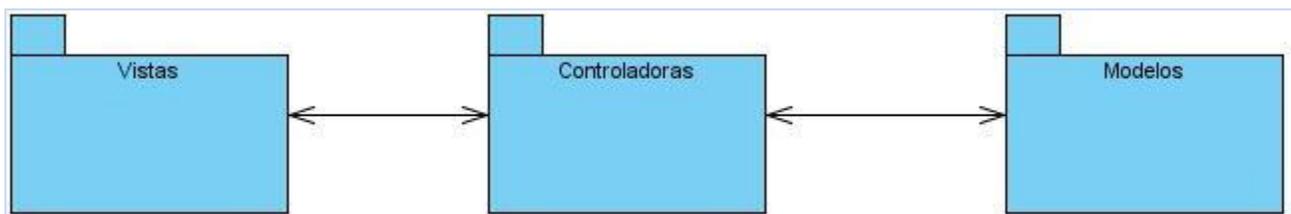


Figura 10. Diagrama de paquetes.

4.2.1. Diagrama de componentes

El diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables o paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

A continuación se muestra el diagrama de componentes del módulo Trasplante de riñón, conformado por el paquete Vistas (ver Figura 11), el paquete Controlador (ver Figura 12) y el paquete Modelo (ver Figura 13).

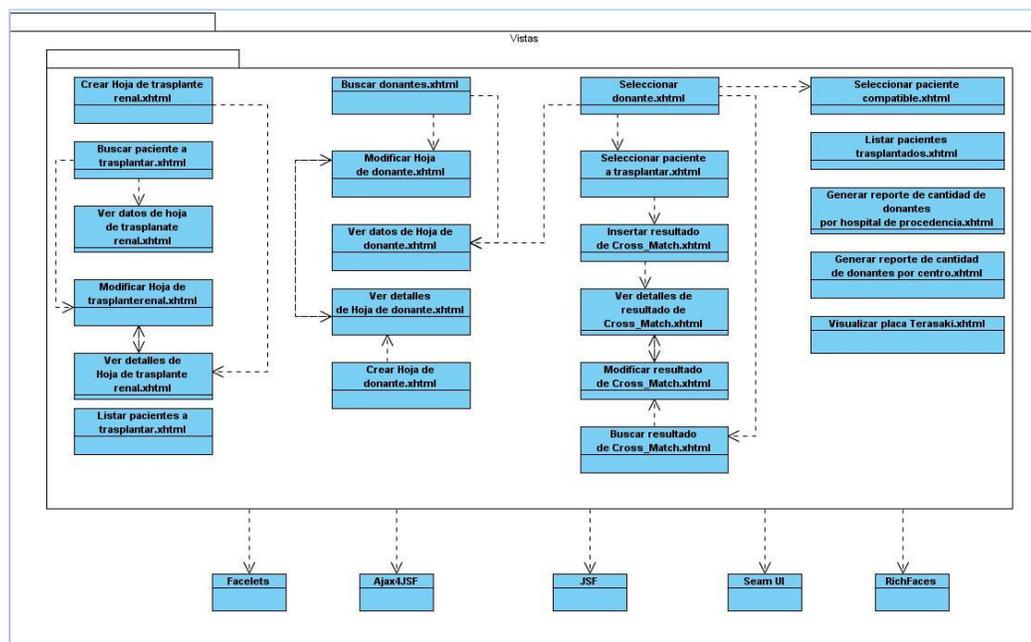


Figura 11. Diagrama de componentes del paquete Vistas.

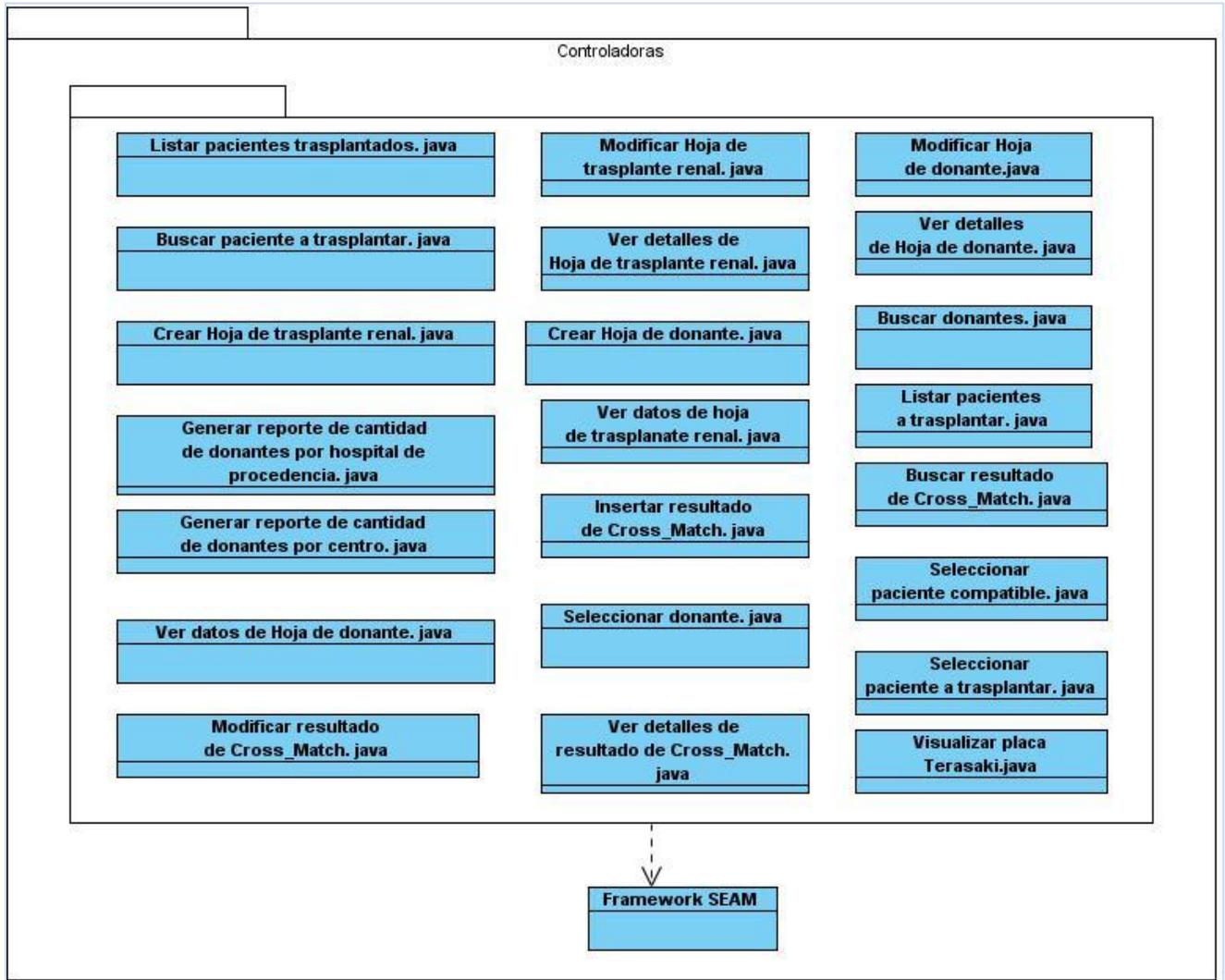


Figura 12. Diagrama de componentes del paquete Controlador.

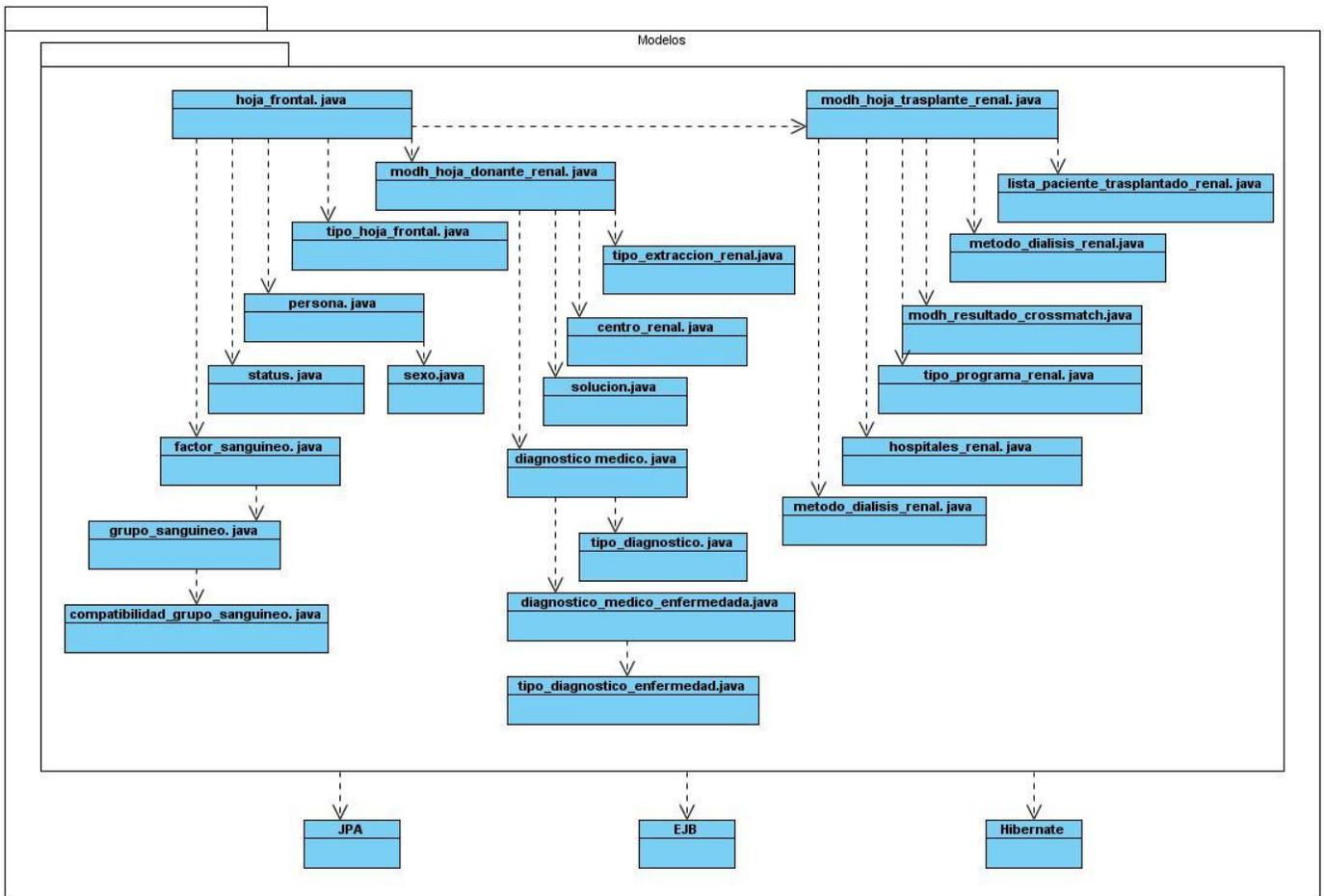


Figura 13. Diagrama de componentes del paquete Modelo.

4.2.2. Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes de hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes de software que representan manifestaciones del código en tiempo de ejecución. (48) Además, está diseñado por nodos, los cuales representan un recurso físico computacional, por lo general con memoria y capacidad de procesamiento.

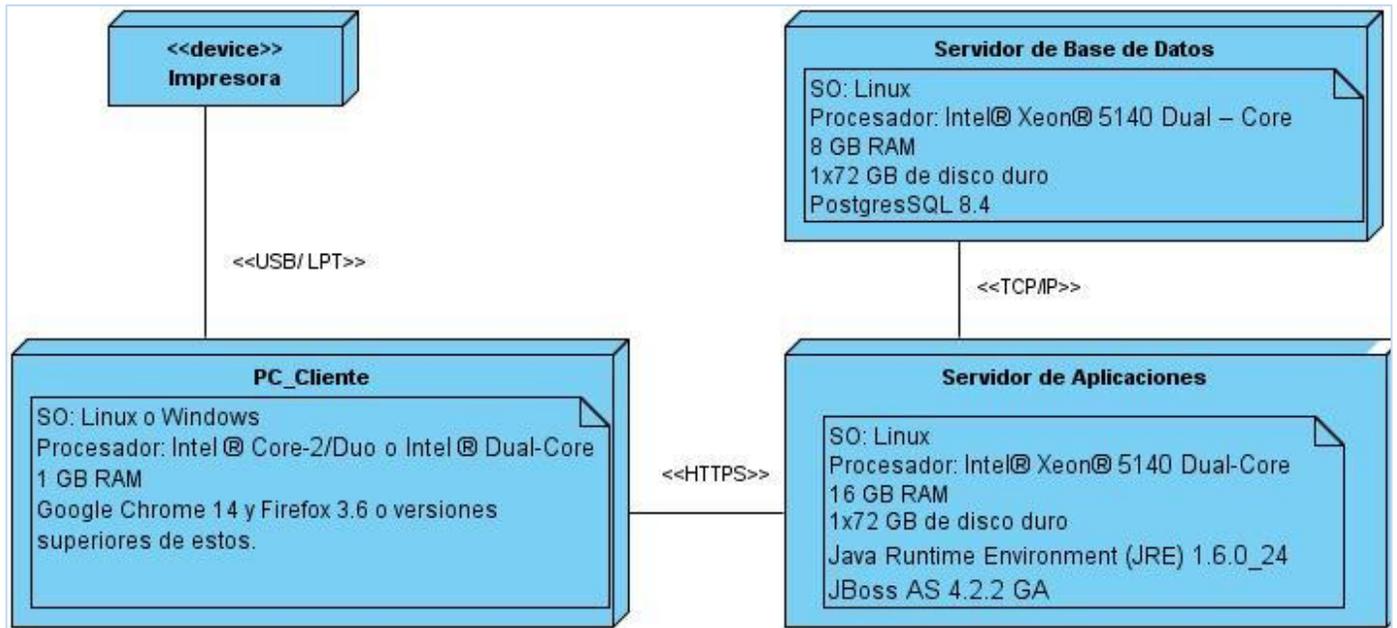


Figura 14. Diagrama de Despliegue.

4.3. Tratamiento de errores

Para darle cumplimiento a los tratamientos de errores es de vital importancia mencionar, que un error es un fallo que ocurre en un programa, no logrando el producto deseado y afectando al cliente. Las excepciones son el mecanismo recomendado para la propagación de errores que se produzcan durante la ejecución de las aplicaciones. Otro punto de vista es cuando se desarrolla un software, que se intenta proveer de cierta funcionalidad al usuario, si esa funcionalidad no se cumple, se puede decir que el software presenta errores. Estos son detectables por el usuario, lo que influye en la calidad externa del software.

En el sistema propuesto se utilizan todos los elementos que brinda la plataforma para el tratamiento de excepciones. El control de las excepciones se lleva a cabo a toda porción de código, donde pueda surgir alguna situación inesperada, especialmente donde se ejecutan sentencias que manipulan los datos que viajan desde y hacia la base de datos, aunque también se controlan los errores que pueden surgir en la

validación de datos provenientes de la interfaz de usuario; utilizando para ello controles de conversión y validación; como el validador que proporciona el framework Hibernate, el cual implementa la validación de datos multicapa, donde las restricciones y constantes de validación son expresadas mediante anotaciones en el Modelo (Entidades), y chequeadas en distintas capas de la aplicación

Las diferentes tecnologías utilizadas permiten capturar y controlar posibles situaciones desde diferentes puntos del sistema. En las páginas clientes, se cuenta con un conjunto de componentes denominados validadores. Además, el marco de trabajo Seam brinda un potente conjunto de excepciones predefinidas, que conjuntamente con la clase FacesMessages, permite tratar estas situaciones desde los controladores correspondientes y mostrar mediante la clase antes mencionada, los resultados del tratamiento. Seam permite además, mediante el fichero de configuración page.xml, todo un flujo de navegación basado en excepciones.

4.4. Seguridad

La seguridad informática es un estado de cualquier tipo de información que indica que ese sistema está libre de peligro, daño o riesgo. Se entiende como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo. (50) Para obtener un sistema seguro se deben tener en cuenta dos elementos básicos como la integridad (la información solo puede ser modificada por quien está autorizado y de manera controlada) y la confidencialidad (la información sólo debe ser visible para los autorizados).

Para analizar la seguridad de un sistema se debe pensar en la forma en que el mismo pudiera sufrir determinada pérdida o daño, para lo cual es necesario identificar las debilidades del sistema. La seguridad del módulo Trasplante de riñón será implementada, teniendo en cuenta los siguientes aspectos. Se garantiza la seguridad mediante un conjunto de reglas que restringen el acceso a directorios, páginas, controles, opciones del menú y servicios del negocio. Además, para el acceso al sistema, se deben establecer permisos, que serán asignados teniendo en cuenta el rol desempeñado por los usuarios en la institución hospitalaria. Se realiza además, el control a nivel de usuarios y contraseñas, garantizando el acceso solo a los niveles establecidos de acuerdo con la función que realiza cada usuario.

En cuanto al manejo de las contraseñas solo podrán ser cambiadas por el usuario o por el administrador del sistema. Se podrá además, llevar un registro de trazas, para tener control de qué hace cada usuario.

Una traza se registra por cada acción que realiza un usuario sobre el sistema.

4.5. Estrategias de codificación. Estándares y estilos a utilizar

Los estándares de codificación establecen un conjunto de reglas que los desarrolladores deben seguir para escribir el código fuente de un software. Estos tienen como objetivo, que dicho código sea entendible por cualquier desarrollador del grupo de trabajo; garantizando un mantenimiento del sistema más rápido y eficiente, ya sea creando nuevas funcionalidades o modificando las ya existentes.

Para el desarrollo del sistema propuesto se utilizan varios estándares de codificación, tales como:

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

Notación Camell: Se utiliza para denotar variables, parámetros y métodos. Si el identificador es una palabra simple, se escribe todo con minúscula, en caso de que sea compuesta, se escribe con minúscula la primera letra del identificador y las que vienen a continuación con mayúscula.

A continuación se muestran algunas excepciones para la nomenclatura basadas en el estándar a utilizar:

✓ Identación.

Inicio y bloque de fin: Se debe 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 cierres (}) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.

✓ Comentarios, separadores, líneas, espacios en blanco y márgenes.

Capítulo 4: Implementación

Líneas en blanco: Se debe 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: Se deben usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código.

Aspectos generales: 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. Con respecto a los espacios en blanco, no se deben usar: después del corchete abierto y antes del cerrado de un arreglo, después del paréntesis abierto y antes del cerrado ni antes de un punto y coma.

✓ **Variables y Constantes.**

Apariencia de constantes: Se deben declarar las constantes con todas sus letras en mayúscula.

Aspectos generales: El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.

✓ **Clases y Objetos.**

Apariencia de clases y objetos: 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 Pascal Ejemplo: MiClase (). Para el caso de las instancias se comenzará con un prefijo que identificará el tipo de dato, este se escribirá en minúscula.

Apariencia de atributos: 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 Camello.

Aspectos generales: El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

✓ **Bases de datos, tablas, esquemas y campos.**

Capítulo 4: Implementación

Apariencia de la base de datos: Los nombres de las Bases de Datos deben comenzar con el prefijo modh a continuación underscored y luego, el nombre completamente en minúscula, en caso de que sea un nombre compuesto se empleará notación. Los nombres serán cortos y descriptivos.

Apariencia de las vistas: El nombre a emplear para las vistas deben comenzar con el prefijo vt seguido de underscored y el nombre debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.

Apariencia de las tablas: El nombre a emplear para las tablas debe comenzar con el prefijo modh seguido de underscored y luego, debe escribirse todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizará underscored para separarlo.

Tablas que representen Relaciones: El nombre a emplear para estas tablas de relación debe comenzar con el prefijo modh seguido de underscored y el nombre de la tabla, será la concatenación del nombre de las dos tablas que la generaron separados por underscored todo en minúscula.

Tablas que representen nomencladores: El nombre a emplear para estas tablas de relación debe comenzar con el prefijo n seguido de underscored. El nombre será corto y descriptivo, todo en minúscula.

Apariencia de los procedimientos almacenados: El nombre a emplear para los procedimientos debe comenzar con el prefijo pa seguido de underscored y luego debe escribirse todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizará underscored para separarlo.

Apariencia de los campos: El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.

Nombre de los campos: Todos los campos identificadores van a comenzar con el identificador id seguido de underscored y posteriormente el nombre del campo.

Sentencias SQL: Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

Aspectos generales: El nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los

procedimientos almacenados, deben permitir que con sólo leerlos se conozca el propósito de los mismos. Controles.

Apariencia de los controles: El nombre que se le da a los controles deben comenzar con las primeras letras en minúscula, las cuales identificarán el tipo de dato al que se refiere (ver Tabla 30), en caso de que sea un nombre compuesto se empleará notación CamellCasing.

Tabla 31. Controles.

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

Las cuestiones anteriormente planteadas permiten arribar a las siguientes conclusiones parciales:

- ✓ La realización del modelo de datos proporcionó una representación visual y física de los datos persistentes del sistema, que en el futuro serán la base de datos.
- ✓ La confección del diagrama de despliegue permitió visualizar las relaciones físicas entre los componentes de hardware y software del sistema, mostrando los requisitos mínimos que estos deben cumplir.
- ✓ Se estableció el tratamiento de excepciones, la seguridad del sistema y las estrategias de codificación que se emplean para el desarrollo del sistema.
- ✓ La adopción de estándares de codificación permitió implementar las funcionalidades identificadas con un código legible y de fácil entendimiento para su posterior mantenimiento.

Conclusiones

Una vez finalizado el desarrollo de la investigación se concluye que:

- ✓ El estudio realizado de los numerosos sistemas de gestión de trasplante arrojó como resultado un conjunto de funcionalidades, que sustentaron el desarrollo de la solución propuesta.
- ✓ La caracterización de la metodología, las técnicas, tecnologías y herramientas utilizadas facilitó el trabajo con las mismas y permitió la obtención de los artefactos correspondientes al desarrollo del sistema.
- ✓ El uso de patrones de diseño incorporó buenas prácticas al resultado de la investigación.
- ✓ La adopción de estándares de codificación permitió implementar las funcionalidades identificadas con un código legible y de fácil entendimiento para su posterior mantenimiento.

Recomendaciones

Sobre la presente investigación, se recomienda:

- ✓ Permitir exportar la placa Terasaki.

Referencias Bibliográficas

1. Historia de la nefrología en Latinoamérica. [En línea] 2 de enero de 2013. <http://www.nefrologia-urologia.blogspot.com/>.
2. **Camacho, Herminio, Ronquillo, Ricardo y Menéndez, Marina.** OPCIONES. [En línea] 15 de agosto de 2012. <http://www.opciones.cu/cuba/2012-08-15/suman-mas-de-5-000-los-trasplantes-renales-en-cuba/>.
3. Sociedad chilena de Nefrología. [En línea] [Citado el: 1 de enero de 2013.] <http://www.nefro.cl/informacion-pacientes/35-trasplante-de-rinon-.html>.
4. ¿Qué es un órgano? [En línea] 22 de abril de 2009. [Citado el: 22 de abril de 2012.] <http://repasandomas.blogspot.com/2009/11/que-es-un-organo.html>.
5. MedlinePlus. [En línea] 6 de agosto de 2012. <http://www.nlm.nih.gov/medlineplus/spanish/dialysis.html>.
6. **Akin, Lousie y Pierce, Daphne.** Cirugías y Procedimientos. [En línea] 27 de septiembre de 2009. <http://carefirst.staywellsolutionsonline.com/Spanish/RelatedItems/90,P06161>.
7. **López, Ángeles.** Preparar al paciente para evitar el rechazo al riñón. [En línea] 28 de julio de 2011. <http://www.elmundo.es/elmundosalud/2011/07/28/noticias/1311853471.html>.
8. Estudios de Histocompatibilidad. [En línea] [Citado el: 1 de mayo de 2013.] http://www.pricai.com.ar/estudios_histocompatibilidad_cross_match.php.
9. **Lorenci, Miguel.** Un nuevo sistema agilizará la gestión de los trasplantes. [En línea] 1 de agosto de 2010. <http://www.elcorreo.com/vizcaya/rc/20100901/sociedad/sistema-carla-aporta-seguridad-201009011721.html>.
10. **Hernández, Esperanza .** Arranca la segunda fase de Carla. [En línea] 18 de septiembre de 2011. <http://www.wayerless.com/2011/09/arranca-la-segunda-fase-de-carla-el-sistema-movil-que-agiliza-los-trasplantes-en-espana/>.
11. Sistema Nacional de Información de Procuración y Trasplante de la República Argentina. [En línea] [Citado el: 29 de enero de 2012.] <http://sintra.incucai.gov.ar/intro.html>.
12. **Mahillo, Beatriz Durán.** Gestión de información sobre donación y trasplante de órganos. [Online] noviembre 18, 2010. http://www.msps.es/estadEstudios/estadisticas/sisInfSanSNS/4ForoSISNS/Documentos/p_bmahillo.pdf.
13. PLIEGO DE PRESCRIPCIONES TÉCNICAS PARA LA CONTRATACIÓN DE ASISTENCIA. Organización Nacional de Trasplante de España. 1989.

Referencias Bibliográficas

14. Trasplante, Organización Nacional de. España: s.n. 2008.
15. Definición e Implementación de un Sistema de Información para un Proceso de Desarrollo de Software. [En línea] [Citado el: 6 de junio de 2013.] <http://upcommons.upc.edu/pfc/bitstream/2099.1/14607/2/Anexos%20A-H.pdf>.
16. Revista cubana de Ciencias Informáticas. [En línea] [Citado el: 6 de junio de 2013.] <http://www.uci.cu/entorno-productivo>.
17. **Ramos, Kariné Blanco.** Experiencias del programa de mejora de procesos en la Universidad de Ciencias Informáticas. [En línea] [Citado el: 6 de junio de 2013.] <http://www.google.com.cu/url?sa=t&rct=j&q=El%20programa%20de%20mejora%20que%20se%20desarrolla%20actualmente%20en%20la%20UCI%20tiene%20como%20objetivo%20la%20definici%C3%B3n%20e%20implementaci%C3%B3n%20de%20los%20procesos%20necesarios%20con%20el%20fin%20de>
18. Carrillo, Isaías Pérez, Pérez, Rodrigo González y Rodríguez, Aureliano David Martín. Metodología de Desarrollo del Software. [En línea] 15 de octubre de 2008. [Citado el: 29 de enero de 2013.] http://www.google.com.cu/url?sa=t&rct=j&q=Las%20Metodolog%C3%ADas%20de%20Desarrollo%20de%20Software%20surgen%20baste%20la%20necesidad%20de%20utilizar%20una%20serie%20de%20procedimientos%20de%20tr%C3%A1gicas%20herramientas%20y%20soporte%20documental%20a%20la%20hora%20de%20desarrollar%20un%20producto%20software.%20Dichas%20metodolog%C3%ADas%20pretenden%20guiar%20a%20los%20desarrolladores%20a%20crear%20un%20nuevo%20software%20pero%20los%20requisitos%20de%20un%20software%20a%20otro%20son%20tan%20variados%20y%20cambiantes%20que%20haya%20dado%20lugar%20a%20que%20exista%20una%20gran%20variedad%20de%20metodolog%C3%ADas%20para%20su%20creaci%C3%B3n.%20&source=web&cd=1&cad=rja&ved=0CCgQFjAA&url=http%3A%2F%2Fsolusoft-g11.googlecode.com%2Ffiles%2FMetodologias%2520de%2520desarrollo.pdf&ei=Wse_UdqBIYKHPdSGgagH&usg=AFQjCNFZ9_K-b0YEM-twoVcsjssFhxWgQQ&bvm=bv.47883778,d.ZWU
19. Desafíos del Curso de Ingeniería de Software. Salazar, Gabriela Bermúdez. 13, Costa Rica : s.n., 2012, Educación en Ingeniería, Vol. 7, págs. 32-43. ISSN 1900-8260.
20. Introducción a JSF (Java Server Faces). Primer artículo de un pequeño manual sobre esta tecnología. [En línea] [Citado el: 27 de enero de 2012.] <http://www.desarrolloweb.com/articulos/2380.php>.
21. **Caiza, Alex Raúl Morillo.** Estudio comparativo de Frameworks Ría para el desarrollo de aplicaciones web con Java Server Faces (JSF). Ibarra-Ecuador: s.n. 2011. pág. 48.

Referencias Bibliográficas

22. Qué es AJAX. [En línea] [Citado el: 24 de enero de 2012.] <http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>.
23. **Sánchez, José Manuel Suárez.** Migración de JSP a Facelets. [En línea] 2008 de octubre de 2008. [Citado el: 7 de junio de 2013.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=migrateJSF2Facelets>.
24. Desafíos del Curso de Ingeniería de Software. Salazar, Gabriela Bermúdez. 13, Costa Rica : s.n., 2012, Educación en Ingeniería, Vol. 7, págs. 32-43. ISSN 1900-8260.
25. Lenguajes HTML y XHTML. [En línea] [Citado el: 27 de noviembre de 2012.] <http://sip.calderon.presidencia.gob.mx/tecnologia/103-lenguajes-html-y-xhtml>.
26. **Eguíluz, Javier Pérez.** Introducción a CSS. [En línea] [Citado el: 28 de enero de 2012.] <http://www.librosweb.es/css/capitulo1.html>.
27. **Álvarez, Amaya Lorenzo; Mora, Mirelio Maure; Hernández, Wilder González; Alemán, Lorena Antelo.** DESARROLLO DE LA ESPECIALIDAD PSICOLOGÍA DEL MÓDULO CONSULTA EXTERNA DEL SISTEMA ALAS-HIS. [Citado el: 19 de junio de 2013] http://www.google.com.cu/url?sa=t&rct=j&q=JBoss+Seam++es+un+framework+que+integra+la+capa+de+presentaci%C3%B3n+%28JSF%29+con+la+capa+de+negocios+y+persistencia+%28EJB%29%2C+funcionando%2C+seg%C3%BAn+versa+su+significado+en+espa%C3%B1ol%2C+como+una+%E2%80%9Ccostura%E2%80%9D+entre+estos+componentes.+Seam+tambi%C3%A9n+se+integra+perfectamente+con+otros+frameworks+como%3A+RichFaces%2C+ICEFaces%2C+MyFaces%2C+Hibernate+y+Spring&source=web&cd=1&ved=0CCoQFjAA&url=http%3A%2F%2Fwww.informatica2013.sld.cu%2Findex.php%2Finformaticasalud%2F2013%2Fpaper%2Fdownload%2F235%2F195&ei=xm_CUYHpFsm7OdCkgGA&usg=AFQjCNGxU_4ok4xfPQdJS-2PJdmvu7BSTg&bvm=bv.48175248,d.ZWU.
28. **López, Estela Suárez y Flores, Claudia Villa.** Desarrollo del Módulo Archivo del Sistema de Información Hospitalaria alas HIS. La Habana : s.n., 2009. pág. 19.
29. **Serra, David Manchado; Franco, Daniel Puntos.** Estudio del servidor de aplicaciones Glassfish y de las aplicaciones J2EE. [En línea] junio de 2010 [Citado el: 19 de junio de 2013]. http://ddd.uab.cat/pub/trerecpro/2013/hdl_2072_206748/SerraManchadoDavidR-ETISa2009-10.txt.
30. **Jorge.** Java: EJB: Enterprise Java Beans. [En línea] [Citado el: 29 de enero de 2012.] <http://www.osmosislatina.com/java/ejb.htm>.
31. **Rondón, Luis.** JAVA J2EE. JPA - Java Persistence API. [En línea] 3 de abril de 2009. [Citado el: 30 de

Referencias Bibliográficas

- enero de 2012.] <http://www.luchorondon.blogspot.com/2009/04/jpa-java-persistence-api.html>.
32. Introducción a JasperReports e iReport (Primera parte). 2006.
33. Que es un reportador. [En línea] noviembre de 2010. <http://www.buenastareas.com/ensayos/Que-Es-Un-Reportador/1127220.html>.
34. PostgreSQL About. [Citado el: 19 de junio de 2013.] <http://www.postgresql.org/about>.
35. **García de Jalón, J., Rodríguez Iñigo, M. J., Alfonso Brazález, A. I., Larzabal, A., Calleja, J., & García, J.** Aprende Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales. Universidad de Navarra . España : s.n., Enero de 2010.
36. Lenguaje Unificado de Modelado o UML. [En línea] [Citado el: 6 de junio de 2013.] <http://resumenuml.bloggratis.es/>.
37. **Cáceres, Isabel Cristina Bernal y Morales, Lisset Cárdenas.** Desarrollo de Software basado en tecnología web para el Módulo de inventarios de la División de Publicaciones y el Control de las Existencias de los productos realizados por la División para la Librería de la Uniersidad Industriasl de Santander. Universidad Industrial de Santander. Bucaramanga-Colombia : s.n., 2009. pág. 30.
38. **Sebastián, Juan.** Modelo Vista Controlador – Definición y Características. [En línea] 13 de noviembre de 2010. [Citado el: 1 de febrero de 2012.] <http://www.java.softonic.com/>.
39. Recopilación de IDEs y Compiladores. [En línea] [Citado el: 19 de junio de 2013.] <http://www.portalhacker.net/index.php?topic=77442.5;wap2>.
40. pgAdmin. [En línea] [Citado el: 20 de diciembre de 2012.] <http://www.pgadmin.org>.
41. **Valdez, Ing. Alfonso Altamirano.** Comparativo IDEs. Comparativos de Entornos de Desarrollo Integrados. [En línea] [Citado el: 28 de diciembre de 2012.] <http://ubicuos.com/wp-content/uploads/2009/05/comparativoides.pdf>.
42. **Henao, Cristian David Hoyos.** Instalando el Plugin Jboss Tools en Eclipse. [En línea] 22 de abril de 2013. [Citado el: 7 de junio de 2013.] <http://codejavu.blogspot.com/2013/04/instalando-el-plugin-jboss-tools-en.html>.
43. iReport. [En línea] [Citado el: 10 de febrero de 2013.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>.
44. Cómo implantar exitosamente un BPM. [En línea] 4 de junio de 2011. <http://www.sdweb.es/blog/2011/06/04/como-implantar-exitosamente-bpm>.
45. **Pressman, Roger.** Ingeniería de Software, un enfoque práctico. 2001.

Referencias Bibliográficas

46. **Cervantes, Dr. Humberto.** Arquitectura de software. [En línea] [Citado el: 4 de abril de 2013.] <http://www.sg.com.mx/content/view/922>.
47. **Viera, Víctor.** Programación avanzada. [En línea] 1 de junio de 2008. <http://www.emagister.com/curso-programacion-avanzada/reutilizacion-codigo>.
48. Buen Master.com. [En línea] [Citado el: 5 de mayo de 2012.] <http://www.buenmaster.com/?a=536>.
49. **Gustavo.** Introducción al ASP.NET MVC. [En línea] 30 de julio de 2009. <http://cup-coffe.blogspot.com/2010/02/introduccion-al-aspnet-mvc.html> .

Bibliografía

1. ¿Cuántos órganos tiene el cuerpo humano y cuáles son? [En línea] 18 de enero del 2012. <http://www.saberespractico.com/estudios/cultura-general/%C2%BFcuantos-organos-tiene-el-cuerpo-humano-y-cuales-son/>.
2. Apuntes de Anatomía. [En línea] 2013. http://www.iqb.es/cbasicas/anatomia/ab6_01.htm.
3. **Argulo, Iván.** Programador PHP+ASP. [Citado el: 4 de abril del 2013.] <http://www.sgmweb.es/modelo.asp>.
4. **Becerra, Luis; Gonzáles, Fernando; Valenzuela, Joaquín; Cedeño, Marcos.** Impacto de las TICS en la Salud. [En línea] 22 de junio de 2010. <http://www.las-tics-en-el-sector-salud.html>.
5. Caden@gramonte. [En línea] 21 de febrero de 2010 <http://www.cadenagramonte.cu/index.php/articulos/ver/5453>.
Carrillo, Isaías Pérez; Pérez, Rodrigo González; Rodríguez, Aureliano David Martín. Metodología de desarrollo de software. [En línea] 29 de octubre de 2008. http://www.google.com/cu/url?sa=t&rct=j&q=Metodologias+de+desarrollo.pdf&source=web&cd=2&ved=0CDEQFjAB&url=http%3A%2F%2Fsolusoft-g11.googlecode.com%2Ffiles%2FMetodologias%2520de%2520desarrollo.pdf&ei=xSuLUd_9Criz4AOz9IHYDg&usq=AFQjCNFZ9_K-b0YEM-twoVcsjssFhxWgQQ&bvm=bv.46226182,d.dmQ&cad=rja.
6. Cuantos.net. [Citado el: 8 de mayo de 2013.] <http://www.cuantos.net/organos-tiene-el-cuerpo-humano/>.
7. CUBADEBATE. [En línea] 14 de agosto de 2012. <http://www.cubadebate.cu/noticias/2012/08/14/cuba-ha-realizado-mas-de-5-000-trasplantes/>.
8. Eclipse (software). [En línea] [Citado el: 1 de febrero de 2012.] http://www.multilingualarchive.com/ma/enwiki/es/Eclipse_%28software%29.
9. En marcha la segunda fase de CARLA. [En línea] 15 de septiembre de 2011. <http://www.consumer.es/web/es/salud/2011/09/15/203282.php>.
10. **Equipo editorial de Fisterra.** Hemodiálisis. [En línea] 24 de marzo de 2010. <http://www.fisterra.com/salud/3proceDT/hemodialisis.asp>.
11. **Gallardo, Luis Hoyos.** Nuevo sistema de trasplante en el 2009. [En línea] 12 de febrero de 2009.

Referencias Bibliográficas

- http://www.soitu.es/soitu/2009/02/12/info/1234462795_716820.html.
12. Herramientas CASE. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
 13. Historia del lenguaje Java. [En línea] [Citado el: 31 de enero de 2012.] http://www.cad.com.mx/historia_del_lenguaje_java.htm.
 14. **Hoyos, Luis Gallardo**. Cooperación Tecnológica Carla. [En línea] 23 de marzo de 2011. <http://www.transplant-observatory.org/rcidt/Reuniones%20RCIDT/X-Cartagena-de-Indias-Colombia-Marzo-2011/PresentacionITONTCartagenadeIndias.pdf>.
 15. INDRA exportará el modelo español de donación y trasplante. [En línea] 23 septiembre 2011. <http://medicinewsinfo.blogspot.com/2011/09/indra-exportara-el-modelo-espanol-de.html>.
 16. Introduction to OMG's Unified Modeling Language. [En línea] [Citado el: 15 de Diciembre de 2010.] http://www.omg.org/gettingstarted/what_is_uml.htm.
 17. **Juan Sebastián**. Modelo Vista Controlador-Definición y Características. [En línea] 13 de noviembre del 2010. <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.
 18. La Demajagua. [En línea] 19 de febrero de 2010. <http://www.lademajagua.co.cu/index.php/nacionales/6095-cuenta-cuba-con-un-solido-programa-de-trasplante-renal>.
 19. **Lorenci, Miguel**. El sistema CARLA aporta «seguridad y agilidad» en la gestión de trasplantes. [En línea] 2 de agosto de 2010. <http://salud.laverdad.es/piel-y-estetica/cirugia-estetica/1817-el-sistema-carla-aporta-seguridad-y-agilidad-en-la-gestion-de-trasplantes>.
 20. **Martínez, Gustavo Torrico**. Introducción al ASP.NET MVC. [En línea] miércoles 30 julio de 2009 <http://www.cup-coffe.blogspot.com/2010/02/introduccion-al-aspsnet-mvc.html>.
 21. MedlinePlus. [En línea] 21 de agosto de 2011. <http://www.nlm.nih.gov/medlineplus/spanish/ency/article/007434.htm>
 22. MedlinePlus. [En línea] 29 de marzo de 2013. <http://www.nlm.nih.gov/medlineplus/spanish/kidneytransplantation.html>.
 23. NOTAS FARMACOLOGICAS. [Citado el: 8 de mayo de 2013.] <http://www.iqb.es/farmacologia/notas/inmunosupresores.htm>.
 24. **Pajín, Leire**. La aplicación CARLA mejora la comunicación para los trasplantes. [En línea] 14 septiembre 2011. http://movil.hechosdehoy.com/articulo_hh.asp?idarticulo=11528&accion=.

Referencias Bibliográficas

25. Redacción Digital Radio Rebelde. Radio Rebelde. [En línea] 6 de diciembre de 2011. <http://www.radiorebelde.cu/noticia/exitoso-programa-trasplantes-renales-cuba-20111206/>
26. Riñón. [En línea] 2013. <http://www.sidisalta.com.ar/Cuerpo%5Cpages%5CRI%C3%91ON.htm>.
27. Saber es Práctico. [En línea] 18 de enero de 2012. <http://www.saberespractico.com/estudios/cultura-general/%C2%BFcuantos-organos-tiene-el-cuerpo-humano-y-cuales-son/>.
28. **Sánchez, Ana Cecilia Aldana.** Cuadernos de nutrición. [En línea] 2013. <http://www.fitness.com.mx/medicina0109.htm>.
29. Selecciones. [Citado el: 8 de mayo de 2013.] http://mx.selecciones.com/contenido/a2054_cual_es_la_funcion_de_los_rinones.
30. Sistema Nacional de Información de Procuración y Trasplante de la República Argentina. [En línea] [Citado el: 29 de enero de 2013.] <http://sintra.incucai.gov.ar/>.
31. Sistemas fundamentales del ser humano. [En línea] [Citado el: 22 de abril del 2013.] <http://www.profesorenlinea.cl/quinto/5Sistemasfundamentales.htm>.
32. **Soria, Bernat.** Trasplantes. [En línea] 18 de febrero de 2009. <http://www.notas.com.es/tag/trasplantes/>.
33. **Utrera, Migdalia; Peña, Burgos; Delgado, Roberto Carlos; Sánchez, Yoan Gómez.** Invasor.cu. [En línea] 16 de agosto de 2012. <http://www.invasor.cu/index.php/en/culture/15268-mas-de-5-000-trasplantes-renales-en-cuba->.
34. Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0. [En línea] 5 de marzo de 2007. [Citado el: 27 de enero de 2012.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
35. Visual Paradigm. [En línea] [Citado el: 18 de Diciembre de 2010.] <http://www.visual-paradigm.com/news/vpsuite33/vpuml63.jsp>.

Anexos

Preguntas realizadas al especialista en la entrevista:

¿Qué objetivo tiene la especialidad de Nefrología?

¿Cuáles son las principales causas de IRC?

¿Cuáles son los procesos más comunes que se realizan para llevar a cabo un trasplante renal? Descríbalos?

¿Qué parámetros tienen en cuenta para ubicar a un receptor en el listado nacional de pacientes a trasplantar?

¿Cuáles son los parámetros a seguir para realizar la compatibilidad donante-receptor?

Descripción de algunos de los Casos de Uso del Sistema.

Tabla 32. Seleccionar paciente a trasplantar.

Objetivo	Seleccionar paciente a trasplantar.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor accede a la opción Seleccionar paciente a trasplantar, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar paciente a trasplantar deseado, el actor introduce los datos que considera como criterios para buscar el paciente a trasplantar, el sistema busca y muestra los paciente a trasplantar que cumplen con los criterios de búsqueda, el actor selecciona el paciente a trasplantar deseado, el sistema carga en la vista anterior cada uno de los paciente a trasplantar seleccionados, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Alta.	
Precondiciones	Debe estar seleccionado el donante.	
Postcondiciones	Se seleccionó un paciente a trasplantar por el actor.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Seleccionar paciente a trasplantar del Seleccionar donante.	
2.		<p>Muestra los datos del donante:</p> <ul style="list-style-type: none"> • Nombre (s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Muestra un listado de los pacientes a tras-</p>

		<p>plantar.</p> <p>Ordenado teniendo en cuenta criterios establecidos, mostrando la cantidad de elementos configurada para mostrar por página, permitiendo navegar por el resultado.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Seleccionar paciente a trasplantar. • Cancelar operación. Ver Sección 2: “Cancelar operación.” • Insertar resultado Cross_Match. Sección 1: “Insertar resultado Cross_Match”. • Visualizar placa Terasaki. Sección 3: “Visualizar placa Terasaki”.
3.	Selecciona el paciente a trasplantar deseado.	
4.		<p>Carga en la vista anterior todos los datos del paciente a trasplantar seleccionado, mostrando los siguientes a tributos:</p> <ul style="list-style-type: none"> • Nombre (s). • Primer apellido. • Segundo apellido. • Carnet de identidad.
5.		El caso de uso termina.
Flujos alternos		
Sección 1: “Insertar resultado Cross_Match.”		
Flujo básico		

	Actor	Sistema
1.	Selecciona la opción de Insertar resultado Cross_Match.	
2.		<ul style="list-style-type: none"> Permite Insertar resultado Cross_Match. Ver caso de uso: “Insertar resultado Cross_Match”.
3.		El caso de uso termina.
Sección 2. “Cancelar operación”.		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Sección 3. “Visualizar placa Terasaki”.		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de Visualizar placa Terasaki.	
2.		Permite Visualizar placa Terasaki. Ver caso de uso: “Visualizar placa Terasaki”.
3.		El caso de uso termina.
Relaciones	CU Incluidos	No existe
	CU Extendidos	Insertar resultado Cross_Match; ver caso de uso: Insertar resultado Cross_Match. Visualizar placa Terasaki; ver caso de uso: Visualizar placa Terasaki.
Requisitos no funcionales	No existe.	
Asuntos	No existe.	

pendientes

Tabla 33. Seleccionar donante.

Objetivo	Seleccionar donante.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor accede a la opción Seleccionar donante, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el donante deseado, el actor introduce los datos que considera como criterios para buscar el donante, el sistema busca y muestra los donante que cumplen con los criterios de búsqueda, el actor selecciona el donante deseado, el sistema carga en la vista anterior cada uno de los donante seleccionados, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Alta.	
Precondiciones	Debe estar registrado el donante.	
Postcondiciones	Se seleccionó un donante dado criterios.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Seleccionar donante.	
2.		Muestra la vista brindando la posibilidad de

		<p>introducir los criterios elementales de búsqueda:</p> <ul style="list-style-type: none"> • Carnet de identidad. <p>Y permite:</p> <ul style="list-style-type: none"> • Buscar donante dado criterios. • Cancelar operación. Ver Sección 2: “Cancelar operación.” • Seleccionar paciente compatible. Sección 3: Seleccionar paciente compatible. • Seleccionar paciente a trasplantar. Sección 4: Seleccionar paciente a trasplantar. • Buscar resultado de Cross_Match. Sección 5: Buscar resultado de Cross_Match. • Ver datos de Hoja de donante. Sección 6: Ver datos de Hoja de donante.
3.	Introduce los datos que considera como criterios para realizar una búsqueda y selecciona la opción de Buscar donante dado criterios.	
4.		Busca los datos del donante que cumple con los criterios de búsqueda.
5.		Si no se encuentra ningún donante que cumpla con los criterios de búsqueda. Ver Sección 1: “No se encuentra información

		que cumpla con los criterios de búsqueda.”
6.		<p>Muestra los datos del donante que cumple con los criterios de búsqueda, mostrando los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre (s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Ordenados ascendentemente por el Carnet de identidad, mostrando la cantidad de elementos configurados para mostrar por página, permitiendo navegar por el resultado.</p>
7.		<p>Permite:</p> <ul style="list-style-type: none"> • Seleccionar donante.
8.	Selecciona el donante deseado.	
9.		<p>Carga en la vista anterior todos los datos del donante seleccionado.</p> <p>El caso de uso termina.</p>
Flujos alternos		
Sección 1: “No se encuentra información que cumpla con los criterios de búsqueda.”		
Flujo básico		

	Actor	Sistema
1.		Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda.”
2.		Regresa al paso 2 del Flujo Normal de Eventos .
Sección 2: “Cancelar operación”.		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Sección 3: “Seleccionar paciente compatible”.		
	Actor	Sistema
1.	Selecciona la opción Seleccionar paciente compatible.	
2.		Permite Seleccionar paciente compatible. Ver caso de uso: “ Seleccionar paciente compatible ”.
3.		El caso de uso termina.

Sección 4: “Seleccionar paciente a trasplantar”.		
	Actor	Sistema
1.	Selecciona la opción Seleccionar paciente a trasplantar.	
2.		Permite Seleccionar paciente a trasplantar. Ver caso de uso: “ Seleccionar paciente a trasplantar ”.
3.		El caso de uso termina.
Sección 5: “Buscar resultado de Cross_Match”.		
	Actor	Sistema
1.	Selecciona la opción Buscar resultado de Cross_Match.	
2.		Permite Buscar resultado de Cross_Match. Ver caso de uso: “ Buscar resultado de Cross_Match ”.
3.		El caso de uso termina.
Sección 6: “Ver datos de Hoja de donante”.		
	Actor	Sistema
1.	Selecciona la opción Ver datos de Hoja de donante.	
2.		Permite Ver datos de Hoja de donante. Ver

		caso de uso: “ Ver datos de Hoja de donante ”.
3.		El caso de uso termina.
Relaciones	CU Incluidos	No existe
	CU Extendidos	<ul style="list-style-type: none"> • Seleccionar paciente compatible; ver caso de uso: Seleccionar paciente compatible. • Seleccionar paciente a trasplantar; ver caso de uso: Seleccionar paciente a trasplantar. • Buscar resultado de Cross_Match; ver caso de uso: Buscar resultado de Cross_Match. • Ver datos de Hoja de donante; ver caso de uso: Ver datos de Hoja de donante.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 34. Ver datos de Hoja de donante.

Objetivo	Ver datos de Hoja de donante.
Actores	Especialista en trasplante.
Resumen	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción de Ver datos de Hoja de donante, el sistema muestra los datos de la Hoja de donante, el caso de uso termina.
Complejidad	Baja.
Prioridad	Media.
Precondiciones	Para ver los datos de una Hoja de donante, esta debe estar seleccionada.

Postcondiciones	Se vieron los datos de una Hoja de donante.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción de Ver datos de Hoja de donante.	
2.		Muestra los datos de la Hoja de donante. Y permite: <ul style="list-style-type: none"> • Salir de la vista actual.
3.	Selecciona la opción de salir de la vista actual.	
4.		Muestra la vista anterior.
5.		El caso de uso termina.
Relaciones	CU Incluidos	No existe.
	CU Extendidos	No existe.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 35. Buscar donantes.

Objetivo	Buscar donantes.
Actores	Especialista en trasplante.
Resumen	El caso de uso inicia cuando el actor accede a la opción Buscar donantes, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar los donantes, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los donantes que cumplen con los criterios de búsqueda, el caso de uso termina.
Complejidad	Media.

Prioridad	Alta.	
Precondiciones	Debe existir el donante.	
Postcondiciones	Se buscó donantes dado criterios.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Buscar donante.	
2.		<p>Brinda la posibilidad de introducir los criterios elementales de búsqueda:</p> <ul style="list-style-type: none"> • Carnet de Identidad. <p>y permite:</p> <ul style="list-style-type: none"> • Buscar donantes dados criterios. • Modificar Hoja del donante. Ver Sección 1: “Modificar Hoja de donante.” • Ver datos de Hoja de donantes. Ver Sección 3: “Ver datos de Hoja de donante”. • Eliminar Hoja del donante. Ver Sección 5: “Eliminar Hoja de donante”. • Cancelar operación. Ver Sección 4: “Cancelar operación.”
3.	Introduce o selecciona los datos que considera como criterios para realizar una búsqueda y selecciona la opción de Buscar donantes dado criterios.	
4.		Valida los datos. Si hay datos incorrectos, ver

		Sección 6: “Existen datos incorrectos.”
5.		Busca los datos de los donantes que cumplen con los criterios de búsqueda.
6.		Si no se encuentra ningún donante que cumpla con los criterios de búsqueda. Ver Sección 2: “No se encontró información que cumpla con los criterios de búsqueda.”
7.		Muestra un listado de donantes que cumplen con los criterios de búsqueda, mostrando los siguientes atributos: <ul style="list-style-type: none"> • Nombre (s). • Primer apellido. • Segundo apellido. • Carnet de Identidad.
8.		El caso de uso termina.

Flujos alternos

Sección 1: “Modificar Hoja de donante”.

Flujo básico

	Actor	Sistema
1	Selecciona la opción Modificar Hoja de donante.	
2		Permite modificar Hoja de donante. Ver caso de uso: Modificar Hoja de donante.
3		El caso de uso termina.

Sección 2: “No se encontró información que cumpla con los criterios de búsqueda”.

Flujo básico

	Actor	Sistema
1		Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda.”
2		Regresa al paso 2 del Flujo Normal de Eventos .

Sección 3: “Ver datos de Hoja de donante”.

Flujo básico

	Actor	Sistema
1	Selecciona la opción Ver datos de Hoja de donante.	
2		Permite ver los datos de la Hoja de donante. Ver caso de uso: Ver datos de Hoja de donante .
3		El caso de uso termina.

Sección 4: “Cancelar la operación”.

Flujo básico

	Actor	Sistema
1	Selecciona la opción de Cancelar operación.	
2		Regresa a la vista anterior.

Sección 5: “Eliminar Hoja de donante”.

Flujo básico

	Actor	Sistema
1	Selecciona la opción Ver datos de Hoja de donante.	
2		Permite eliminar la hoja de donante. Ver caso de

		uso: Eliminar Hoja de donante.
3		El caso de uso termina.
Sección 5: “Existen datos incorrectos”.		
Acción del Actor		Respuesta del Sistema
1		Muestra un indicador sobre los campos incorrectos.
2		Regresa al paso 3 del Flujo normal de eventos.
Relaciones	CU Incluidos	No existe.
	CU Extendidos	<ul style="list-style-type: none"> • Modificar la Hoja de donante; ver caso de uso: Modificar Hoja de donante. • Eliminar Hoja de donante; ver caso de uso: Eliminar Hoja de donante. • Ver datos de la Hoja de donante; ver caso de uso: Ver datos de Hoja de donante.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 36. Modificar Hoja de donante.

Objetivo	Modificar Hoja de donante.
Actores	Especialista en trasplante.
Resumen	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción Modificar Hoja de donante, el sistema muestra los datos de la Hoja de 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 Hoja de donante, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Alta.	
Precondiciones	Para modificar los datos de una Hoja de donante, esta debe haber sido seleccionada.	
Postcondiciones	Se modificó una Hoja de donante por el actor.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción Modificar Hoja de donante.	
2.		<ul style="list-style-type: none"> • Muestra todos los datos de la Hoja de donante. <p>Y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar las modificaciones. • Cancelar la operación. Ver Sección 3: “Cancelar operación.”
3.	Modifica los datos que necesita.	
4.	Selecciona la opción de aceptar Modificar Hoja de donante.	
5.		Valida los datos. Si hay datos incompletos,

		ver Sección 1: “Existen datos incompletos.”. Si hay datos incorrectos, ver Sección 2: “Existen datos incorrectos.”
6.		Actualiza los datos de Hoja de donante.
7.		Muestra los detalles de la Hoja de donante. Ver caso de uso: Ver detalles de Hoja de donante.
8.		El caso de uso termina.

Flujos alternos

Sección 1: “Existen datos incompletos.”

	Actor	Sistema
1.		Muestra el mensaje de error “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
2.		Regresa al paso 3 del Flujo Normal de Eventos.

Sección 2: “Existen datos incorrectos.”

	Actor	Sistema
1.		Muestra un indicador sobre los campos incorrectos.
2.		Regresa al paso 3 del Flujo Normal de Eventos.

Sección 3: “Cancelar operación.”

Flujo básico

	Actor	Sistema
--	-------	---------

1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Relaciones	CU Incluidos	Ver detalles del área; ver caso de uso: Ver detalles de Hoja de donante.
	CU Extendidos	No existe.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 37. Eliminar Hoja de donante.

Objetivo	Eliminar Hoja de donante.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción Eliminar Hoja de donante, el sistema elimina la Hoja de donante, el caso de uso termina.	
Complejidad	Baja.	
Prioridad	Media.	
Precondiciones	Para eliminar una Hoja de donante, esta debe haber sido seleccionada.	
Postcondiciones	Se eliminó una Hoja de donante por el actor.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción Eliminar Hoja de donante.	

2.		<p>Muestra el mensaje de advertencia “Se eliminará la Hoja de donante seleccionada. Si selecciona Sí se perderán todos los datos. ¿Desea continuar?”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar la eliminación de la Hoja de donante. • Cancelar la operación. Ver Sección 1: “Cancelar operación.”
3.	Selecciona la opción de aceptar la eliminación de la Hoja de donante.	
4.		Ocultar la Hoja de donante.
5.		El caso de uso termina.

Sección 1: “Cancelar operación.”

Flujo básico

	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina

Relaciones	CU Incluidos	No existe.
	CU Extendidos	No existe.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 38. Ver detalles de Hoja de donante.

Objetivo	Ver detalles de Hoja de donante.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción de Ver detalles de Hoja de donante, el sistema muestra los datos de la Hoja de donante, el caso de uso termina.	
Complejidad	Baja.	
Prioridad	Media.	
Precondiciones	Para ver los datos de una Hoja de donante, esta debe estar seleccionada.	
Postcondiciones	Se vieron los datos de una Hoja de donante.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor selecciona una Hoja de donante y accede a la opción de Ver detalles de Hoja de donante.	
2.		Muestra los datos de la Hoja de donante. Y permite: <ul style="list-style-type: none"> • Salir de la vista actual. • Modificar Hoja de donante. Ver Sección 1: " Modificar Hoja de donante". • Eliminar Hoja de donante. Ver Sección 2: "Eliminar Hoja de donante ".
3.	Selecciona la opción de salir de la vista actual.	
4.		Muestra la vista anterior.

5.		El caso de uso termina.
Sección 1: “Modificar Hoja de donante”.		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción de Modificar Hoja de donante.	
1.		Permite Modificar la Hoja de donante. Ver caso de uso: Modificar Hoja de donante.
2.		Regresa al paso 2 del Flujo Normal de Eventos
Sección 2: “Eliminar Hoja de donante”.		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de Eliminar Hoja de donante.	
2.		Permite Eliminar la Hoja de donante. Ver caso de uso: Eliminar Hoja de donante.
3.		Regresa al paso 1 del Flujo Normal de Eventos.
Relaciones	CU Incluidos	No existe
	CU Extendidos	<ul style="list-style-type: none"> • Modificar Hoja de donante; ver caso de uso: Modificar Hoja de donante. • Eliminar Hoja de donante; ver caso de uso: Eliminar Hoja de donante.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 39. Ver detalles de resultado de Cross_Match.

Objetivo	Ver detalles de resultado de Cross_Match.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor selecciona un resultado de Cross_Match y accede a la opción de Ver detalles de resultado de Cross_Match, el sistema muestra los datos del resultado de Cross_Match, el caso de uso termina.	
Complejidad	Baja.	
Prioridad	Media.	
Precondiciones	Para ver los datos de un resultado de Cross_Match, este debe estar seleccionado.	
Poscondiciones	Se vieron los datos de un resultado de Cross_Match.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor selecciona Insertar resultado de Cross_Match y accede a la opción de Ver detalles de resultado de Cross_Match.	
2.		<p>Muestra los datos del donante:</p> <ul style="list-style-type: none"> • Nombre(s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Muestra los datos del paciente:</p> <ul style="list-style-type: none"> • Nombre(s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Muestra el resultado de Cross_Match.</p>

		<p>Permite:</p> <ul style="list-style-type: none"> • Salir de la vista actual. • Modificar resultado de Cross_Match. Ver Sección 1: “Modificar resultado de Cross_Match”. • Eliminar resultado de Cross_Match. Ver Sección 2: “Eliminar resultado de Cross_Match”.
3.	Selecciona la opción de salir de la vista actual.	
4.		Muestra la vista anterior.
5.		El caso de uso termina.

Sección 1: “Modificar resultado de Cross_Match”.

Flujo básico

	Actor	Sistema
1.	Selecciona la opción de Modificar resultado de Cross_Match.	
2.		Permite Modificar el resultado de Cross_Match. Ver caso de uso: Modificar resultado de Cross_Match.
3.		Regresa al paso 1 del Flujo Normal de Eventos.

Sección 1: “Eliminar resultado de Cross_Match”.

Flujo básico

	Actor	Sistema
1.	Selecciona la opción de Eliminar resultado de Cross_Match.	
2.		Permite Eliminar el resultado de Cross_Match. Ver caso de uso: Eliminar resultado de

		Cross_Match.
3.		El caso de uso termina.
Relaciones	CU Incluidos	No existe.
	CU Extendidos	<ul style="list-style-type: none"> • Modificar resultado de Cross_Match; ver caso de uso: Modificar resultado de Cross_Match. • Eliminar resultado de Cross_Match; ver caso de uso: Eliminar resultado de Cross_Match.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 40. Modificar resultado de Cross_Match.

Objetivo	Modificar resultado Cross_Match.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor selecciona un resultado de Cross_Match y accede a la opción Modificar resultado de Cross_Match, el sistema muestra los datos del resultado de Cross_Match 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 resultado de Cross_Match, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Alta.	
Precondiciones	Para modificar los datos de un resultado de Cross_Match, este debe haber sido seleccionado.	
Postcondiciones	Se modificó un resultado de Cross_Match por el actor.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema

1.	El caso de uso inicia cuando el actor selecciona un resultado de Cross_Match y accede a la opción Modificar resultado de Cross_Match.	
2.		<ul style="list-style-type: none"> • Muestra todos los datos del donante, del paciente y del resultado de Cross_Match. Brinda la posibilidad de cambiar el valor del resultado de Cross_Match, seleccionando uno diferente. Y permite: <ul style="list-style-type: none"> • Aceptar las modificaciones. • Cancelar la operación. Ver Sección 1: “Cancelar operación.”
3.	Modifica los datos que necesita.	
4.	Selecciona la opción de aceptar Modificar resultado de Cross_Match.	
5.		Valida los datos. Si hay datos incompletos, ver Sección 2: “Existen datos incompletos.”. Si hay datos incorrectos, ver Sección 3: “Existen datos incorrectos.”
6.		Actualiza los datos del resultado Cross_Match.
7.		Muestra los datos del resultado de Cross_Match. Ver caso de uso: Ver datos de resultado de Cross_Match.
8.		El caso de uso termina.
Sección 1: “Cancelar operación.”		

Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Sección 2. “Existen datos incompletos.”		
	Actor	Sistema
1.		Muestra un indicador sobre los campos incompletos.
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 3: “Existen datos incorrectos”.		
Flujo básico		
	Actor	Sistema
1		Muestra un indicador sobre los campos incorrectos.
2		Regresa al paso 3 del Flujo Normal de Eventos .
Relaciones	CU Incluidos	Ver detalles de resultado de Cross_Match. Ver caso de uso : “Ver detalles de resultado de Cross_Match”.
	CU Extendidos	No existe.
Requisitos no funcionales	No existe.	
Asuntos	No existe.	

pendientes	
------------	--

Tabla 41. Eliminar resultado de Cross_Match.

Objetivo	Eliminar resultado de Cross_Match.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor selecciona un resultado de Cross_Match y accede a la opción Eliminar resultado de Cross_Match, el sistema elimina el resultado de Cross_Match, el caso de uso termina.	
Complejidad	Baja.	
Prioridad	Media.	
Precondiciones	Para eliminar un resultado de Cross_Match, este debe haber sido seleccionado.	
Postcondiciones	Se eliminó un resultado de Cross_Match por el actor.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
	El caso de uso inicia cuando el actor selecciona un resultado de Cross_Match y accede a la opción Eliminar resultado de	
	1. Cross_Match.	
	2.	<p>Muestra el mensaje de advertencia “Se eliminará el resultado de Cross_Match seleccionado. Si selecciona Sí se perderán todos los datos. ¿Desea continuar?”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar la eliminación del resultado de Cross_Match.

		<ul style="list-style-type: none"> • Cancelar la operación. Ver Sección 1: "Cancelar operación."
3.	Selecciona la opción de Aceptar la eliminación de la entidad.	
4.		Ocultar el resultado de Cross_Match.
5.		Regresa a la vista anterior.
6.		El caso de uso termina.
Sección 1: "Cancelar operación."		
Flujo básico		
Actor		Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Relaciones	CU Incluidos	No existe.
	CU Extendidos	No existe.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 42. Buscar resultado de Cross_Match.

Objetivo	Buscar resultado Cross_Match.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor accede a la opción Buscar resultado Cross_Match, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el resultado Cross_Match, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra los resultado Cross_Match que cumplen con los criterios de búsqueda, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Alta.	
Precondiciones	Debe haberse seleccionado un donante.	
Postcondiciones	Se buscó un resultado Cross_Match dado criterios.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede al Seleccionar donante y selecciona la opción Buscar resultado Cross_Match.	
2.		<p>Muestra los datos del donante:</p> <ul style="list-style-type: none"> • Nombre(s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Y brinda la posibilidad de seleccionar:</p> <ul style="list-style-type: none"> • Resultado. <p>y permite:</p> <ul style="list-style-type: none"> • Buscar resultado Cross_Match dados criterios.

		<ul style="list-style-type: none"> • Modificar resultado Cross_Match. Ver Sección 1: “Modificar resultado Cross_Match.” • Eliminar resultado Cross_Match. Ver Sección 4: “Eliminar resultado Cross_Match”. • Cancelar operación. Ver Sección 3: “Cancelar operación.”
3.	Selecciona los datos que considera como criterios para realizar una búsqueda y selecciona la opción de Buscar resultado Cross_Match dado criterios.	
4.		Valida los datos. Si hay datos incorrectos, ver Sección 5: “Existen datos incorrectos”.
5.		Busca los datos de los resultados Cross_Match que cumplan con los criterios de búsqueda.
6.		Si no se encuentra ningún resultado Cross_Match que cumpla con los criterios de búsqueda. Ver Sección 2: “No se encontró información que cumpla con los criterios de búsqueda.”
7.		<p>Muestra un listado con los pacientes que cumplen con los criterios de búsqueda, mostrando los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre(s). • Primer apellido.

		<ul style="list-style-type: none"> • Segundo apellido. • Carnet de identidad. • Resultado.
8.		El caso de uso termina.

Flujos alternos

Sección 1: “Modificar resultado Cross_Match”.

Flujo básico

	Actor	Sistema
1.	Selecciona un resultado Cross_Match.	
2.	Selecciona la opción Modificar resultado Cross_Match.	
3.		Permite modificar el resultado Cross_Match. Ver caso de uso: Modificar resultado Cross_Match .

Sección 2: “No se encontró información que cumpla con los criterios de búsqueda”.

Flujo básico

	Actor	Sistema
1.		Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda.”
2.		Regresa al paso 2 del Flujo Normal de Eventos .

Sección 3: “Cancelar la operación”.

Flujo básico

	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.

Sección 4: “Eliminar Hoja de donante”.		
Flujo básico		
	Actor	Sistema
1.	Selecciona un donante.	
2.	Selecciona la opción Ver datos de Hoja de donante.	
3.		Permite ver los datos de la Hoja de donante. Ver caso de uso: Ver datos de Hoja de donante.
Sección 5: “Existen datos incorrectos”.		
	Actor	Sistema
1.		Muestra un indicador sobre los campos incorrectos.
2.		Regresa al paso 3 del Flujo Normal de Eventos.
Relaciones	CU Incluidos	No existe.
	CU Extendidos	<ul style="list-style-type: none"> • Modificar resultado Cross_Match; ver caso de uso: Modificar resultado Cross_Match. • Eliminar resultado Cross_Match; ver caso de uso: Eliminar resultado Cross_Match.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 43. Listar pacientes trasplantados.

Objetivo	Listar pacientes trasplantados.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor accede a la opción Listar pacientes trasplantados, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar los pacientes trasplantados, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y lista los pacientes trasplantados que cumplen con los criterios de búsqueda, el sistema permite crear, ver, modificar y eliminar los pacientes trasplantados, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Media.	
Precondiciones	Debe de existir un paciente trasplantado.	
Postcondiciones	Se listaron pacientes trasplantados dados criterios.	
Flujo de eventos		
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Listar pacientes trasplantados.	
2.		<p>Muestra un listado de pacientes trasplantados con los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre (s). • Primer apellido. • Segundo apellido. • Carnet de identidad. <p>Ordenado ascendentemente por el Nombre(s), mostrando la cantidad de elementos configurada para mostrar por página, permitiendo navegar por el resultado.</p>

		<p>Si no se encuentra ningún paciente trasplantado. Ver Sección 1: “No se encuentra información.”</p> <p>Permite:</p> <ul style="list-style-type: none"> • Realizar una búsqueda simple. Ver Sección 2: “Realizar una búsqueda simple.” • Cancelar operación. Ver Sección 3: “Cancelar operación.”
3.		El caso de uso termina.

Flujos alternos

Sección 1: “No se encuentra información.”

Flujo básico

	Actor	Sistema
1.		Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda.”
2.		Regresa al paso 2 del Flujo Normal de Eventos .

Alternativa 2: “Realizar una búsqueda simple.”

Flujo básico

	Actor	Sistema
1.		<p>Brinda la posibilidad de introducir los criterios de búsqueda simple:</p> <ul style="list-style-type: none"> • Fecha inicio. • Fecha fin.

2.	Introduce los datos que considera como criterios para realizar una búsqueda simple y selecciona la opción de Buscar pacientes trasplantados.	
3.		Busca los datos de los pacientes trasplantados que cumplen con los criterios de búsqueda.
4.		Regresa al paso 2 del Flujo Normal de Eventos .
Alternativa 4: "Cancelar operación."		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.
Relaciones	CU Incluidos	No existe
	CU Extendidos	No existe
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 44. Generar reporte de cantidad de donantes por centro.

Objetivo	Generar reporte de cantidad de donantes por centro.
Actores	Especialista en trasplante.

Resumen	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de cantidad de donantes por centro, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes genera el reporte de información. El actor selecciona la opción de exportar el reporte, el sistema imprime los datos del reporte, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Media.	
Precondiciones	No existe.	
Postcondiciones	Se generó un reporte de cantidad de donantes por centro dado criterios.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Generar reporte de cantidad de donantes por centro.	
2.		<p>Brinda la posibilidad de introducir los criterios para generar el reporte:</p> <ul style="list-style-type: none"> • Fecha de inicio. • Fecha fin. <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar generar reporte de cantidad donantes por centro. • Cancelar operación. Ver Sección 1: "Cancelar operación."
3.	Introduce o selecciona los datos que considera como criterios para generar el reporte y	

	<p>selecciona la opción de Generar Reporte de cantidad de donantes por centro.</p>	
4.		<p>Valida los datos. Si hay datos incompletos, ver Sección 2: “Existen datos incompletos.”. Si hay datos incorrectos, ver Sección 3: “Existen datos incorrectos.”. Si en el rango de fecha seleccionado, la fecha de inicio tiene un valor mayor a la fecha de fin, ver Sección 4: “Rango de fechas incorrecto”.</p>
5.		<p>Busca los datos de cantidad de donantes por centro que cumplen con los criterios seleccionados, consultando las entidades:</p> <ul style="list-style-type: none"> • Hoja de donante.
6.		<p>Si no se encuentra ninguna información que cumpla con los criterios seleccionados. Ver Sección 5: “No se encuentra información que cumpla con los criterios seleccionados.”</p>
7.		<p>Genera el reporte a partir de los criterios seleccionados, mostrando los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre(s). • Primer apellido. • Segundo apellido. • Centro que extrae. <p>Ordenados ascendentemente por el Primer apellido, mostrando la cantidad de elementos configurados para mostrar por páginas, permitiendo navegar</p>

		por el resultado.
8.		Permite: <ul style="list-style-type: none"> • Exportar el reporte.
9.	Selecciona la opción de Exportar el reporte.	
10.		Exporta los datos del reporte. Ver caso de uso: Exportar .
11.		El caso de uso termina.

Sección 1: “Cancelar operación”.

Flujo básico

	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.

Sección 2: “Existen datos incompletos”.

	Actor	Sistema
1.		Muestra un indicador sobre los campos incompletos.
2.		Regresa al paso 3 del Flujo Normal de Eventos .

Sección 3: “Existen datos incorrectos”.

	Actor	Sistema
1.		Muestra un indicador sobre los campos incorrectos.

2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 4: “Rango de fechas incorrecto”.		
	Actor	Sistema
1.		Muestra el mensaje de error “La fecha de inicio debe tener un valor anterior o igual a la fecha de fin”.
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 5: “No se encuentra información que cumpla con los criterios seleccionados”.		
	Actor	Sistema
1.		Muestra el mensaje de información “No se encuentra información que cumpla con los criterios seleccionados.”
2.		Regresa al paso 2 del Flujo Normal de Eventos .
Relaciones	CU Incluidos	No existe.
	CU Extendidos	Imprime los datos del reporte; ver caso de uso: Exportar .
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Tabla 45. Generar reporte de cantidad de receptores por hospital de procedencia.

Objetivo	Generar reporte de cantidad de receptores por hospital de procedencia.	
Actores	Especialista en trasplante.	
Resumen	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de cantidad de receptores por hospital de procedencia, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes genera el reporte de información. El actor selecciona la opción de exportar el reporte, el sistema imprime los datos del reporte, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Media.	
Precondiciones	No existe.	
Postcondiciones	Se generó un reporte de cantidad de receptores por hospital de procedencia dado criterios.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor accede a la opción Generar reporte de cantidad de receptores por hospital de procedencia.	
2.		<p>Brinda la posibilidad de introducir los criterios para generar el reporte:</p> <ul style="list-style-type: none"> • Fecha de inicio. • Fecha fin. <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar generar reporte de cantidad de receptores por hospital de procedencia.

		<ul style="list-style-type: none"> • Cancelar operación. Ver Sección 1: “Cancelar operación”.
3.	Introduce o selecciona los datos que considera como criterios para generar el reporte y selecciona la opción de Generar Reporte de cantidad de receptores por hospital de procedencia.	
4.		<p>Valida los datos. Si hay datos incompletos, ver Sección 2: “Existen datos incompletos.”. Si hay datos incorrectos, ver Sección 3: “Existen datos incorrectos.”. Si en el rango de fecha seleccionado, la fecha de inicio tiene un valor mayor a la fecha de fin, ver Sección 4: “Rango de fechas incorrecto”.</p>
5.		<p>Busca los datos de cantidad de donantes por hospital de procedencia que cumplen con los criterios seleccionados, consultando las entidades:</p> <ul style="list-style-type: none"> • Hoja de donante.
6.		<p>Si no se encuentra ninguna información que cumpla con los criterios seleccionados. Ver Sección 5: “No se encuentra información que cumpla con los criterios seleccionados.”</p>
7.		<p>Genera el reporte a partir de los criterios seleccionados, mostrando los siguientes atributos:</p> <ul style="list-style-type: none"> • Nombre(s).

		<ul style="list-style-type: none"> • Primer apellido. • Segundo apellido. • Centro que dona. <p>Ordenados ascendentemente por el Primer apellido, mostrando la cantidad de elementos configurados para mostrar por páginas, permitiendo navegar por el resultado.</p>
8.		<p>Permite:</p> <ul style="list-style-type: none"> • Exporta el reporte.
9.	Selecciona la opción de Exportar el reporte.	
10.		Imprime los datos del reporte. Ver caso de uso: Exportar .
11.		El caso de uso termina.

Sección 1: “Cancelar operación”.

Flujo básico

	Actor	Sistema
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		El caso de uso termina.

Sección 2: “Existen datos incompletos”.

	Actor	Sistema
--	-------	---------

1.		Muestra un indicador sobre los campos incompletos.
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 3: “Existen datos incorrectos”.		
	Actor	Sistema
1.		Muestra un indicador sobre los campos incorrectos.
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 4: “Rango de fechas incorrecto”.		
	Actor	Sistema
1.		Muestra el mensaje de error “La fecha de inicio debe tener un valor anterior o igual a la fecha de fin”.
2.		Regresa al paso 3 del Flujo Normal de Eventos .
Sección 5: “No se encuentra información que cumpla con los criterios seleccionados”.		
	Actor	Sistema
1.		Muestra el mensaje de información “No se encuentra información que cumpla con los criterios seleccionados.”
2.		Regresa al paso 2 del Flujo Normal de Eventos .
Relaciones	CU Incluidos	No existe.

	CU Extendidos	No existe.
Requisitos no funcionales	No existe.	
Asuntos pendientes	No existe.	

Diagramas de clases del diseño.

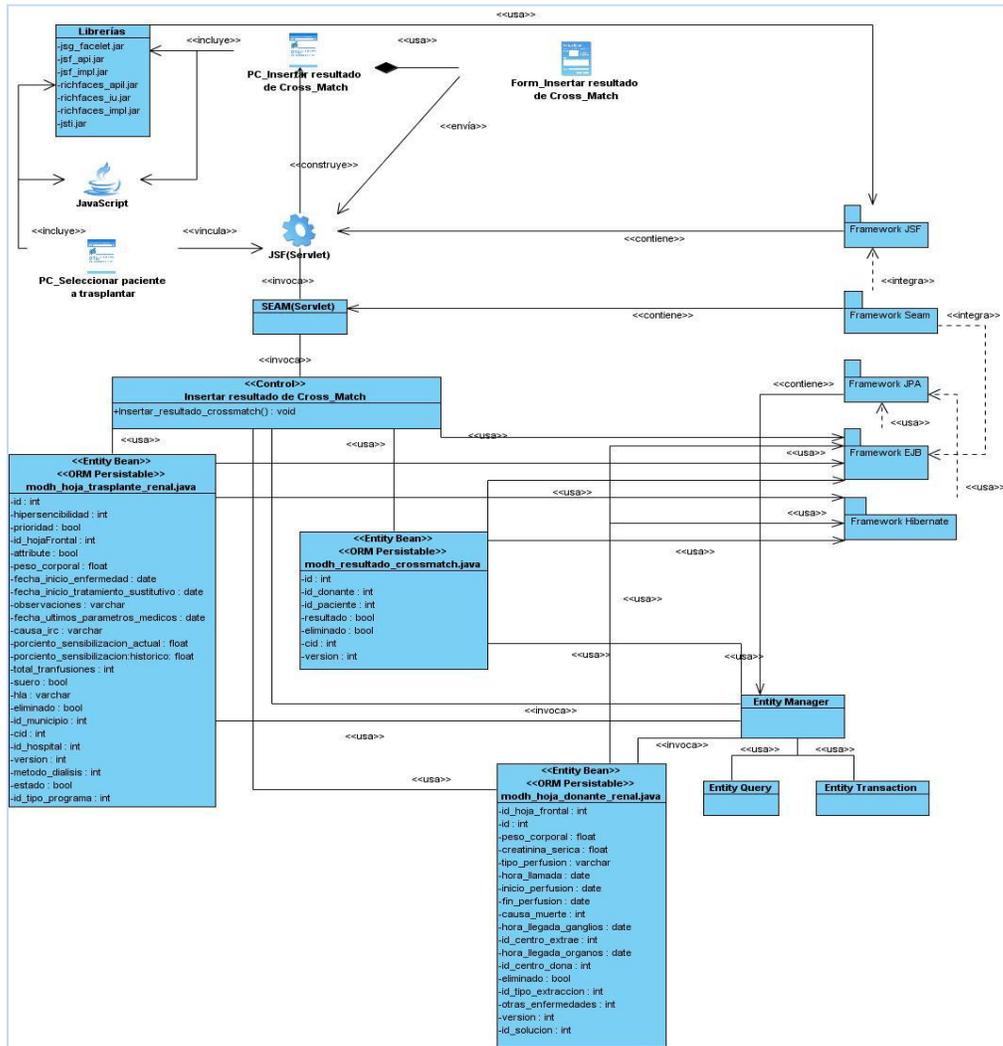


Figura 15. Insertar resultado de Cross_Match.

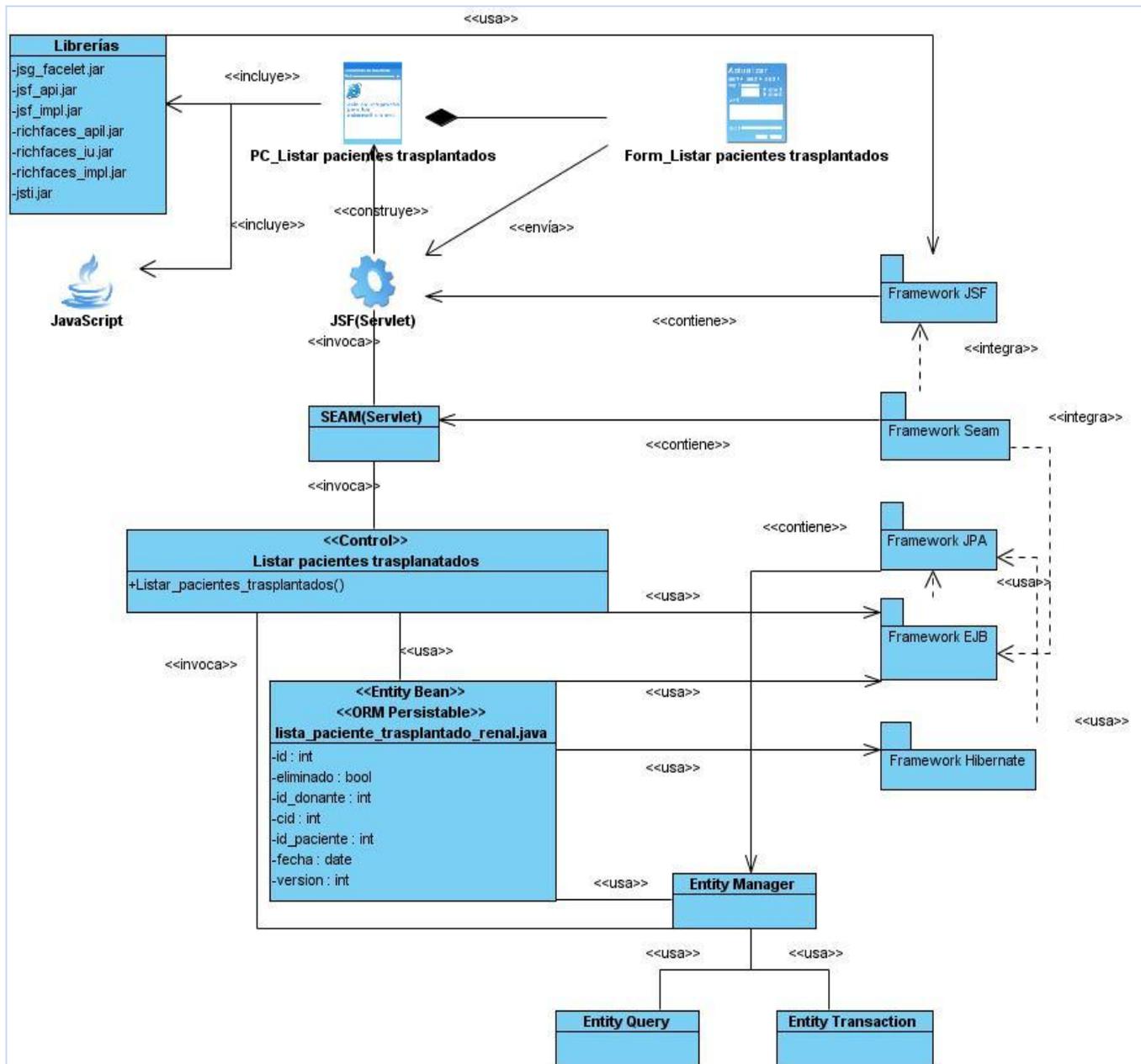


Figura 16. Listar pacientes trasplantados.

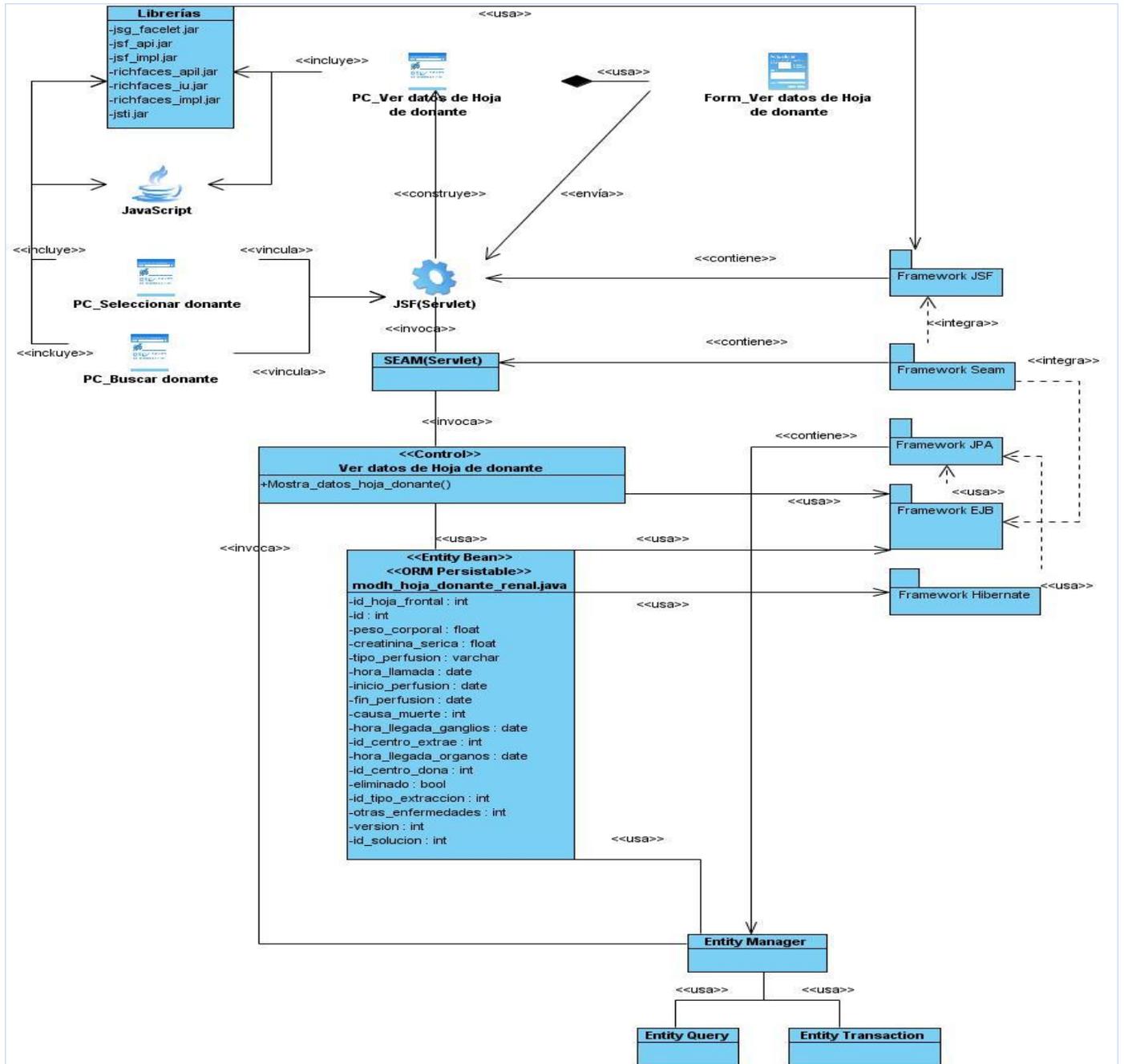


Figura 17. Ver datos de Hoja de donante.

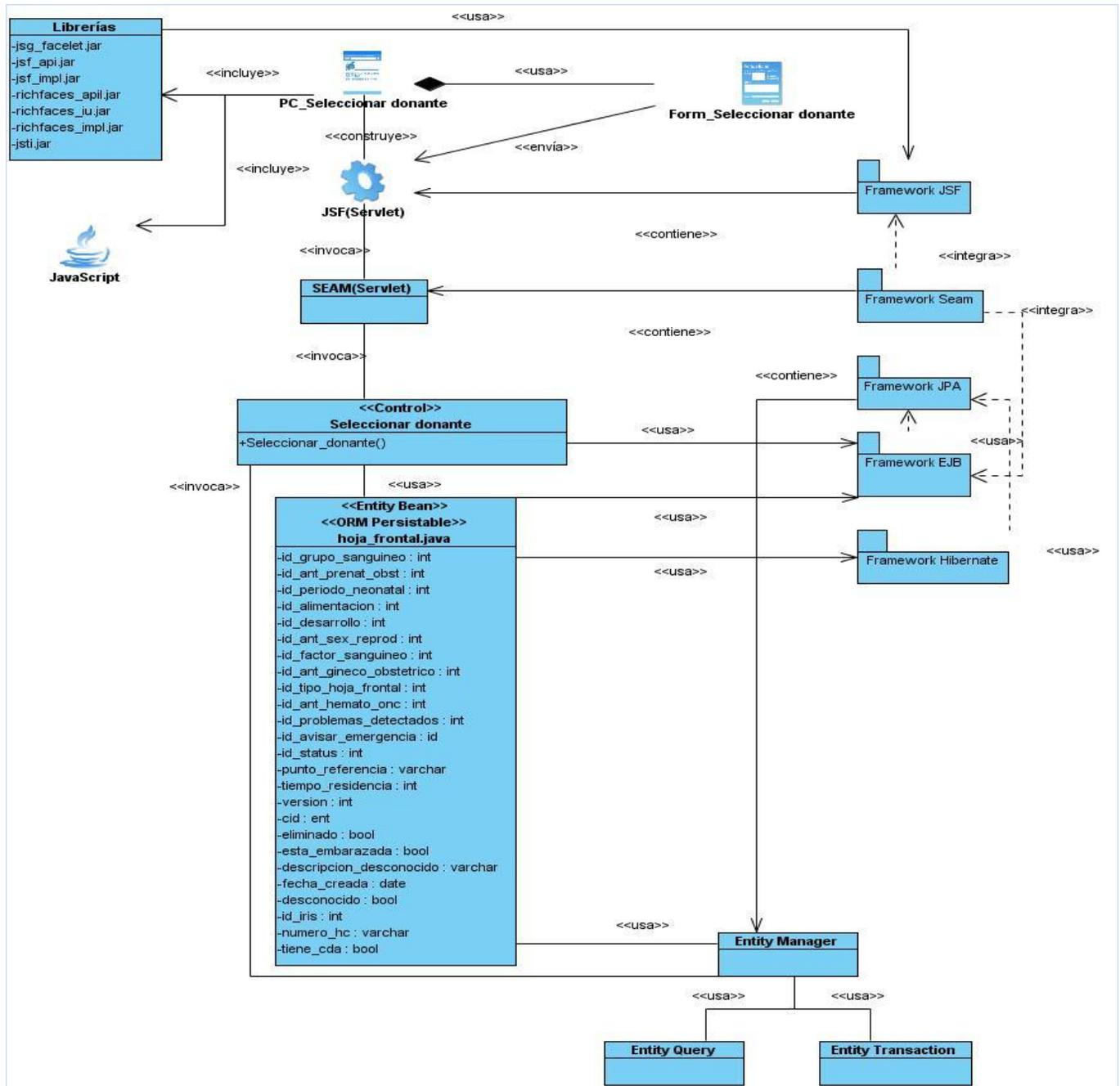


Figura 18. Seleccionar donante.

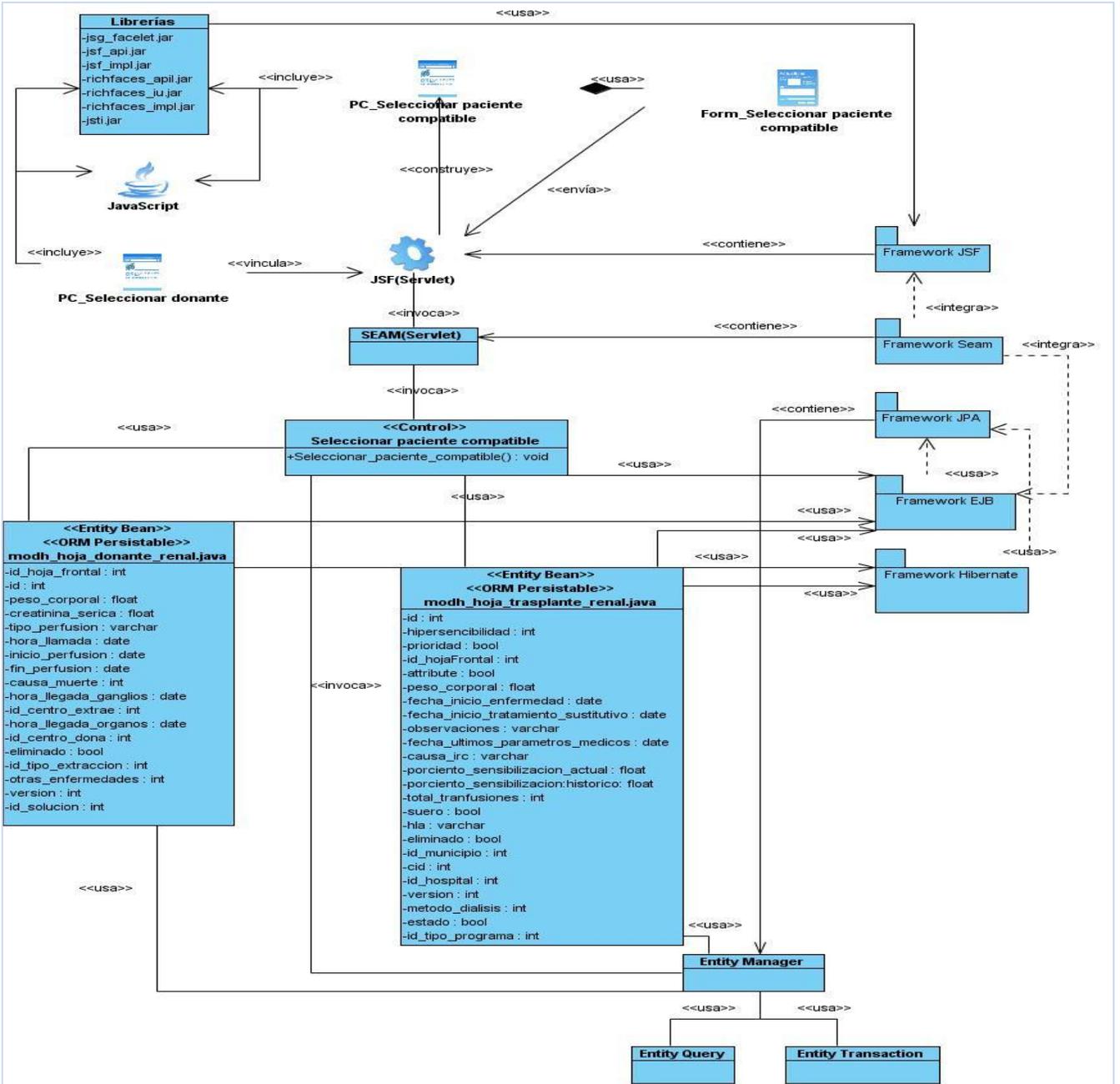


Figura 19. Seleccionar paciente compatible.

Glosario de términos

Deficiencia: Pérdida o anomalía de una estructura o función psicológica, fisiológica o anatómica del individuo (Nivel de órganos).

Trasplante: Es el proceso de quitar un órgano deficiente en el cuerpo del ser humano por uno en mejores condiciones.

Cuerpo del ser humano: Es la estructura física y material del hombre, estructurado por cabeza cuerpo y extremidades.

Riñones: Los riñones son considerados los principales químicos del organismo. Son dos glándulas de color rojo oscuro colocadas simétricamente a los lados de la columna vertebral, en la región lumbar. Miden 10 cm de largo y pesan unos 150 gr cada uno.

Muerte encefálica: (1) Es cuando alguna persona muere de muerte craneal. (2) Cese irreversible de las funciones del encéfalo, o sea, de los hemisferios cerebrales, el tallo encefálico y el cerebelo.

Diálisis peritoneal: Filtra la sangre para eliminar los desechos peligrosos del cuerpo y el exceso de sal y agua. Usa la membrana que recubre el abdomen, llamada membrana peritoneal, para filtrar la sangre.

Hemodiálisis: Filtra la sangre para eliminar los desechos peligrosos del cuerpo y el exceso de sal y agua. La hemodiálisis se logra con un aparato.

Donante: Es aquella persona que dona algún órgano para otra persona.

Donante Cadavérico: Es aquella persona que haya sufrido de muerte encefálica y sea un donante.

Insuficiencia: (1) Escasez o falta de la cantidad que se necesita de una cosa. (2) Incapacidad o inadecuación de una persona para desempeñar un trabajo: reconoció su insuficiencia en este tema y presentó su dimisión. (3) Incapacidad de un órgano para realizar adecuadamente sus funciones.

Órgano: Parte diferenciada del cuerpo que participa en la realización de una función.

Cadáver: Es el cuerpo de un ser humano muerto.

Receptor: Es aquella persona que está en espera de un donante para ser trasplantado.

Emisor: Es aquella persona que dona un órgano.

Sistema de trasplante: Investigación para automatizar la gestión de la información generada durante el proceso de trasplante de cualquier órgano.

Trasplante de riñón: Es una intervención quirúrgica, que se realiza en un centro específicamente autorizado, por un equipo con experiencia, mediante el cual se implanta un riñón de otra persona en su cuerpo.

Insuficiencia renal crónica (IRC): Es una insuficiencia total o casi total en el funcionamiento del riñón para excretar los desechos, concentrar la orina y regular las electrólisis. También se conoce como enfermedad renal en etapa Terminal y se presenta cuando los riñones ya no pueden funcionar al nivel necesario para la vida diaria, es decir que la insuficiencia renal crónica progresa a tal punto en que la función de los riñones es menos del 10% de su capacidad normal.

Antígenos Leucocitarios Humanos (HLA): (1) Es un examen de sangre para buscar una proteína específica que se encuentra en la superficie de los glóbulos blancos. (2) Son proteínas que ayudan al sistema inmunitario del cuerpo a diferenciar entre sus propias células y sustancias extrañas y dañinas.

Hipersensibilidad: Son personas que ya han sido trasplantadas y necesitan de un nuevo implante o han recibido muchas transfusiones que les han generado esa mayor susceptibilidad al rechazo.

Compatibilidad: Es cuando un donante comparte con el receptor la mayor cantidad de antígenos que pueden generar una respuesta inmune particularmente fuerte, como el complejo mayor de histocompatibilidad (también llamado sistema HLA en humanos) y las glicoproteínas que determinan el grupo sanguíneo.

Cross-Match: Es una prueba que se le realiza a los receptores para determinar si son compatibles con los donantes.

Fármacos inmunosupresores: Medicamentos que contribuyen a evitar el rechazo del órgano trasplantado.

Placa Terasaki: Consiste en la incubación de una pequeña cantidad de suero con linfocitos en pequeños orificios.

Transfusión de sangre: Es un procedimiento médico relativamente sencillo durante el cual un paciente recibe sangre o algún componente de la sangre a través de una vía intravenosa. Esta vía es un tubo muy fino que se introduce en la vena con una pequeña aguja.

Hoja de datos del receptor: Se recogen los datos personales del receptor y los análisis que se le han realizado.

Hoja de donante: Recoge los datos personales del donante y los exámenes que se le han realizado.

Caso de Uso: Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Requisito: (1) Una condición o capacidad necesitada por el usuario para resolver un problema o lograr un objetivo. (2) Una condición o capacidad que debe poseer un producto o componente de producto para satisfacer un contrato, estándar, especificación, u otros documentos obligatorios formales. (3) Una representación documentada de una condición o capacidad como en (1) o (2).

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

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

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

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

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

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

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

LPT: Puerto paralelo para conectar periféricos a una computadora.

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

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

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

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

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

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

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

Plataforma: Sistema que sirve como base para hacer funcionar determinados módulos de hardware o de

software con los que es compatible.

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

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

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

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