

**Universidad de las Ciencias Informáticas**  
**Facultad 7**



**Título: Módulo Admisión del Sistema de  
Información Hospitalaria alas HIS**

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

**Autores:** Rodney Ledo Ramírez  
Alberto Ortega Palacios

**Tutor:** Ing. Pedro Ernesto Salas Oliva

Ciudad de La Habana, junio del 2009  
“Año del 50 aniversario del triunfo de la Revolución”

## Declaración de autoría

---

### DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 26 días del mes de junio del año 2009.

**Rodney Ledo Ramírez**

\_\_\_\_\_  
Firma del Autor

**Alberto Ortega Palacios**

\_\_\_\_\_  
Firma del Autor

**Ing. Pedro Ernesto Salas Oliva**

\_\_\_\_\_  
Firma del Tutor

## Datos de contacto

---

### DATOS DE CONTACTO

Tutor: Ing. Pedro Ernesto Salas Oliva

Instructor recién graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor Facultad # 7. Ha impartido las asignaturas Probabilidad y Estadística, Investigación de Operaciones y Algebra Lineal. Forma parte del proyecto Gestión Hospitalaria y actualmente ocupa el roll de analista del módulo Admisión.

Correo electrónico: [psalas@uci.cu](mailto:psalas@uci.cu)

# Agradecimientos

---

## AGRADECIMIENTOS

### **De Rodney**

Quiero agradecer especialmente a mis padres por apoyarme en todo momento a lo largo de mi vida, guiarme hacia el camino correcto, ofrecerme amor, educación, confianza y dedicación. Sin ellos, no hubiese podido alcanzar esta meta.

A toda mi familia que siempre me ha impulsado a ser mejor persona cada día.

A mi tutor y a los profesores del proyecto porque su ayuda me ha servido de mucho.

A todos mis amigos y compañeros que siempre han estado a mi lado en los buenos y malos momentos.

A Lixandra por su comprensión y apoyo incondicional.

### **De Alberto:**

A mi familia, y en especial a mi mamá y mi hermanita que son todo para mí, y hacen que me esfuerce más cada día.

A todos aquellos amigos, que me han acompañado estos 5 años, que se gradúan junto a mí, con los que compartí grandes momentos de mi vida. Nunca los olvidaré.

A mis compañeros de aula, con los que di los primeros pasos, siendo testigos de los momentos que pasamos juntos en esta universidad, los cuales me aguantaron todo este tiempo, tarea que no fue nada fácil.

A mis compañeros de proyecto y tutor por su apoyo y preocupación.

A mis profesores por haberme formado como un profesional y hombre de bien.

A todos los que aportaron su granito de arena para que hoy pueda hacer mi sueño realidad.

# Dedicatoria

---

## DEDICATORIA

### **De Rodney:**

A mis queridos padres que tanto se han esforzado para que este día se haga realidad. Sé que hoy ven cumplido su mayor sueño.

A mi hermano, para el que he tratado de ser un ejemplo a seguir.

A toda mi familia y en general a las personas que me han apoyado a lo largo de estos años y han contribuido a mi formación humana y profesional.

### **De Alberto:**

A mi mamá que me ha dado todo y siempre me ha apoyado, siendo madre y padre al mismo tiempo, guiándome y aconsejándome en momentos esenciales de mi vida, conduciéndome a tomar siempre el camino correcto.

A mi papá, que aunque pude estar poco tiempo con él, me enseñó muchas cosas que me han hecho triunfar en distintos campos, y sé que en estos momentos estuviera orgulloso de que haya llegado hasta aquí. Donde quiera que estés, gracias.

A mis hermanitas lindas, Arlyn y Yadira por estar siempre a mi lado, y apoyarme en todo.

A mi familia, por darme sus consejos en los momentos necesarios.

A mis amigos: Yusniel, Yusliel, Jorgito, Ray, Yosmel entre otros, y compañeros de aula que anduvieron junto a mí, soportándome todos estos años.

A mi compañero de tesis Rodney, por su apoyo y por soportarme durante todo este curso.

Muchas gracias a todos.

# Resumen

---

## RESUMEN

Las instituciones hospitalarias dividen sus áreas operativas en diversos servicios para lograr una mejor organización de los flujos de trabajo de la entidad. Entre ellos se encuentra el servicio de admisión, cuyo principal objetivo es gestionar los ingresos, egresos, traslados, la historia clínica del paciente y emitir reportes relacionados con la población de las salas y la coordinación entre las camas.

En la mayoría de los casos, procesar toda la información generada en dicha área consume mucho tiempo y esfuerzo, pero además está expuesta a la ocurrencia de errores debido a que todas las actividades se desarrollan manualmente y la información es mantenida en papel. Es por ello que se propone la implementación del módulo Admisión del Sistema de Información Hospitalario alas HIS para facilitar la gestión de información relacionada con los procesos en el área de Admisión de las instituciones hospitalarias.

El desarrollo del sistema está guiado por el Proceso Unificado de Desarrollo y se basa en tecnologías libres, multiplataforma y sobre una arquitectura en capas. Se utiliza Java como lenguaje de programación y se implementa el patrón de arquitectura Modelo Vista Controlador.

Una vez logrado el sistema se espera optimizar los procesos que se desarrollan en el área de admisión, lograr una mejor organización de la información y agilizar la prestación de servicios a los pacientes. También se brindarán una serie de reportes que favorecerán la toma de decisiones a nivel médico y administrativo.

## PALABRAS CLAVE

Gestión de información, admisión, historia clínica, ingresos, egresos, reportes, traslados

# Tabla de contenidos

---

## TABLA DE CONTENIDOS

INTRODUCCIÓN .....	1
CAPITULO 1. FUNDAMENTACIÓN TEÓRICA.....	1
1.1    Conceptos básicos relacionados con el dominio del problema (2).....	1
1.1.1    Inscripción .....	1
1.1.2    Admisión .....	1
1.1.3    Historia clínica.....	1
1.1.4    Ingreso .....	2
1.1.5    Ingreso fuera de servicio .....	2
1.1.6    Sala.....	2
1.1.7    Cama.....	2
1.1.8    Servicio .....	2
1.1.9    Egreso.....	3
1.2    Sistemas automatizados existentes vinculados al campo de acción.....	3
1.2.1    Hosix-V .....	3
1.2.2    Care2x .....	3
1.2.3    C-DAC's Sushrut .....	4
1.2.4    Galenhos(R) .....	5
1.2.5    X-HIS .....	5
1.2.6    SIGHO .....	6
1.2.7    Galen Hospital .....	6
1.3    Tendencias y tecnologías actuales.....	7
1.3.1    Navegadores Web.....	7
1.3.2    Aplicaciones web.....	7
1.3.3    Arquitectura cliente-servidor.....	8
1.3.4    Arquitectura en capas.....	8
1.3.5    Tecnologías utilizadas en el proceso de desarrollo .....	9
1.3.5.1    Java.....	9
1.3.5.2    Capa de presentación .....	10
1.3.5.2.1    Java Server Faces (JSF) .....	10
1.3.5.2.2    RichFaces 3.2 .....	10
1.3.5.3    Capa de negocio .....	11
1.3.5.3.1    JBoss Seam 2.0 .....	11
1.3.5.3.2    JBoss Server 4.2 .....	11
1.3.5.3.3    JBoss tools.....	12
1.3.5.4    Capa de datos.....	12
1.3.5.4.1    Hibernate.....	12
1.3.5.4.2    PostgreSQL 8.3.....	13
1.3.5.5    Interfaces de comunicación.....	13
1.3.5.5.1    Servicios web.....	13

# Tabla de contenidos

---

1.3.5.5.2	XML.....	14
1.3.5.5.3	HL7 .....	14
1.3.5.6	Procedo Unificado de Desarrollo .....	15
1.3.5.6.1	Notación de Modelado de Procesos del Negocio.....	15
1.3.5.6.2	UML (Lenguaje Unificado de Modelado).....	16
1.3.5.7	Herramientas.....	17
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA.....		18
2.1	Flujo actual de los procesos involucrados en el campo de acción .....	18
2.2	Objeto de automatización.....	19
2.3	Información que se maneja .....	20
2.4	Modelo de negocio.....	21
2.4.1	Actores de procesos de negocio. ....	21
2.4.2	Realizar admisión.....	22
2.4.4	Procesar movimiento hospitalario .....	26
2.5	Propuesta del sistema .....	29
2.5.1	Especificación de los requerimientos de software .....	29
2.5.1.1	Requerimientos funcionales .....	29
2.5.1.2	Requerimientos no funcionales.....	31
2.5.1.2.1	Usabilidad .....	31
2.5.1.2.2	Seguridad .....	31
2.5.1.2.3	Rendimiento.....	32
2.5.1.2.4	Soporte .....	32
2.5.1.2.5	Hardware .....	33
2.5.1.2.5.1	Estaciones de trabajo .....	33
2.5.1.2.5.2	Servidores .....	33
2.5.1.2.6	Software.....	33
2.5.1.2.7	Restricciones de diseño .....	34
2.5.1.2.8	Requisitos para la documentación de usuarios en línea y ayuda del sistema. ....	34
2.5.1.2.9	Interfaz.....	34
2.5.1.2.9.1	Interfaces de usuario .....	34
2.5.1.2.9.2	Interfaces de comunicación .....	35
2.5.1.2.10	Portabilidad.....	35
2.5.2	Modelo de casos de uso del sistema.....	35
2.5.2.1	Definición de actores .....	35
2.5.2.2	Vista global de actores.....	36
2.5.2.3	Diagrama de casos de uso .....	37
2.5.2.4	Descripción textual de los casos de uso.....	38
CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA .....		44
3.1	Descripción de la arquitectura, fundamentación .....	44
3.1.1	Patrones de arquitectura .....	44
3.2	Diseño .....	46



# Tabla de contenidos

---

3.2.1	<i>Patrones de diseño</i> .....	47
3.2.2	<i>Patrones en la capa de datos</i> .....	48
3.2.3	<i>Estructuración</i> .....	49
3.2.4	<i>Diagramas de clases</i> .....	50
3.2.5	<i>Diagramas de interacción</i> .....	54
3.2.6	<i>Descripción de las clases</i> .....	58
CAPÍTULO 4 IMPLEMENTACIÓN .....		62
4.1	<i>Modelo de datos</i> .....	62
4.1.1	<i>Descripción de las tablas</i> .....	64
4.2	<i>Implementación</i> .....	67
4.2.1	<i>Diagrama de despliegue</i> .....	67
4.2.2	<i>Diagrama de componentes</i> .....	68
4.3	<i>Tratamiento de errores</i> .....	70
4.4	<i>Estrategias de codificación. Estándares y estilos a utilizar</i> .....	70
CONCLUSIONES GENERALES.....		73
RECOMENDACIONES .....		74
REFERENCIAS BIBLIOGRÁFICAS.....		75
BIBLIOGRAFÍA .....		77
GLOSARIO DE TÉRMINOS .....		80

# Introducción

---

## INTRODUCCIÓN

En los últimos años, las Tecnologías de la Información y las Comunicaciones (TIC) han impactado todas las ramas de la sociedad, incluidas las costumbres y maneras de hacer las diferentes actividades. El desarrollo vertiginoso y gran aceptación de estas tecnologías, se debe a los beneficios que brindan. Entre ellos, la automatización de tareas, que permite agilizar considerablemente los procesos y actividades que en una institución se realizan. Lo que permite reducir costos e incrementar su eficiencia. La incorporación de las TIC en el ámbito institucional puede resultar un elemento clave para mejorar la competitividad e impulsar el crecimiento económico. Una de las esferas donde se han aplicado estos beneficios es en la salud.

En la medicina, los avances de la ciencia han sido aplicados en numerosos campos, pero no se trata solo de la actualización tecnológica por la adquisición de equipamientos modernos. Es indispensable contar con una herramienta que permita recoger y tratar la información. De modo que sea útil para la toma de decisiones, y diseñada para manejar los aspectos administrativos y clínicos de un hospital. Brindar servicios de alta calidad para el cuidado de la salud es una actividad dependiente del control sobre los procesos por los cuales se obtiene la información.

El tratamiento y la obtención de información clara, precisa, confiable y oportuna para el planeamiento, la organización y la toma de decisiones, resultan en este contexto un factor estratégico de éxito. Es por esto que para desarrollar un nuevo modelo de atención, que tenga como centro las necesidades del paciente y cumpla los requisitos esenciales de calidad en la prestación de servicios, es de gran apoyo tener en cuenta las ventajas que ofrecen las TIC.

Las aplicaciones que resuelven estos problemas son conocidas como Sistemas de Información Hospitalario (más conocidas por sus siglas en inglés HIS). Estas tienen como propósito almacenar, recuperar, procesar, reinterpretar datos médico-administrativos de cualquier institución hospitalaria. Además de permitir la optimización de los recursos humanos y materiales para satisfacer las necesidades de generación de información de las áreas operativas, administrativas, clínicas y de investigación en las organizaciones de salud, y por tanto minimizar los inconvenientes burocráticos que enfrentan los pacientes. Todo sistema de información hospitalaria genera reportes e informes dependiendo del área o

# Introducción

---

servicio para el cual se requiera, dando lugar a la retroalimentación de la calidad de la atención de los servicios de salud.

Durante las décadas de los 50 y 60 del siglo XX, los sistemas de información se introdujeron casi exclusivamente por necesidades financieras y de gestión económica de los centros. Un sistema de información hospitalario típico de estos años, tenía poco de sistema clínico, ya que el énfasis estaba puesto en la contabilidad, nóminas y recursos humanos.

En la década de los 70 comenzaron a dejarse sentir con más fuerza las necesidades clínicas, pero principalmente como un producto de las necesidades financieras. Estas cuestiones financieras se centraban en maximizar los ingresos y capturar los costes, y los sistemas de atención al paciente eran capaces de servir como vehículo para documentar las órdenes y peticiones, e indirectamente, sus costos. Los hospitales estaban ya experimentando una necesidad creciente de datos de servicios específicos y del hospital en conjunto.

La década de los 80 impulsó un cambio dramático. Se comenzó a presionar para mejorar la coordinación entre diferentes servicios. Los servicios clínicos comenzaron a recibir mayor presión para mejorar la productividad. En los 80 un sistema de información hospitalario estaba compuesto por un sistema de contabilidad y un sistema de atención al paciente. A partir de la década de los 90, los sistemas se mueven en la dirección de centrarse sobre el paciente, estar orientados clínicamente y apoyar la toma de decisiones médicas teniendo en cuenta que el proceso de producción principal de un hospital es la asistencia al paciente, no el proceso financiero. De aquí, que el núcleo de la arquitectura de los sistemas de información son los procesos clínicos.

La forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado, es lo que se conoce como Sistema Nacional de Salud (SNS). La Organización Mundial de la Salud (OMS) lo define como: «Un complejo de elementos interrelacionados que contribuyen a la salud en los hogares, los lugares de trabajo, los lugares públicos y las comunidades, así como el medio ambiente físico y psicosocial en el sector de salud y otros sectores afines» (1).

# Introducción

---

Además, es el conjunto de unidades administrativas, de producción, investigación y servicios, responsabilizados con la atención integral de la salud de una población. Los sistemas sanitarios de los distintos países del mundo, tienen como misión fundamental, establecer la rectoría del sistema público de salud a través del diseño, ejecución y supervisión de políticas y estrategias que contribuyan al fortalecimiento e integración de los diversos entes que prestan el servicio, con la finalidad de mejorar la calidad de vida y salud de la población.

De acuerdo con la complejidad de las acciones preventivas, curativas y de rehabilitación, así como la especialización de los servicios de salud brindados, los diferentes niveles de atención médica se han organizado en:

- **Atención primaria de salud:** dirige sus acciones a la promoción y protección de la salud.
- **Atención médica secundaria:** su función fundamental es tratar al hombre ya enfermo.
- **Atención médica terciaria:** brinda servicios especializados y de alta complejidad.

Las instituciones hospitalarias como parte de la atención médica secundaria, dividen sus áreas operativas en diversos servicios, para lograr una mejor organización de los flujos de trabajo y un mejor funcionamiento de la entidad. El primer punto de contacto con los pacientes que acuden a una unidad de salud a recibir atención médica es a través del servicio de Admisión. El principal objetivo de este servicio es registrar la admisión de pacientes en algunas áreas de la unidad por cualquiera de las vías por las que pudieran llegar, a través de la persona encargada en la recepción del hospital, crear su historia clínica o consultarla en caso de que exista; y los traslados internos, externos y el alta del paciente. Este servicio, es el responsable de mantener actualizada la información referente a la población de las salas, y la coordinación entre las camas de los pacientes ingresados en el centro médico.

Cumplir con estas tareas se torna muy difícil y complejo, por el significativo volumen de información que en estas instituciones se genera diariamente. En la mayoría de los casos, esta se encuentra dispersa o no está disponible en tiempo y forma necesarios, debido a que es mantenida en papel. Excepto en casos aislados que utilizan hojas electrónicas de cálculo o algún sistema adquirido por la institución, que no resuelven las necesidades que en materia de gestión de la información se tienen para el apoyo a la toma de decisiones.

# Introducción

---

Por todo lo anterior, en algunos casos se cometen errores, durante el llenado de planillas y la obtención de datos, tales como las estadísticas relacionadas con el movimiento hospitalario. Porque es muy complicado extraer esa información y realizar los cálculos buscando en cada uno de los registros. Un error muy común es duplicar la historia clínica del paciente, posibilitando la existencia de información repetida o redundante que ocupan grandes espacios de almacenamiento. Por ende, es más propensa al deterioro, la pérdida y el retraso en los procesos que conduce a una demora en la prestación de servicios. Así como, a la toma de decisiones equivocadas y la carencia de información estadísticas en tiempo real de las distintas áreas, servicios y departamentos.

En este sentido, se determina que el **Problema a resolver** es ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de Admisión de las instituciones hospitalarias?

El **Objeto de estudio** del presente trabajo se refiere al Proceso de gestión de información en las instituciones hospitalarias. Mientras que el **Campo de acción** se enmarca en el Proceso de gestión de información en el área de Admisión en las instituciones hospitalarias.

Para resolver el problema identificado se propone el siguiente **Objetivo general**: Desarrollar el módulo Admisión del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **Tareas a desarrollar**:

1. Analizar los procesos de negocio asociados al área de Admisión de las instituciones hospitalarias.
2. Asimilar la arquitectura definida por el Área Temática de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
3. Obtener mediante el Proceso Unificado de Desarrollo, los flujos de trabajo de “Modelo de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
4. Implementar los procesos de negocio relacionados con realizar admisión y los reportes del módulo de Admisión.

En este sentido se puede destacar que el desarrollo del módulo de Admisión del Sistema de Información Hospitalaria, proporcionará un grupo de beneficios entre los que pueden ser mencionados los siguientes:

# Introducción

---

1. Historia Clínica Electrónica única por paciente y su seguimiento
2. Seguridad y confidencialidad de toda la información médica del paciente.
3. Poner a disposición del profesional médico herramientas de gestión clínica y gestión administrativa que den respuesta a sus necesidades reales.
4. Organización de los diferentes procesos que se llevan a cabo en el Hospital.
5. Control de gestión y utilización óptima de recursos de salud.
6. Obtención de datos estadísticos en tiempo real, posibilitando realizar estudios científicos de las enfermedades más comunes, entre otros.

El presente documento se encuentra estructurado en cuatro capítulos, el primero de ellos, **FUNDAMENTACIÓN TEÓRICA**, ubica al lector en el ambiente de desarrollo del módulo Admisión, justificándose las tendencias, tecnologías, metodologías y herramientas que fueron utilizadas para el desarrollo del mismo. Seguidamente el capítulo, **CARACTERÍSTICAS DEL SISTEMA**, contiene un marco conceptual asociado a la información que será manipulada por el sistema, descripción de los procesos del negocio, modelo de negocio, especificación de los requisitos de software y definición de los casos de uso.

El tercer capítulo **ANÁLISIS Y DISEÑO DEL SISTEMA** se centra en la definición del modelo de análisis, modelo de clases de análisis y en el diseño. En el cuarto y último, **IMPLEMENTACIÓN**, se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de la información, en el área admisión de las instituciones hospitalarias.

# Fundamentación teórica

---

## CAPITULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza una exposición detallada de los principales conceptos relacionados con el campo de acción, así como un estudio de las tendencias, tecnologías, herramientas y metodologías actuales, con el fin de proponer las más adecuadas a la solución del problema. El mismo tiene como propósito dar a conocer el estado del arte correspondiente a los Sistemas de Información Hospitalario a nivel nacional y global.

### **1.1 Conceptos básicos relacionados con el dominio del problema (2)**

#### **1.1.1 Inscripción**

La inscripción del paciente es la actividad delantera en todo hospital. Cada paciente tiene que registrarse antes de conseguir cualquier consulta, tratamiento, e investigaciones. La inscripción involucra, información general y demográfica sobre el paciente al que se le asigna un único número de inscripción central y se le entrega una tarjeta de identificación.

#### **1.1.2 Admisión**

Existe un proceso estrechamente relacionado con la hospitalización de los pacientes al que generalmente se llama Admisión. Esta sección es la encargada de almacenar toda la información referente a los pacientes durante su estancia en el hospital así como manejar y mantener actualizados todos estos datos.

#### **1.1.3 Historia clínica**

Sumergida dentro de la inscripción y admisión hospitalarias se encuentra la historia clínica (HC), que es un documento que surge en el contacto entre el equipo de salud y los usuarios. Este es único y válido desde el punto de vista clínico y legal. Es creada en el proceso de inscripción en un hospital y luego es manejada por los médicos para almacenar el comportamiento evolutivo de un paciente durante su estancia en el centro. No se limita a ser una narración o exposición de hechos simplemente, sino que incluye juicios, documentos y procedimientos; es un documento que se va haciendo en el tiempo, documentando fundamentalmente la relación médico-paciente.

## Fundamentación teórica

---

### **1.1.4 Ingreso**

Un ingreso es la admisión de un paciente al hospital, que puede ser por urgencias cuando se da el caso de que el paciente llega por cuerpo de guardia y la dolencia que presenta requiere de un ingreso urgente, o de manera electiva, esto significa que viene por consulta externa y que el ingreso ya estaba planificado. El ingreso se practica en una sala perteneciente a un servicio determinado, al paciente se le asigna una cama de la sala correspondiente al servicio por el que está ingresado, aunque puede suceder que sea ingresado fuera de servicio.

### **1.1.5 Ingreso fuera de servicio**

En el momento del ingreso el paciente debe ser asignado a un servicio especializado para atender las dolencias que presente, puede suceder que las salas de dicho servicio estén llenas y sea urgente el ingreso del paciente, en este caso se realiza lo que se conoce como ingreso fuera de servicio. Esto significa que se ingresa al paciente en un servicio que no es el que le corresponde y se especifica para el que debería ser trasladado en caso de liberarse alguna cama del servicio correspondiente.

### **1.1.6 Sala**

Recinto o lugar del hospital en el que se encuentran habilitadas las camas donde se realizarán los ingresos. Pertenece a determinado servicio.

### **1.1.7 Cama**

Ubicación física en la que se encuentra el paciente ingresado. Pertenece a una sala determinada.

### **1.1.8 Servicio**

Tiene asociadas salas y camas. Hace referencia a las diferentes especialidades del hospital, por ejemplo: Nefrología, Dermatología, Cardiología, Neurología, entre otras.



## **1.1.9 Egreso**

Es la salida o Alta del hospital. El paciente egresa porque mejoró totalmente, o por causas de fallecimiento.

## **1.2 Sistemas automatizados existentes vinculados al campo de acción**

En el competitivo mercado del software existen varios ejemplos de sistemas para hospitales y organizaciones de salud. De todos ellos, solo se analizan los de mayor envergadura que brindan el servicio de admisión.

### **1.2.1 Hosix-V**

Es uno de los productos desarrollados por la empresa SIVSA, con más de 15 años de experiencia (3) en el desarrollo de software para el sector de la salud. Utiliza una arquitectura cliente-servidor por lo que es accesible un navegador web mediante el protocolo http. Está compuesto por un grupo de módulos orientados a la gestión específica de cada servicio o departamento, cubriendo los aspectos más importantes de una unidad de salud. Ha sido desarrollado utilizando como lenguaje de programación Microsoft Visual Basic y herramientas que son propiedades de la compañía Microsoft.

Posee un módulo Hospitalización que permite controlar los procesos correspondientes a los ingresos de los pacientes. Su organización autoriza al sistema a registrar e informar, de cualquier movimiento desde la llegada del paciente hasta su salida, así como de los preingresos existentes. Ofrece una gestión completa de las camas hospitalarias. Permite ver el nivel de ocupación, las camas libres por totales y por servicio, los traslados, facilitando con todo ello la optimización de recursos del hospital. Su precio oscila entre los \$500,000 y 2, 000,000 de dólares.

### **1.2.2 Care2x**

Este sistema, desarrollado desde el 2002, rápidamente fue soportado por una comunidad de desarrolladores en el mundo, basado en estándares de código abierto. Integra datos, funciones y flujo de tareas en un entorno de cuidados de la salud (4). Está conformado por cuatro componentes principales

## Fundamentación teórica

---

que a su vez pueden funcionar de forma independiente. Soluciona el problema de la dependencia de plataforma y la redundancia e incompatibilidad de datos. Está implementado en PHP 4.0.4, utilizando como gestor de Base de Datos MySQL 3.2x - 4.0.x y corre sobre un servidor Web Apache 1.X.

Cuenta con un módulo de admisión en el cual existen funcionalidades como la inscripción de un paciente nuevo, búsqueda de paciente ingresado y búsqueda de casos archivados. Una de las principales desventajas que presenta es que está desarrollado en PHP 4.04, versión que ya no cuenta con soporte técnico por parte de los desarrolladores y aunque es software libre su adaptación a un entorno hospitalario es bastante engorrosa.

### **1.2.3 C-DAC's Sushrut**

Uno de los productos que comercializa la empresa C-DAC, radicada en la India es el C-DAC's Sushrut, que es un sistema de información hospitalario desarrollado con el objetivo de suavizar el flujo del tratamiento de un paciente en el hospital (5). Este sistema ha sido desarrollado utilizando una arquitectura cliente servidor, y puede ser utilizado en múltiples plataformas. Incorpora un sistema de información clínica computarizada e integrada para mejorar la administración del hospital y el cuidado de la salud de los pacientes. Además provee un archivo médico electrónico muy preciso de los pacientes, incluyendo un almacén de datos del cual los archivos pueden ser utilizados para requerimientos estadísticos y para la investigación.

El módulo registro del paciente es el primer proceso que se gestiona en el sistema de información total del hospital. El registro contiene información general y demográfica sobre el paciente, quien cuenta con un número de registro central único y además con una tarjeta de identificación paciente, conocida como tarjeta del registro. El módulo hospitalización comienza cuando se asigna al paciente una cama en la sala, además se ocupa del tratamiento completo del paciente durante su estancia en el hospital. Es un sistema de muy buenas prestaciones pero su principal desventaja consiste en los altos precios que hay que pagar para obtenerlo.

### 1.2.4 Galenhos(R)

Ha sido diseñado con el propósito de apoyar a los establecimientos de salud en el correcto registro de información, clínica o administrativa, y la generación de información gerencial. Su desarrollo nace de la asistencia técnica prestada por la agencia de Estados Unidos para el desarrollo internacional (USAID), a través del proyecto Partners for Health Reform plus (PHRplus), a un hospital de Perú. Entre las ventajas que ofrece este sistema están:

1. Información estandarizada: favorece un registro adecuado, donde se utilizan todos los estándares establecidos por el ministerio de salud.
2. Bases de datos consolidadas y exportables: eliminan los problemas provenientes de la existencia de sistemas paralelos, incompatibles e incomunicados entre sí.
3. Generación de reportes clínicos de uso gerencial. Facilita la evaluación de su capacidad resolutive y eficiencia en el manejo de los recursos.
4. Altos estándares de seguridad informática: responden a la elevada sensibilidad de la información que se maneja, protegiendo la operación regular del sistema y favoreciendo los controles o auditorias.
5. Diseño modular: considera la ampliación y futuros desarrollos, que se extiendan en toda la complejidad hospitalaria con módulos adicionales.

Fue desarrollado en el lenguaje de programación Visual Basic 6.0 y utiliza el Microsoft SQL server como gestor de base de datos. Su principal desventaja radica en que no es un sistema multiplataforma y se ejecuta mediante una aplicación de escritorio.

### 1.2.5 X-HIS

Es una solución global de la compañía ISOFT, del grupo IBA Health, en un entorno abierto que, además de gestionar electrónicamente la historia clínica del paciente, permite la integración con otros sistemas de información. Este software funciona en múltiples plataformas (Unix, Windows, Linux), es independiente de la base de datos y posee una arquitectura cliente-servidor y en 3 capas. Incluye un módulo admisión diseñado para registrar y controlar la información administrativa del paciente durante su paso por el área

## Fundamentación teórica

---

de hospitalización. Es software propietario y se comercializa bajo altos precios por lo que no constituye una opción viable para los países subdesarrollados.

### **1.2.6 SIGHO**

El Sistema de Información para la Gerencia Hospitalaria (SIGHO) es un software basado en la Norma Oficial Mexicana NOM-168-SSA1-1998 referente al resguardo y uso del expediente clínico electrónico para facilitar las actividades de gerencia dentro del hospital y se apoya de estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos tales como el CIE-10 y CIE9MC.

El SIGHO cuenta con un módulo admisión que permite registrar los datos de los pacientes hospitalizados para después utilizarlos para búsquedas brindarle información actualizada a los módulos con los que se relaciona como son Consulta externa, Cirugía y Emergencia. Tiene como inconveniente ser un software propietario que se ejecuta mediante una aplicación escritorio.

### **1.2.7 Galen Hospital**

En Cuba, se han desarrollado también sistemas de gestión hospitalarios, tal es el caso de la empresa Softel que brinda soluciones para la salud y que radica en la Universidad de las Ciencias Informática. Uno de los productos desarrollados por Softel es Galen Hospital, sistema que está orientado hacia la informatización de la gestión de pacientes como elemento básico de control para mejorar la atención médica.

Brinda la información requerida para la actividad gerencial a todos los niveles y la elaboración de reportes estadísticos. Incluye un módulo hospitalización que asume los procesos vinculados con el registro, admisión y alta de los pacientes. Utiliza como gestor de bases de datos SQL server 2000 y sus módulos corren sobre los sistemas operativos Windows 2000 y Windows XP, por lo que los principales inconvenientes de este sistema es que es una aplicación que se ejecuta en el escritorio y además depende de tecnologías que son propiedades de compañías privadas.

## **1.3 Tendencias y tecnologías actuales**

La computación desde sus inicios ha sufrido muchos cambios, desde los grandes ordenadores que permitían realizar tareas en forma limitada y de uso un tanto exclusivo de organizaciones muy selectas, hasta los actuales ordenadores ya sean personales o portátiles que tienen las mismas e incluso mayores capacidades que los primeros y que están cada vez más introducidos en el quehacer cotidiano de una persona. Esta evolución de los sistemas informáticos ha dado lugar a la creación de numerosas herramientas que ayudan a organizar, diseñar y construir una aplicación informática, permitiendo minimizar el esfuerzo del equipo de desarrollo.

### **1.3.1 Navegadores Web**

La difusión de los distintos sistemas a través de Internet se logra mediante los navegadores Web. Un navegador Web o browser es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Internet. La funcionalidad básica de un navegador Web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos. Existen diferentes navegadores Web tales como Mozilla, Windows Internet Explorer, Firefox, Netscape, Opera, Konqueror, entre otros. (6)

### **1.3.2 Aplicaciones web**

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, Java Script, etc.) en la que se confía la ejecución al navegador (7).

## Fundamentación teórica

---

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales (8).

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva.

### **1.3.3 Arquitectura cliente-servidor**

Consiste básicamente en que un programa, cliente, realiza peticiones a otro programa, servidor, que le da respuesta. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

### **1.3.4 Arquitectura en capas**

El objetivo principal de esta arquitectura es la separación de la lógica de negocios de la lógica de diseño. Su ventaja principal radica en que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que

## Fundamentación teórica

---

sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API (de las siglas Application Program Interface) que existe entre niveles (9).

El diseño más utilizado actualmente es el diseño en tres capas, consiste en:

- **Capa de presentación:** es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz grafica y debe tener la característica de ser entendible y fácil de usar para el usuario.
- **Capa de negocio:** es donde residen los programas que se ejecutan. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

### 1.3.5 Tecnologías utilizadas en el proceso de desarrollo

#### 1.3.5.1 Java

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90 (10). El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria (11).

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación a código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o

compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible (12).

El lenguaje Java se creó con cinco objetivos principales:

- Usar la metodología de la programación orientada a objetos.
- Permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Incluir por defecto soporte para trabajo en red.
- Ejecutar código en sistemas remotos de forma segura.
- Fácil de usar y toma lo mejor de otros lenguajes orientados a objetos, como C++.

### **1.3.5.2 Capa de presentación**

#### **1.3.5.2.1 Java Server Faces (JSF)**

Es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE (13). JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Dos librerías de etiquetas personalizadas para Java Server Pages que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Beans administrados.

#### **1.3.5.2.2 RichFaces 3.2**

Es una biblioteca de componentes para JSF y un avanzado framework para la integración de AJAX con facilidad en la capacidad de desarrollo de aplicaciones de negocio. Los componentes RichFaces vienen listos para su uso sin tener que hacer muchas configuraciones, por lo que los desarrolladores pueden ahorrar tiempo de inmediato para aprovechar las características de los componentes para crear



aplicaciones web que proporcionan mejoras en gran medida en la experiencia del usuario más fiable. También incluye un fuerte apoyo para el soporte de temas de aplicaciones JSF y aprovecha al máximo los beneficios de JSF framework incluyendo, la validación y conversión junto con la gestión de estática y dinámica los recursos (14).

### **1.3.5.3 Capa de negocio**

#### **1.3.5.3.1 JBoss Seam 2.0**

Es un framework desarrollado por JBoss, una división de Red Hat. Combina a dos frameworks: Enterprise JavaBeans(EJB) y JavaServerFaces(JSF) (15). Gracias a él puede accederse a cualquier componente EJB desde la capa de presentación refiriéndote a él mediante su nombre de componente seam.

Seam introduce el concepto de contextos. Cada componente de Seam existe dentro de un contexto. El contexto conversacional por ejemplo captura todas las acciones del usuario hasta que éste sale del sistema o cierra el navegador - inclusive puede llevar un control de múltiples pestañas y mantiene un comportamiento consistente cuando se usa el botón de regresar del navegador (16).

Puede automáticamente generarse una aplicación web de altas, bajas cambio y modificaciones a partir de una base de datos existente utilizando una herramienta de línea de comandos llamada seam-gen incluida con el framework. Seam puede ser integrado con las bibliotecas de componentes JBoss RichFaces.

#### **1.3.5.3.2 JBoss Server 4.2**

Es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado (17). Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, esto hace que el desarrollo de las aplicaciones empresariales sean mucho más simples (18).

### **1.3.5.3.3 JBoss tools**

Es un conjunto de plugins y mejoras para Eclipse diseñado para ayudar a los desarrolladores de JBoss y J2EE a crear aplicaciones utilizando el menor tiempo y esfuerzo posible. Esta suite tiene herramientas para desarrollar en Seam 2.1, integración con Hibernate Tools, JBoss AS Tools, soporte para Drools y JBPM.

### **1.3.5.4 Capa de datos**

#### **1.3.5.4.1 Hibernate**

Es una herramienta de mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL (19).

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) (20). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base de datos operando sobre objetos, con todas las características de la programación orientada a objetos (POO). Convierte los datos entre los tipos utilizados por Java y los definidos por SQL , genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

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 (21).

### **1.3.5.4.2 PostgreSQL 8.3**

Es un sistema de gestión de base de datos relacional orientada a objetos de código abierto (22). Algunas de sus principales características son (23):

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Permite mantener la integridad referencial.

### **1.3.5.5 Interfaces de comunicación**

#### **1.3.5.5.1 Servicios web**

Un servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y

reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares (24).

### **1.3.5.5.2 XML**

Lenguaje de marcas ampliable (siglas en inglés de Extensible Markup Language). Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específico, por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML (25).

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

### **1.3.5.5.3 HL7**

Es un conjunto de estándares para el intercambio electrónico de información médica. HL7 (en inglés Help Level 7), hace referencia al nivel siete (aplicación) del modelo OSI. Los estándares HL7 son desarrollados por la organización ANSI del mismo nombre. Existe una idea errónea acerca de la organización HL7, consistente en que desarrolla software. En realidad, HL7 desarrolla especificaciones. A partir del 17 de diciembre de 2003, la ANSI ha aprobado una serie de estándares contenidos en la especificación de la Versión 3 HL7. Estos estándares emplean una metodología formal de modelamiento de casos de uso y una sintaxis que emplea XML.

### **1.3.5.6 Proceso Unificado de Desarrollo**

Es un Proceso Unificado de Desarrollo (en inglés RUP, Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (más conocido por sus siglas en inglés, UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Sus principales características se centran en: implementar las mejores prácticas en ingeniería de software, disciplinar la forma de asignar tareas y responsabilidades, quién hace qué, cuándo y cómo, administrar requisitos, usar arquitectura basada en componentes y controlar cambios y modelado visual del software.

RUP posee tres características fundamentales: su desarrollo es iterativo e incremental por lo que divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo; es guiado por los casos de uso, un caso de uso describe un fragmento de las funcionalidades del sistema que proporciona al usuario un resultado importante, los casos de uso guían el diseño construcción y prueba del sistema, esto significa que guían el proceso de desarrollo; RUP está centrada en la arquitectura, lo que le permite a los desarrolladores una mayor visibilidad del sistema, pues la arquitectura es una vista del diseño completo del software con las características más importantes resaltadas, dejando a un lado los detalles. Utiliza como lenguaje de modelado UML (Unified Modeling Language).

#### **1.3.5.6.1 Notación de Modelado de Procesos del Negocio**

La Notación de Modelado de Procesos del Negocio (en inglés BPMN, Business Process Management Notacion) es una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo.

BPMN fue inicialmente desarrollado por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el OMG (Object Management Group), luego de la fusión de las dos organizaciones en el año 2005. Su versión actual es la 1.1 y hay una versión futura propuesta, la 2.0.

El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente entendible por parte de todos los involucrados e interesados del negocio. Entre estos interesados están: los analistas de

## Fundamentación teórica

---

negocio, quienes definen y redefinen los procesos; los desarrolladores técnicos, responsables de implementar los procesos y los gerentes y administradores del negocio, quienes monitorean y gestionan los procesos. En síntesis BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.

Actualmente hay una amplia variedad de lenguajes, herramientas y metodologías para el modelado de procesos de negocio. La cada vez mayor adopción de la notación BPMN como estándar ayudará a unificar la expresión de conceptos básicos de procesos de negocio, por ejemplo procesos públicos y privados, orquestación, coreografía, etc. así como conceptos avanzados de modelado, por ejemplo manejo de excepciones, compensación de transacciones, entre otros.

El modelado en BPMN se realiza mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. Con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso. Las cuatro categorías básicas de elementos son:

- **Objetos de flujos:** eventos, actividades, rombos de control de flujo.
- **Objetos de conexión:** flujo de secuencia, flujo de mensaje, asociación.
- **Swimlanes:** carriles de piscina.
- **Artefactos:** objetos de datos, grupo, anotación.

Estas cuatro categorías de elementos ofrecen la oportunidad de realizar un diagrama simple de procesos de negocio. En estos tipos de diagramas se permite definir un tipo personalizado de objeto de flujo o un artefacto, si con ello se hace el diagrama más comprensible.

### 1.3.5.6.2 UML (Lenguaje Unificado de Modelado)

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos

concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

### **1.3.5.7 Herramientas**

Se definieron las herramientas que conforman el ambiente de desarrollo para el sistema de información hospitalario que se desea construir. Para ello se utilizará el Visual Paradigm como herramienta CASE (Computer Aided Software Engineering) basada en UML y que permite crear los diagramas y artefactos que se generen como parte de la documentación. Para la programación y desarrollo del sistema el entorno de desarrollo integrado Eclipse que cuenta con facilidades de refactorización, completamiento de código y es una herramienta multiplataforma que permite la integración con Hibernate y un controlador de versiones; y como cliente para el gestor de base de datos el PostgreSQL Maestro por las bondades visuales que presenta y la facilidad de su manejo.

### **Conclusiones**

En este capítulo se ha realizado un estudio de los principales sistemas existentes en el mundo de la gestión hospitalaria que incluyen el módulo Admisión. Además fue realizada una valoración de las principales tecnologías, metodologías y herramientas que son empleadas en el desarrollo de la aplicación. Los sistemas estudiados no satisfacen las necesidades que en dicha área existen, por lo que se decide implementar un software que resuelva los problemas presentes en estos.

# Características del sistema

---

## CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

En este capítulo se describe detalladamente el flujo de trabajo y las reglas del negocio. Se plantean los requisitos funcionales y no funcionales de la aplicación a desarrollar y se describe la solución propuesta en términos de casos de uso del sistema.

### **2.1 Flujo actual de los procesos involucrados en el campo de acción**

Un proceso es una actividad que utiliza recursos y que se gestiona, es un conjunto de tareas lógicamente relacionadas que tienen como objetivo conseguir un resultado bien definido dentro de un negocio.

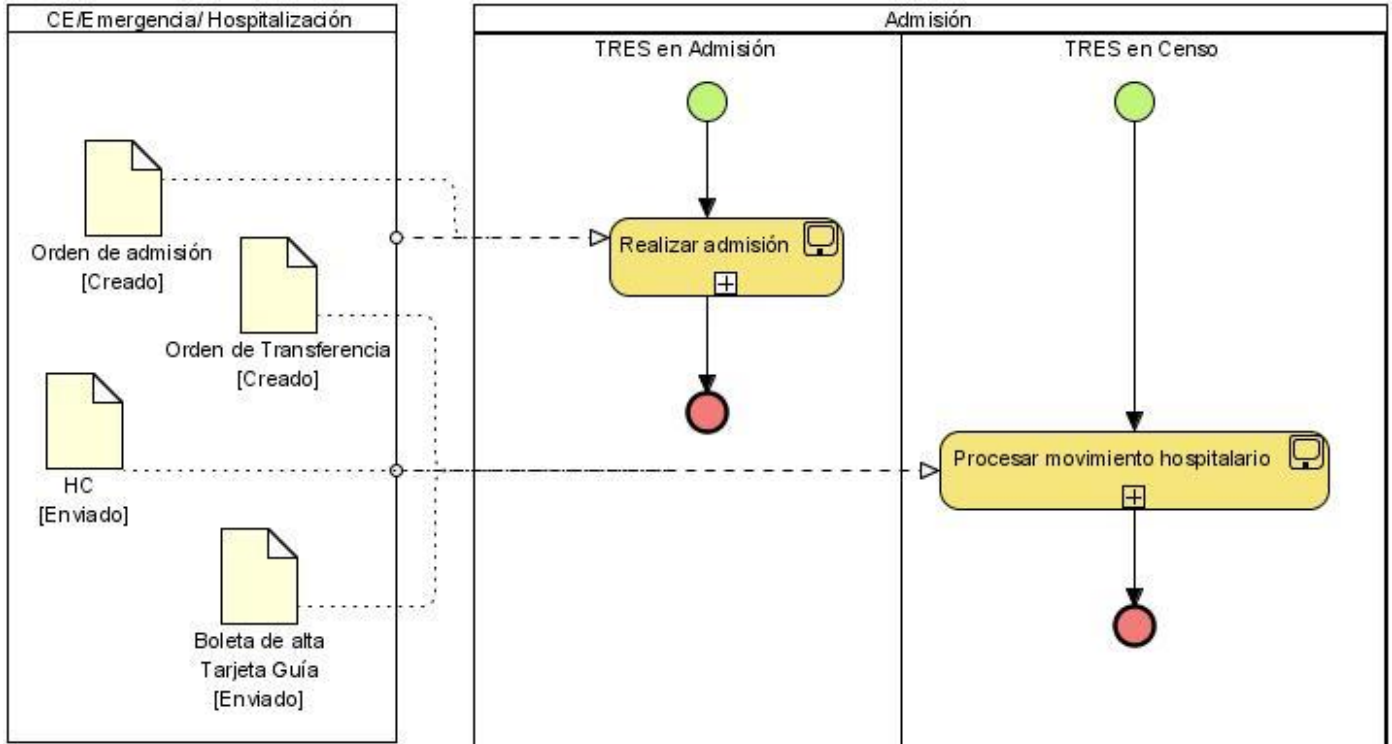
El área admisión es responsable de ejecutar dos procesos fundamentales:

- **Realizar admisión:** permite ingresar a los pacientes en los diferentes servicios, en dependencia de la orientación de su médico.
- **Procesar movimiento Hospitalario:** permite realizar las transferencias, egresos y el conteo diario de las camas disponibles y ocupadas por servicio.



# Características del sistema

## 1 Diagrama de procesos actuales



## 2.2 Objeto de automatización

De la idea de optimizar la calidad de los servicios que se prestan en admisión, se desean automatizar todos los procesos que en esta área se llevan a cabo:

**Realizar admisión:** la automatización de este proceso permitirá gestionar los ingresos en los distintos servicios de la institución hospitalaria

**Procesar movimiento hospitalario:** permitirá generar reportes y datos estadísticos relacionados fundamentalmente con los tipos de movimientos hospitalarios que se efectúan en la institución.

## Características del sistema

---

### 2.3 Información que se maneja

Como resultado de estos procesos se generan y archivan algunos documentos que son usados para mantener información relacionada con el paciente, su paso por la institución hospitalaria e información estadística. Algunos de ellos se describen a continuación.

- **Orden de admisión:** es un documento emitido por un médico, donde especifica los datos de ingreso de un paciente.
- **Registro de admisiones:** mantiene información actualizada de cada una de las admisiones efectuadas en la institución hospitalaria.
- **Panel de control de camas:** permite consultar la disponibilidad de camas con que cuenta cada uno de los departamentos y servicios.
- **Registro de egresos:** archiva información actualizada de cada uno de los egresos efectuadas en la institución hospitalaria.
- **Registro de transferencias:** guarda la información relacionada con los movimientos de cama de los pacientes, que pueden ser entre servicios o en un mismo servicio.
- **Censo de pacientes fuera de servicio:** recoge todos los pacientes que debían haber sido hospitalizados en un servicio, pero por falta de disponibilidad de camas en él, tuvo que ser ingresado en otro.
- **Censo diario de los departamentos y Censo diario de los servicios:** registra diariamente información relacionada con los habitantes de cada servicio y departamento de la institución hospitalaria.
- **Orden de transferencia:** documento que emite un médico donde especifica que un paciente debe ser trasladado y el motivo de traslado.

## Características del sistema

---

- **Permiso de salida:** guarda información sobre permisos de salida temporal, que se le da al paciente, de la institución hospitalaria

### 2.4 Modelo de negocio

Para conseguir sus objetivos, una empresa organiza su actividad por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (por ejemplo, trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan la estructura de la información y las políticas de la empresa. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades, roles y reglas de negocio.

#### 2.4.1 Actores de procesos de negocio.

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

ACTORES INVOLUCRADOS	
Rol	Funciones
Técnico (a) de Registros y Estadísticas de Salud (TRES) en admisión.	Es el (la) principal ejecutor (a) de todas las actividades que se realizan dentro del proceso de las admisiones hospitalarias.
Técnico (a) de Registros y Estadísticas de Salud (TRES) en Servicio.	Es el (la) representante del área de Secretaría de Hospitalización de la Sección de Admisión en cada servicio.

## Características del sistema

---

### 2.4.2 Realizar admisión

<b>PROCESO:</b>	Realizar admisión
<b>Misión:</b>	Realizar las admisiones de pacientes, así como buscar los complementos de los mismos en el área de Archivo.
<b>Responsable:</b>	Especialista funcional del área de Admisión
<b>ACTORES INVOLUCRADOS</b>	
<b>Rol</b>	<b>Funciones</b>
Técnico (a) de Registros y Estadísticas de Salud (TRES) en admisión.	Es el (la) principal ejecutor (a) de todas las actividades que se realizan dentro del proceso de las admisiones hospitalarias.
Técnico (a) de Registros y Estadísticas de Salud (TRES) en Servicio.	Es el (la) representante del área de Secretaría de Hospitalización de la Sección de Admisión en cada servicio.
<b>ACTIVIDADES</b>	
<b>Actividad:</b>	Procesar admisión
<b>Flujo de Información</b>	
El paciente o su acompañante se presenta en el área de admisión con una Orden de admisión (OA), debe portar además un elemento identificativo, dígame cédula de identidad, partida de nacimiento y en caso de ser extranjero su pasaporte. El (la) TRES en admisión busca en el Panel de control de camas, la cama asignada en la Orden de admisión para verificar que se encuentre disponible.	
<b>Actividad:</b>	Verificar disponibilidad de cama
<b>Flujo de Información</b>	
Si la cama que se le asignó al paciente se encuentra ocupada en el momento del ingreso, el (la) TRES en admisión, se comunica con el servicio para el cual va el paciente y con el área de Censo para investigar si esta cama será liberada por un egreso.	
<b>Actividad:</b>	Solicitar cambio de cama

## Características del sistema

---

### Flujo de Información

Si la cama que fue asignada al paciente en la Orden de admisión se encuentra ocupada, el paciente es enviado con su médico a que le cambie la cama asignada.

**Actividad:** Consultar Historia Clínica (HC)

### Flujo de Información

El (la) TRES en admisión verifica con la Sección de Archivo que el paciente tenga historia clínica en el hospital.

**Actividad:** Buscar Historia clínica (HC)

### Flujo de Información

En el área de archivo se busca sobre el índice de pacientes, al paciente que ha sido solicitado por el (la) TRES en admisión. Si encuentra la Tarjeta Índice de paciente el Archivo la envía a el (la) TRES en admisión.

**Actividad:** Crear Historia clínica (HC)

### Flujo de Información

Si el número de Historia Clínica del paciente no existe, se hace necesario crear la misma. Para esto, se toman los datos del paciente y se anotan en el registro de hospitalizaciones.

**Actividad:** Confirmar admisión

### Flujo de Información

Si el paciente tiene Historia clínica el (la) TRES en admisión coloca información de la Tarjeta índice de paciente en la Orden de Admisión y en la Hoja frontal de la HC. La TIP que ha sido enviada por Archivo activo se coloca en el panel de control de camas.

Si el paciente no tenía Historia clínica se crea la TIP y se coloca información de la Tarjeta índice de paciente en la Orden de Admisión y en la Hoja frontal de la HC. La TIP que se crea se coloca en el Panel de control de camas.

Finalmente, se adjunta la Orden de admisión a la hoja frontal de la HC, así como las Historias de las especialidades correspondientes en dependencia del servicio en el que ha ingresado el paciente, en el caso que el servicio no tiene una hoja especializada se le coloca la HC general Parte II y III.

Si el paciente se ubica en un servicio que no es el tratante, se saca una copia de la Orden de admisión y se guarda en el registro Control de camas prestadas.

Cuando se confirma la admisión se actualiza el Registro de admisiones.

**Actividad:** Solicitar complementos

### Flujo de Información

El (la) técnico (a) de registro y estadísticas de salud en servicio, hace una solicitud al área de admisión para que se busque un complemento. El (la) TRES en admisión, anota los datos del paciente al que se le quiere buscar el complemento en el libro de Solicitudes de complementos, luego este libro es enviado a archivo donde buscan el complemento y lo envían a admisión.

Luego de recibir el complemento solicitado el (la) TRES en admisión lo envía con el mensajero al

## Características del sistema

---

servicio de hospitalización.

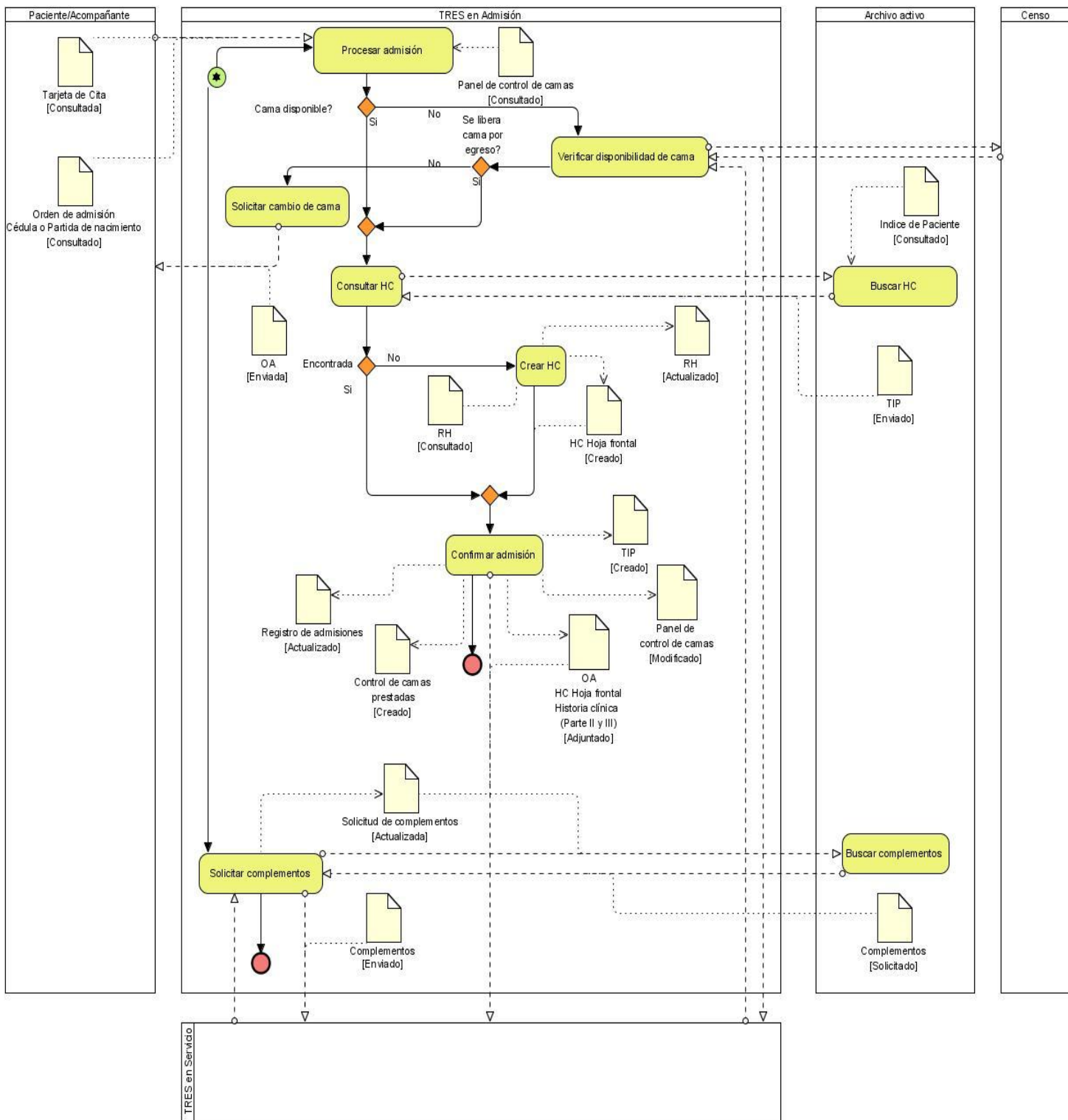
**Actividad:** Buscar complementos

### **Flujo de Información**

El área de Archivo recibe una solicitud del complemento de un paciente, lo busca y lo envía con el (la) mensajero (a) al área de Admisión.

### **DIAGRAMA DE PROCESO**

# Características del sistema



## Características del sistema

---

### 2.4.4 Procesar movimiento hospitalario

<b>PROCESO:</b>	Procesar movimiento hospitalario
<b>Misión:</b>	Realizar las transferencias, egresos y el conteo diario de las camas disponibles y ocupadas por servicio.
<b>Responsable:</b>	Especialista funcional del área de Admisión
<b>ACTORES INVOLUCRADOS</b>	
<b>Rol</b>	<b>Funciones</b>
Técnico (a) de registro y estadísticas de salud (TRES) en Censo.	Es el (la) principal ejecutor (a) de todas las actividades que se realizan dentro del proceso de los movimientos hospitalarios del paciente.
Técnico (a) de registro y estadísticas de salud (TRES) en Servicio.	Es el (la) representante del área de Secretaría de Hospitalización de la Sección de Admisión en cada servicio.
<b>ACTIVIDADES</b>	
<b>Actividad:</b>	Transferir paciente
<b>Flujo de Información</b>	
El (la) técnico (a) de registro y estadísticas de salud (TRES) en servicio, se presenta en el área de Censo con la Orden de transferencia. El (la) técnico (a) de registro y estadísticas de salud (TRES) en censo procesa la transferencia, verificando en el panel de control de camas, que la ubicación que le ha sido asignada al paciente esté disponible. La transferencia puede ser entre servicios o dentro del mismo servicio. Luego se actualiza el Panel de control de camas, cambiando la Tarjeta de índice de paciente, de la ubicación antigua a la que ha sido transferido. Al mismo tiempo se llena el Registro de transferencias.	
<b>Actividad:</b>	Realizar egreso
<b>Flujo de Información</b>	
El (la) TRES en Censo, recibe la Historia clínica que incluye la Boleta de alta o en el caso que el médico decida quedarse con la HC se envía la Boleta de alta con la Tarjeta guía explicando los motivos por los que se queda la HC. Luego busca la TIP que está ubicada en el Panel de control de camas y la adjunta a la HC o a la Boleta de alta, modificando así el Panel de control de camas al dejar disponible una ubicación. De esta forma la TIP que se coloca en la Boleta de alta será enviada al Archivo junto con la Historia clínica. Finalmente, (la) TRES en Censo actualiza el	



## Características del sistema

---

Registro de egresos.

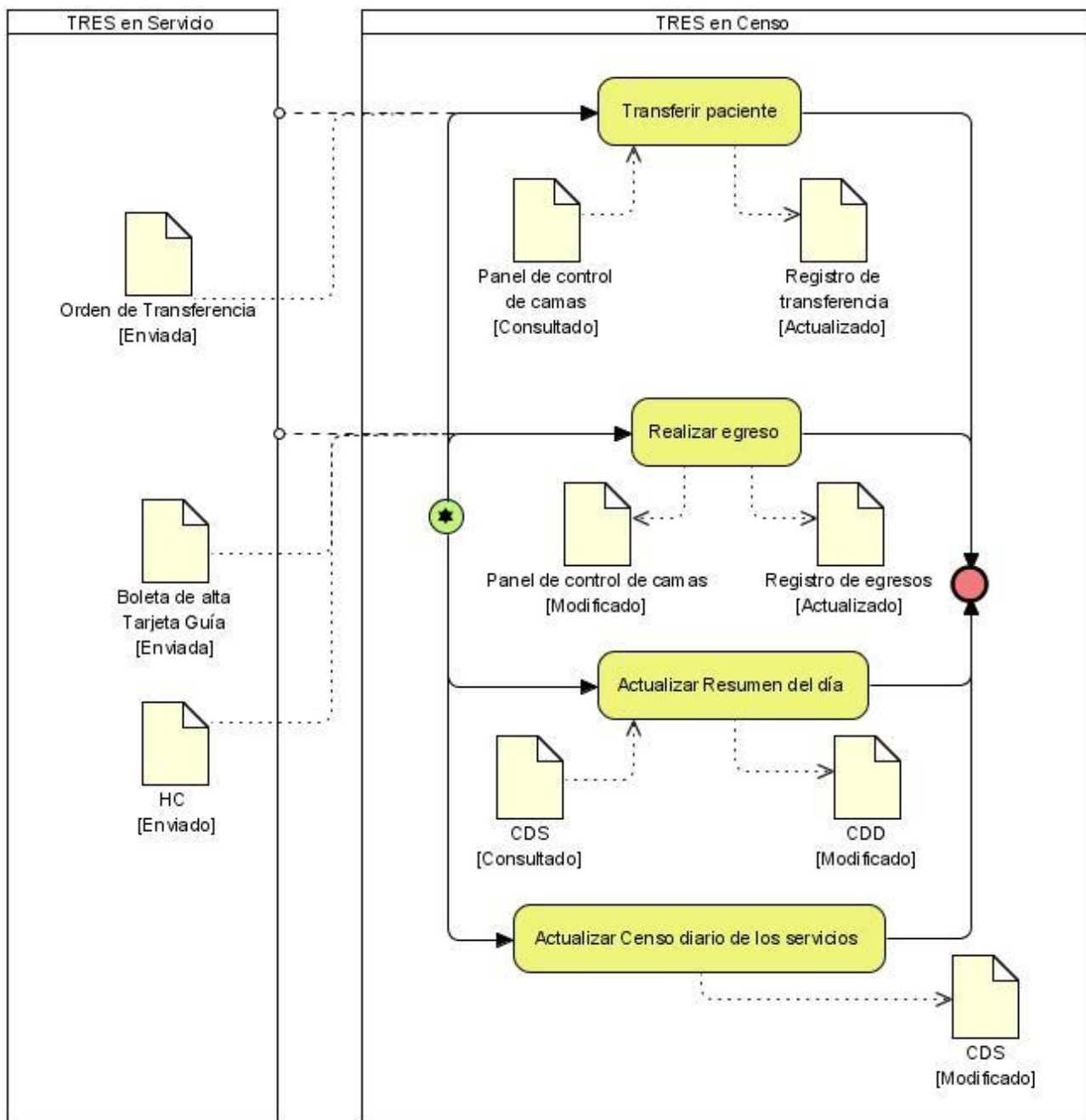
**Actividad:** Actualizar resumen del día

### **Flujo de Información**

El (la) TRES en Censo, diariamente genera el Censo diario de los departamentos, en el turno de la noche, a partir del Censo diario de los servicios.

### **DIAGRAMA DE PROCESO**

# Características del sistema



### **2.5 Propuesta del sistema**

Teniendo en cuenta los estudios realizados y después de hacer un profundo análisis del objeto de estudio, se llega a la conclusión de implementar un sistema que gestione la información que se genera y es manipulada en las instituciones hospitalarias. El mismo, debe estar centrado en el paciente desde el instante en que arriba al hospital hasta que culmine su atención en los distintos servicios, manteniendo toda la información que sobre él se genere.

Además la puesta en práctica de este sistema no solo beneficia a los pacientes, sino también a todos los especialistas y demás usuarios del software. Para cumplir con lo anteriormente dicho, se han capturado los requisitos funcionales partiendo de la necesidad real de los procesos de negocio de las instituciones hospitalarias.

#### **2.5.1 Especificación de los requerimientos de software**

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo los usuarios) y los desarrolladores sobre qué debe y que no debe hacer el sistema.

##### **2.5.1.1 Requerimientos funcionales**

- 1 Realizar admisión
  - 1.1 Realizar admisión
  - 1.2 Seleccionar Orden de admisión
  - 1.3 Buscar admisiones
  - 1.4 Ver datos de admisión
  - 1.5 Modificar admisión
  - 1.6 Eliminar admisión
  - 1.7 Ver detalles de admisión

## Características del sistema

---

### 2 Generar reportes

2.1 Generar Registro de admisiones y transferencias

2.2 Generar Registro de egresos

2.3 Generar Censo diario

2.4 Generar Censo de pacientes fuera de servicio

2.5 Generar Censo por sexo

2.6 Generar admisiones y transferencias por procedencia y mes

2.7 Generar admisiones y transferencias con diagnóstico preliminar

2.8 Generar reporte de ocupación

2.9 Generar reporte de admisiones y transferencias por mes de Medicina crítica por motivo de alta

2.10 Generar reporte de admisiones y transferencias por mes sin Medicina crítica sin anulaciones

2.11 Generar movimiento de camas por paciente

2.12 Generar bitácora de camas

2.13 Generar registro de altas con servicio de admisión

2.14 Generar reporte de admisiones y transferencias por investigación de demencia

2.15 Generar índice de hospitalizados

2.16 Generar reporte de recepciones por pacientes

2.17 Generar movimiento de camas por egresos

2.18 Generar reporte de altas en un mes por permisos

2.19 Generar reporte de transferencias y recepciones

2.20 Generar auditoría de admisiones por fechas de registro

2.21 Generar reporte de movimiento de camas

2.22 Generar chequeo histórico de uso de camas

2.23 Generar chequeo de transferencias

2.24 Generar reporte de admisiones y transferencias por mes de Medicina crítica

2.25 Generar chequeo de admisiones

2.26 Generar reporte de admisiones y transferencias para un día en particular

2.27 Generar reporte de recepciones tipo T2

## Características del sistema

---

- 2.28 Generar reporte de auditorías de transferidos
- 2.29 Generar reporte de admisiones y transferencias por tipo
- 2.30 Generar reporte de auditorías de recibidos
- 2.31 Generar reporte de altas por motivo de alta
- 2.32 Generar reporte de altas de un mes de indocumentados
- 2.33 Generar reporte de altas detalladas del movimiento hospitalario
- 2.34 Generar censo de indocumentados
- 2.35 Generar reporte de pacientes al cierre de mes según admisiones
- 2.36 Generar censo de cantidad por departamento y servicio
- 2.37 Generar reporte de referidos por procedencia
- 2.38 Generar reporte de referidos por procedencia y diagnóstico

### **2.5.1.2 Requerimientos no funcionales**

Los requisitos no funcionales se refieren a las propiedades o cualidades del sistema. Estos responden a cualidades que el producto debe tener y las características para que este sea atractivo, confiable, usable y seguro.

#### **2.5.1.2.1 Usabilidad**

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

Usuarios normales: 20 días

Usuarios avanzados: 30 días

#### **2.5.1.2.2 Seguridad**

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

## Características del sistema

---

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

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

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

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

### **2.5.1.2.3 Rendimiento**

El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.

El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

### **2.5.1.2.4 Soporte**

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

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

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

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

## Características del sistema

---

### **2.5.1.2.5 Hardware**

#### **2.5.1.2.5.1 Estaciones de trabajo**

En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador que cumpla con los estándares web (se recomienda IE 7 o superior o Firefox 2.x).

Por lo que se escogieron estaciones de trabajo de 256Mb de memoria RAM y un microprocesador de 2.0Hz con Sistema operativo Linux.

#### **2.5.1.2.5.2 Servidores**

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

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

#### **2.5.1.2.6 Software**

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

## Características del sistema

---

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

### **2.5.1.2.7 Restricciones de diseño**

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

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

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

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

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

### **2.5.1.2.9 Interfaz**

#### **2.5.1.2.9.1 Interfaces de usuario**

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

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

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

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



## Características del sistema

---

### **2.5.1.2.9.2 Interfaces de comunicación**

Para el intercambio electrónico de datos entre aplicaciones se usará el estándar HL7 (Health Level Seven).

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

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

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

### **2.5.1.2.10 Portabilidad**

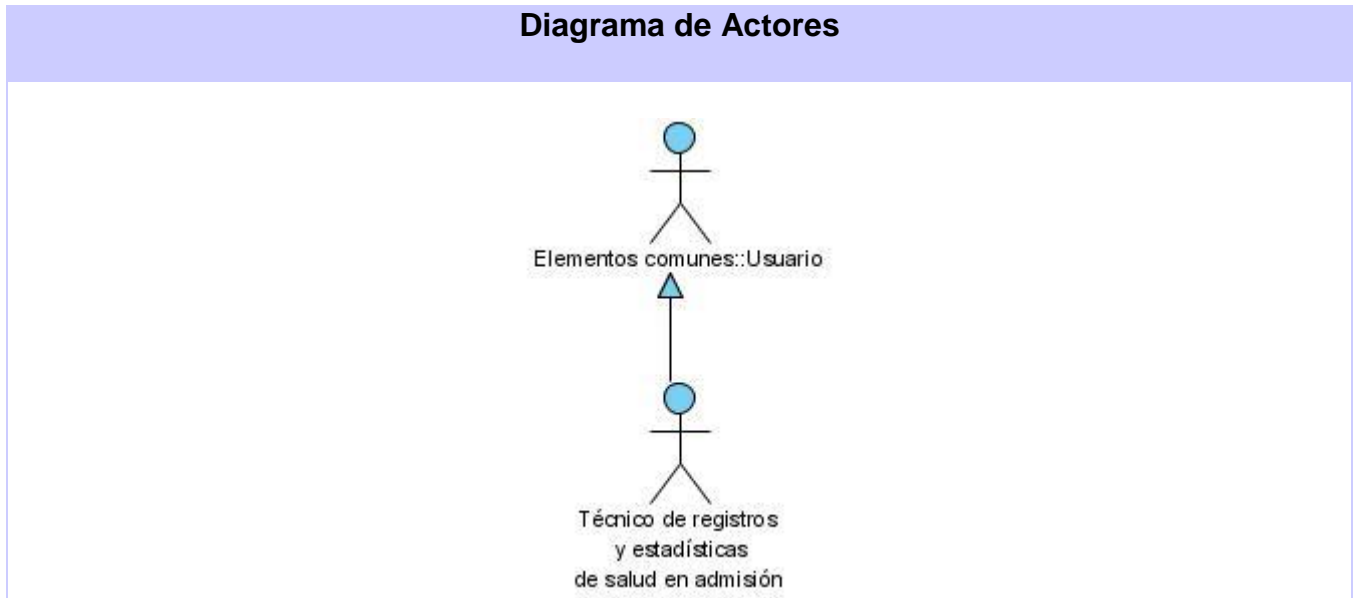
El producto podrá ser utilizado bajo los sistemas operativos Linux o Windows.

## **2.5.2 Modelo de casos de uso del sistema**

### **2.5.2.1 Definición de actores**

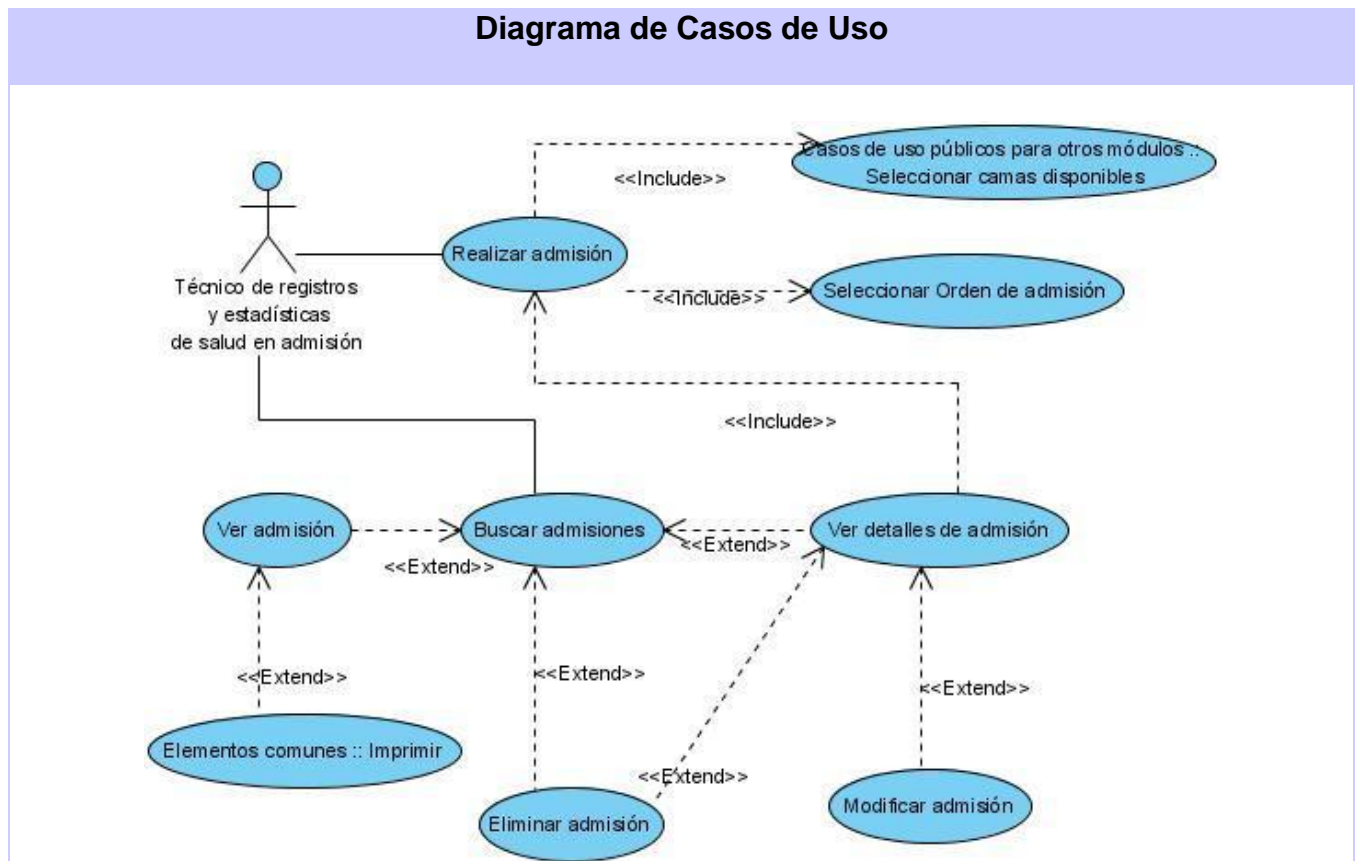
**Técnico de registros y estadísticas de salud en admisión:** es el usuario generalizado que desarrolla el proceso realizar admisión y tiene acceso a generar los reportes relativos a la sección de admisión.

## 2.5.2.2 Vista global de actores



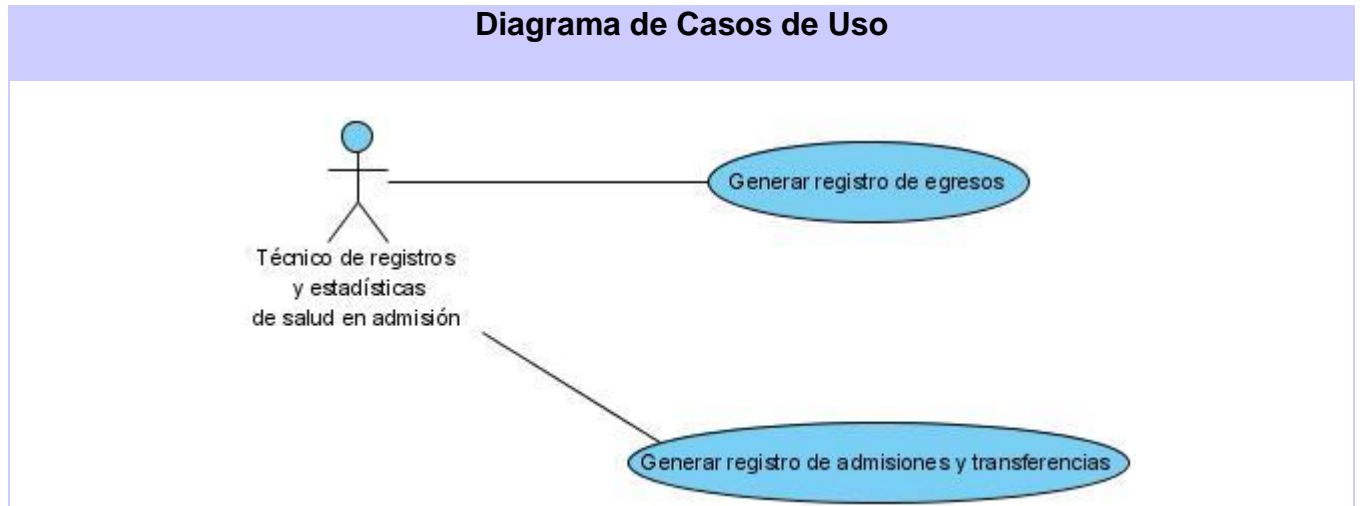
## 2.5.2.3 Diagrama de casos de uso

Realizar admisión



## Características del sistema

Generar reportes



### 2.5.2.4 Descripción textual de los casos de uso

Realizar admisión	
<b>Propósito:</b>	Realizar admisión
<b>Actores:</b>	Elementos comunes :: Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción Realizar admisión. El sistema brinda la posibilidad de seleccionar el paciente que se desea admitir, e introducir los datos para el ingreso. El actor introduce los datos del ingreso y el sistema realiza la admisión. El caso de uso termina.
<b>Precondición:</b>	No existe
<b>Poscondición:</b>	Se realizó una admisión.
<b>Referencias:</b>	RF-1.1

## Características del sistema

---

<b>Seleccionar Orden de admisión</b>	
<b>Propósito:</b>	Seleccionar Orden de admisión
<b>Actores:</b>	Elementos comunes :: Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción Seleccionar Orden de admisión, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la Orden de admisión deseada, el sistema carga en la vista anterior la Orden de admisión seleccionada, el caso de uso termina.
<b>Precondición:</b>	Debe existir al menos una Orden de admisión
<b>Poscondición:</b>	Se seleccionó Orden de admisión dado criterios
<b>Referencias:</b>	RF-1.2
<b>Complejidad</b>	Media

<b>Buscar admisiones</b>	
<b>Propósito:</b>	Buscar admisiones
<b>Actores:</b>	Técnico de registros y estadísticas de salud en admisión
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción Buscar admisiones, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar el ingreso, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra las admisiones que cumplen con los criterios de búsqueda, el caso de uso termina.
<b>Precondición:</b>	Debe existir al menos un paciente hospitalizado.
<b>Poscondición:</b>	Se buscó una admisión dado criterios.
<b>Referencias:</b>	RF-1.3
<b>Complejidad</b>	Media

## Características del sistema

---

Ver datos de admisión	
<b>Propósito:</b>	Ver datos de admisiones
<b>Actores:</b>	Técnico de registros y estadísticas de salud en admisión
<b>Resumen:</b>	El caso de uso inicia cuando el actor selecciona una admisión y accede a la opción de Ver datos de admisión, el sistema muestra los datos de la admisión, el caso de uso termina.
<b>Precondición:</b>	Para ver los datos de una admisión, esta debe estar seleccionada.
<b>Poscondición:</b>	Se vieron los datos de una admisión.
<b>Referencias:</b>	RF-1.4
<b>Complejidad</b>	Media

Modificar admisión	
<b>Propósito:</b>	Seleccionar Orden de admisión
<b>Actores:</b>	Elementos comunes :: Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el actor selecciona una admisión y accede a la opción Modificar admisión. El sistema muestra los datos de la admisión 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 admisión, el caso de uso termina.
<b>Precondición:</b>	Para modificar los datos de una admisión, esta debe haber sido seleccionada.
	Se modificó una admisión por el usuario.
<b>Referencias:</b>	RF-1.5
<b>Complejidad</b>	Media

## Características del sistema

---

Eliminar admisión	
<b>Propósito:</b>	Eliminar admisión
<b>Actores:</b>	Elementos comunes :: Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el actor selecciona una admisión y accede a la opción Eliminar admisión. El sistema elimina la admisión y el caso de uso termina.
<b>Precondición:</b>	Para eliminar una admisión, esta debe haber sido seleccionada
<b>Poscondición:</b>	Se eliminó una admisión por el usuario.
<b>Referencias:</b>	RF-1.6
<b>Complejidad</b>	Media

Ver detalles de admisión	
<b>Propósito:</b>	Ver datos de admisiones
<b>Actores:</b>	Técnico de registros y estadísticas de salud en admisión
<b>Resumen:</b>	El caso de uso inicia cuando el actor selecciona una admisión y accede a la opción de Ver datos de admisión, el sistema muestra los datos de la admisión, el caso de uso termina.
<b>Precondición:</b>	Para ver los datos de una admisión, esta debe estar seleccionada.
<b>Poscondición:</b>	Se vieron los datos de una admisión.
<b>Referencias:</b>	RF-1.8
<b>Complejidad</b>	Media

## Características del sistema

---

<b>Generar Registro de admisiones y transferencias</b>	
<b>Propósito:</b>	Generar registro de admisiones y transferencias
<b>Actores:</b>	Técnico de registros y estadísticas de salud
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción Generar Registro de admisiones y transferencias, el sistema brinda la posibilidad de introducir los criterios para generar el registro. El sistema genera el reporte de información. El actor selecciona la opción de imprimir el registro, el sistema imprime los datos del registro, el caso de uso termina.
<b>Precondición:</b>	Debe existir al menos una admisión o transferencia.
<b>Poscondición:</b>	Se generó un Registro de admisiones y transferencias dado criterios
<b>Referencias:</b>	RF-2.1
<b>Complejidad</b>	Media



## Características del sistema

---

<b>Generar Registro de egresos</b>	
<b>Propósito:</b>	Generar Registro de egresos
<b>Actores:</b>	Técnico de registros y estadísticas de salud
<b>Resumen:</b>	El caso de uso inicia cuando el actor accede a la opción Generar Registro de egresos. 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 los egresos realizados, genera el reporte de información. El actor, selecciona la opción de imprimir el reporte. El sistema imprime los datos del reporte y el caso de uso termina.
<b>Precondición:</b>	Debe haberse registrado al menos un egreso.
<b>Poscondición:</b>	Se generó el Registro de egresos
<b>Referencias:</b>	RF-2.2
<b>Complejidad</b>	Media

### Conclusiones

En el presente capítulo fueron explicados detalladamente los procesos que se realizan en el área de admisión. Al analizar estos procesos se evidencian las dificultades a las que son expuestas los pacientes y los trabajadores de las instituciones hospitalarias. Es por ello que se diseña una aplicación que permita optimizar esos procesos. Se realizó el modelo del negocio y se identificaron los actores, trabajadores y casos de uso del negocio, así como los requerimientos funcionales y no funcionales del sistema, basados en los requisitos de desarrollo que propone la metodología RUP con UML.

## CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA

### 3.1 Descripción de la arquitectura, fundamentación

Un sistema software es una única entidad, pero al arquitecto y a los desarrolladores les resulta útil presentar el sistema desde diferentes perspectivas para comprender mejor el diseño. Estas perspectivas son vistas del modelo del sistema. Todas las vistas juntas representan la arquitectura.

La arquitectura abarca decisiones importantes sobre: la organización del sistema software; los elementos que compondrán el sistema y sus interfaces, junto con sus comportamientos, tal y como se especifican en las colaboraciones entre estos elementos; la composición de elementos estructurales y del comportamiento en subsistemas progresivamente más grandes; el estilo de la arquitectura que guía esta organización, los elementos y sus interfaces, sus colaboraciones y su composición.

Sin embargo la arquitectura software está afectada no solo por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y compromisos económicos, tecnológicos y la estética (26).

#### 3.1.1 Patrones de arquitectura

Un patrón es una solución a un problema que aparece con frecuencia. Se presentan como plantillas donde se les asigna un nombre y un resumen de los problemas, las fuerzas que lo hacen surgir, y las ventajas y desventajas que provee su utilización.

El patrón de arquitectura que más se pone de manifiesto en el diseño de la aplicación es el Modelo-Vista-Controlador (MVC). Este patrón es útil principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de mejor manera, facilitando la programación en diferentes capas de forma paralela e independiente.

La vista(o presentación) es la representación de los datos en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web la vista es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.

## Análisis y diseño del sistema

---

La capa de presentación está desarrollada básicamente con JSF, usando la librería de componentes RichFaces 3.2.0 G.A., esta integra fácilmente con el framework de integración escogido (Seam) y permite generar vistas no necesariamente basadas en HTML (PDF, etc.), adiciona además controles listos para usar y el framework de extensión AJAX para los controles JSF básicos Ajax4Jsf.

Incluye conversión y validación de campos, establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario, un modelo orientado a eventos y combinado con Facelets, se elimina la necesidad de dos motores de renderización (uno para JSF y otro para JSP) mejorando el rendimiento en general además de que brinda la capacidad añadida de la tecnología de plantillas de Facelets. Por su parte los controles para interfaz de usuario de Seam adicionan varias mejoras a JSF, desde validación, integración de la navegación en la interfaz de usuario basada en flujos de navegación o procesos del negocio, etc.

El controlador es el encargado de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando los datos en caso de que sea requerido. En esta capa y como framework de integración se usa Seam, un poderoso y moderno framework creado para unificar (como su nombre lo indica) todas las tecnologías estándares JSF, EJB3, JPA, además de BPM (Business Process Management).

Fue creado desde el inicio para eliminar la complejidad a nivel desde arquitectura hasta API, permitiendo la creación de complejas aplicaciones web basadas en POJOs, componentes de interfaz de usuario y el mínimo y solamente necesario XML. Se integra con librerías de controles de código abierto basadas en JSF como RichFaces, ICEFaces, etc.

Una de sus principales innovaciones es en el campo de la administración de estado, mientras que en los frameworks tradicionales todo el estado es administrado básicamente en la sesión HTTP, Seam provee una mayor granularidad de contextos de estado. La principal, quizás es el contexto conversacional, así como el asociado a procesos del negocio, con estos se logra un uso más eficiente de la memoria. Integra además el concepto de espacios de trabajo, permitiendo que el usuario tenga en varias ventanas del navegador actividades del negocio con contextos completamente aislados. Seam integra de forma

## Análisis y diseño del sistema

---

transparente la administración de procesos del negocio vía JBoss JBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones y complejas interacciones con el usuario.

El concepto de conversación permite definir funcionalidades realmente atómicas en las que se involucren varios pedidos al servidor donde los cambios hechos a las entidades solo son persistidos en la base de datos al final de conversación, gracias al contexto de persistencia extendido implementado por Seam y basado en Hibernate. Como interfaces para comunicarse con sistemas legados se establecen servicios web que serían la fachada para los componentes Seam reales.

El modelo es la representación de la información que maneja la aplicación. El modelo en si son los datos puros que puestos en el contexto del sistema provee de información al usuario o a la aplicación misma. Para el acceso a datos se usa la implementación de JPA de Hibernate 3.3, minimizando por un lado las configuraciones en XML sin chequeo de tipos y por otro lado usando los servicios del contenedor de EJB3 y/o los contextos de persistencias administrados por Seam, se elimina gran parte del código “infraestructural” en cuanto a transacciones, la transmisión del contexto de persistencia, etc. Además se pueden establecer validaciones gracias a los Hibernate Validators.

Algunas incumbencias son horizontales, se encuentran en todas las capas de la aplicación, la seguridad es una de ellas. En este caso, toda la autorización, desde el acceso a directorios, páginas, controles, opciones del menú, servicios del negocio, está basado en reglas. Lo que permite que ninguna de las “reglas del negocio” esté escrita como código en la aplicación y que el cambio de alguna de ellas, no requiera cambio alguno en el código, solo en la definición de alguna regla en un fichero de configuración. El Seam Security Framework permite todo esto gracias a su integración con el potente motor de reglas JBoss Rules.

### **3.2 Diseño**

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe como implementar el sistema. Debe ser suficientemente específico para que el sistema pueda ser implementado sin ambigüedades.

# Análisis y diseño del sistema

---

Sus objetivos fundamentales son:

- Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones.
- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes.
- Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Ser capaces de visualizar y reflexionar sobre el diseño utilizando una notación común.
- Crear una abstracción sin costuras de la implementación del sistema, en el sentido de que la implementación es un refinamiento directo del diseño que rellena lo existente sin cambiar la estructura. Esto permite la utilización de tecnologías como la generación de código y la ingeniería de ida y vuelta entre el diseño y la implementación.

## 3.2.1 Patrones de diseño

Entre los patrones de diseño que más se utilizan están los GRASP que son los patrones generales para la asignación de responsabilidades. De todos ellos se destacan principalmente por su utilización en el diseño los patrones:

- Experto: es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo.
- Creador: el patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que:

tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase.

- Alta cohesión: Expresa que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase.
- Bajo acoplamiento: Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

### 3.2.2 Patrones en la capa de datos

En la capa de datos, y para lograr una mayor flexibilidad y por tanto asimilar mejor los cambios futuros que se produzcan en la base de datos se utilizan los siguientes patrones:

- Active record: consiste en que un objeto envuelve una tupla de una estructura de datos de un recurso externo, como una fila en una base de datos, y adiciona alguna lógica del dominio a dicho objeto (27).
- Identity field: la idea fundamental de este patrón es que guarda el campo id de la base de datos en un objeto para mantener la relación entre el objeto cargado en memoria y la fila correspondiente en la base de datos.
- Foreign Key Mapping: consiste en mapear una asociación entre objetos por cada relación entre tablas por claves foráneas.
- Query Object: mantener un objeto que permita hacer consultas a la base de datos.
- Lazy load: el objeto que mapea una fila en la base de datos no necesariamente debe cargar toda la información desde un principio. Debe conocer la forma de obtenerla y cargarla solo cuando sea necesario.

### 3.2.3 Estructuración

En UML, un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

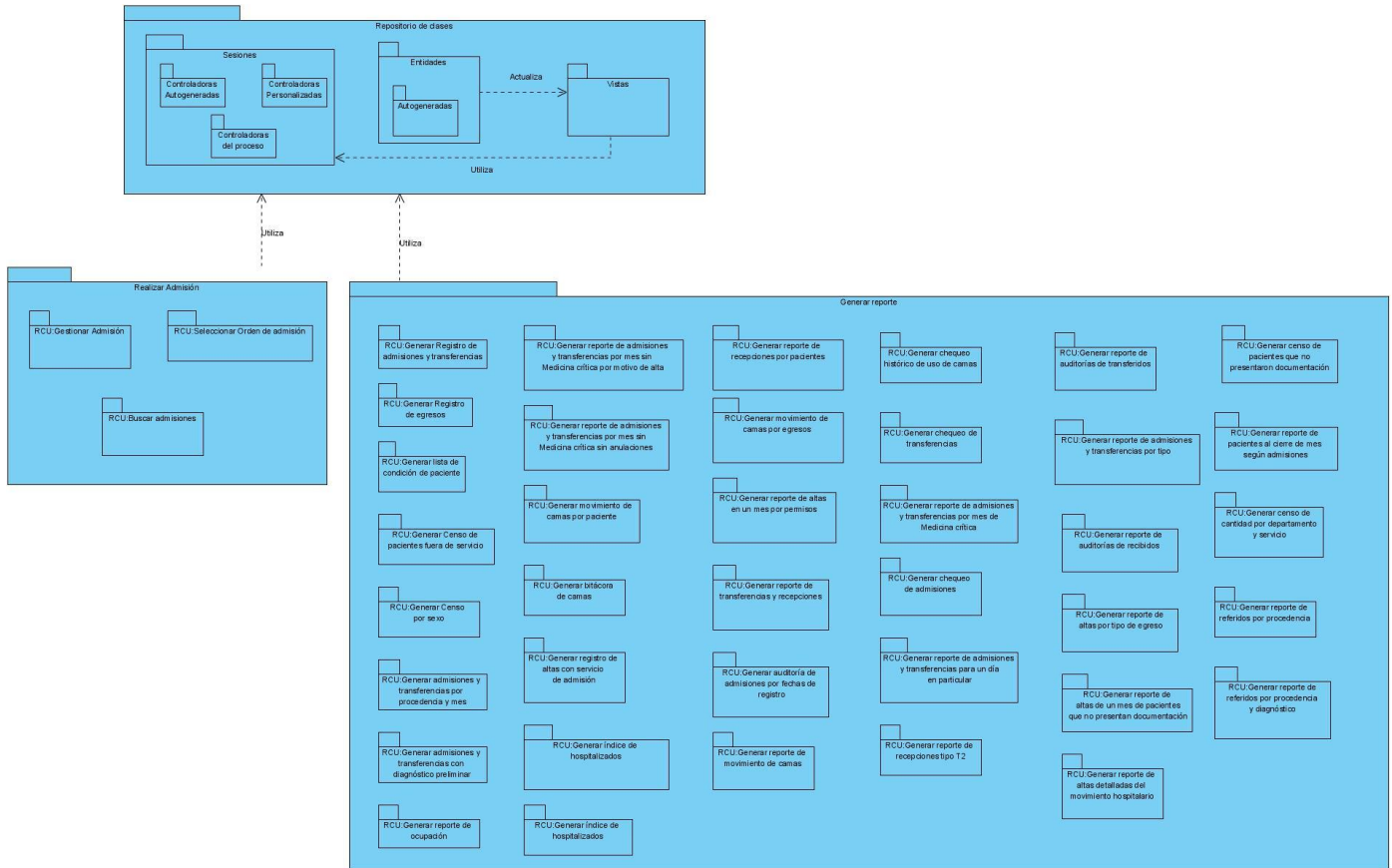
Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre ellos. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido.

Para el sistema propuesto se ha decidido organizar la realización de los casos de uso de forma que cumpla con los patrones de diseño y de arquitectura explicados previamente. Para ello se cuenta con tres paquetes: Sesiones, Entidades y Vista, que contendrán las clases controladoras, las entidades o modelo, y la vista o presentación respectivamente.

El paquete Sesiones contendrá a su vez tres paquetes que agruparán: las clases controladoras autogeneradas por las herramientas de generación de código, las personalizaciones de las clases autogeneradas y las clases controladoras que no son autogeneradas agrupadas por procesos y por casos de uso. El paquete Entidades contiene las entidades autogeneradas y el paquete Vista todas las interfaces de la aplicación.

# Análisis y diseño del sistema

## 2 Diagrama de paquetes



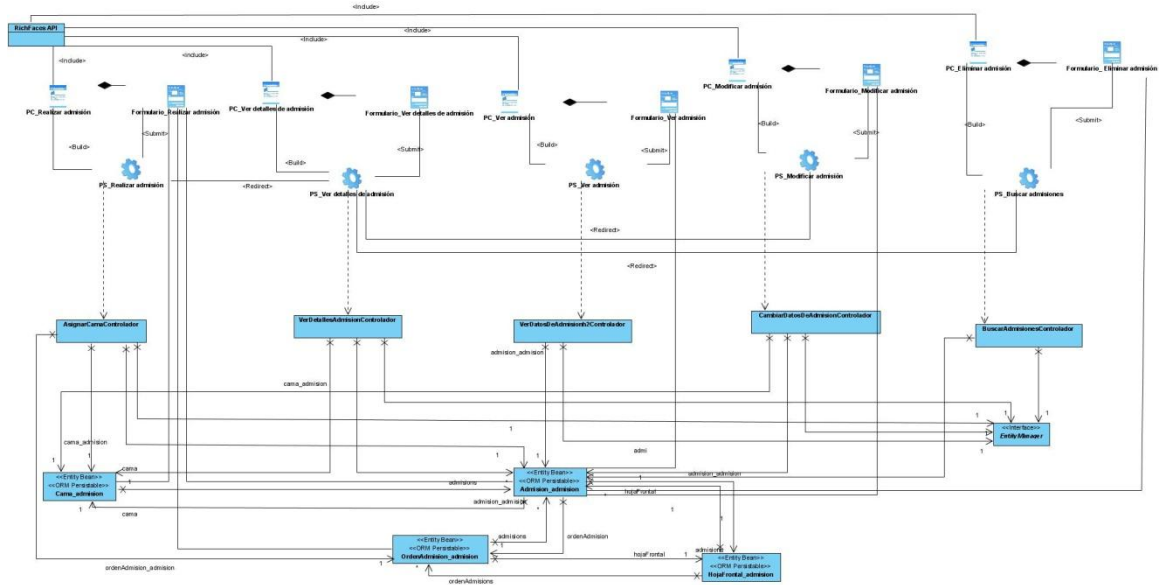
### 3.2.4 Diagramas de clases

El diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.



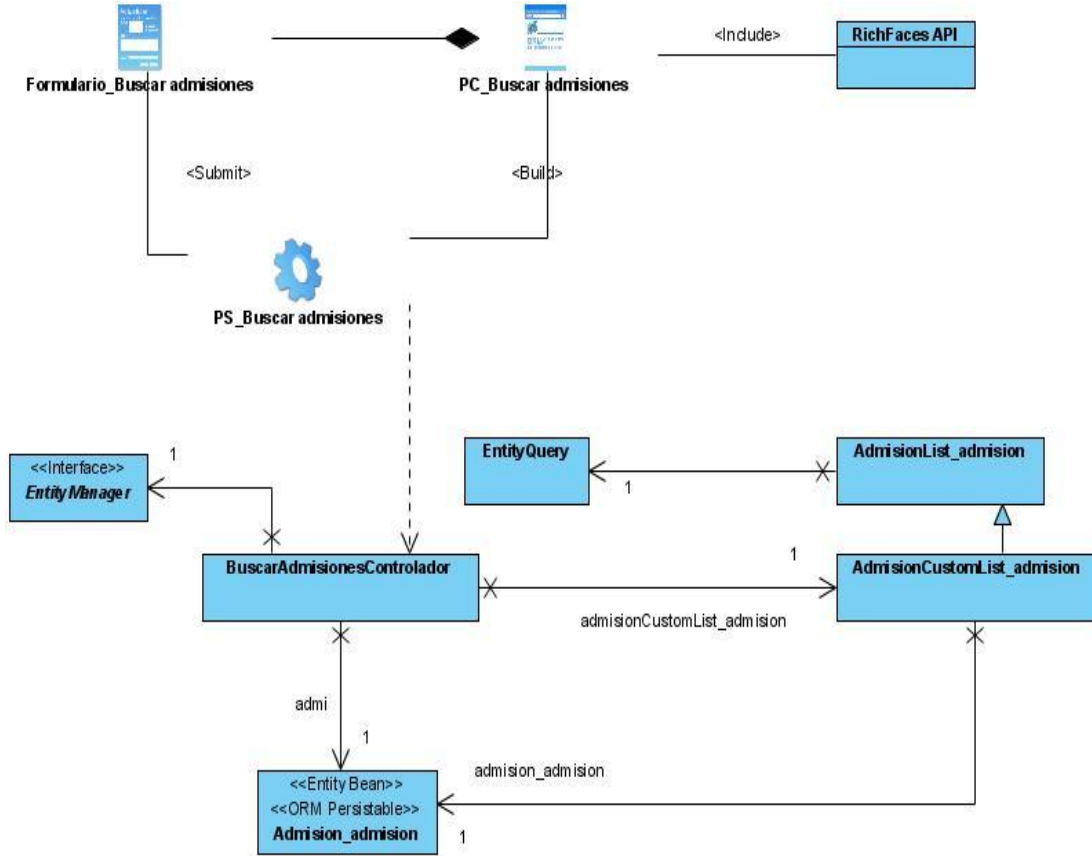
# Análisis y diseño del sistema

## 3 Realizar admisión

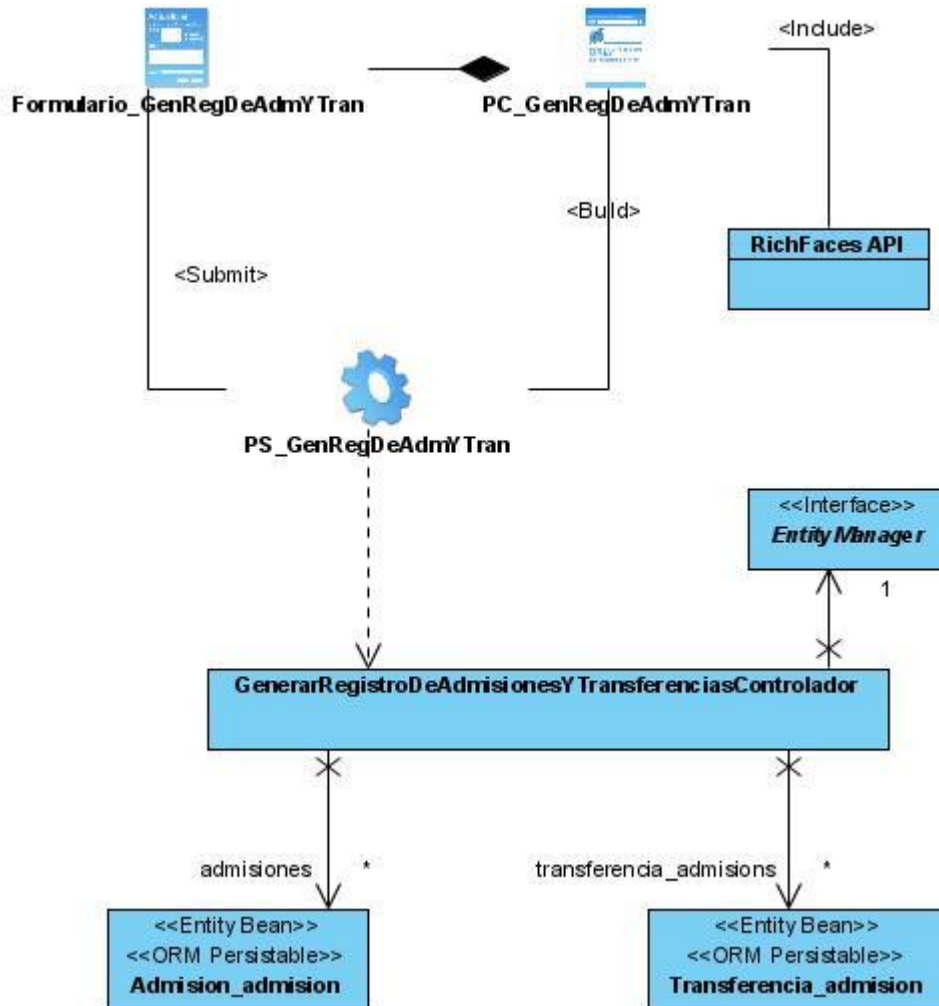


# Análisis y diseño del sistema

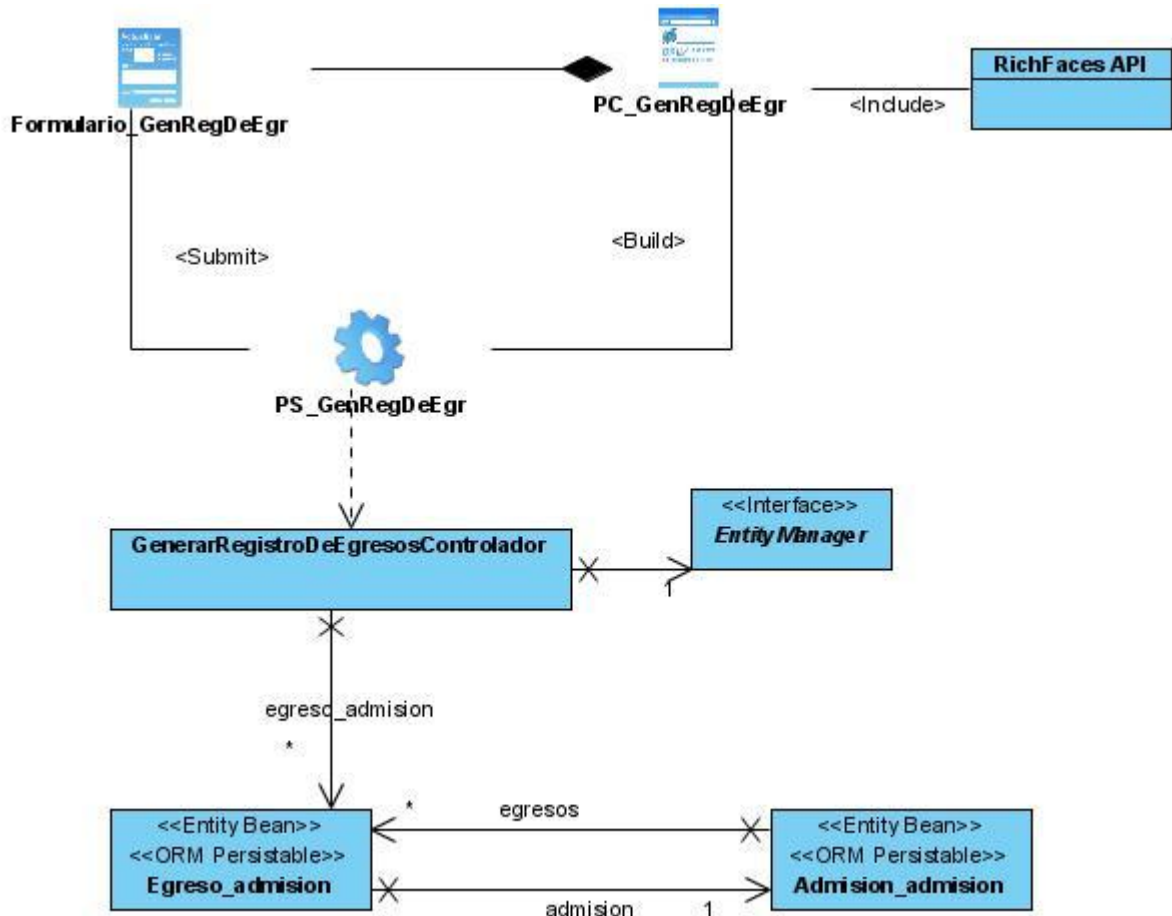
## 4 Buscar admisiones



## 5 Generar registro de admisiones y transferencias



## 6 Generar registro de egresos



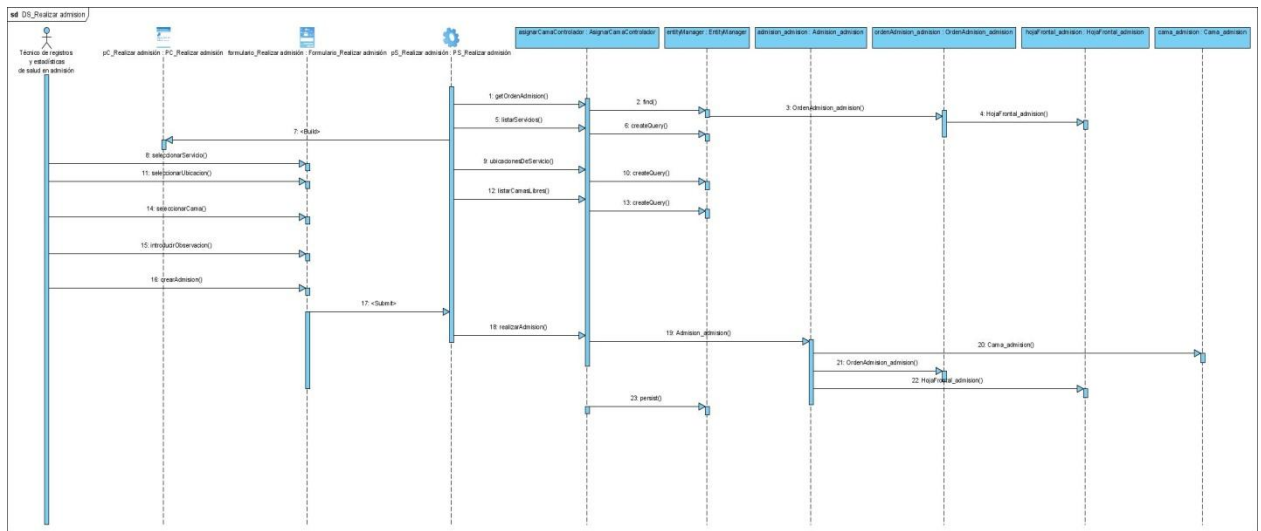
### 3.2.5 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concreta de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento.

# Análisis y diseño del sistema

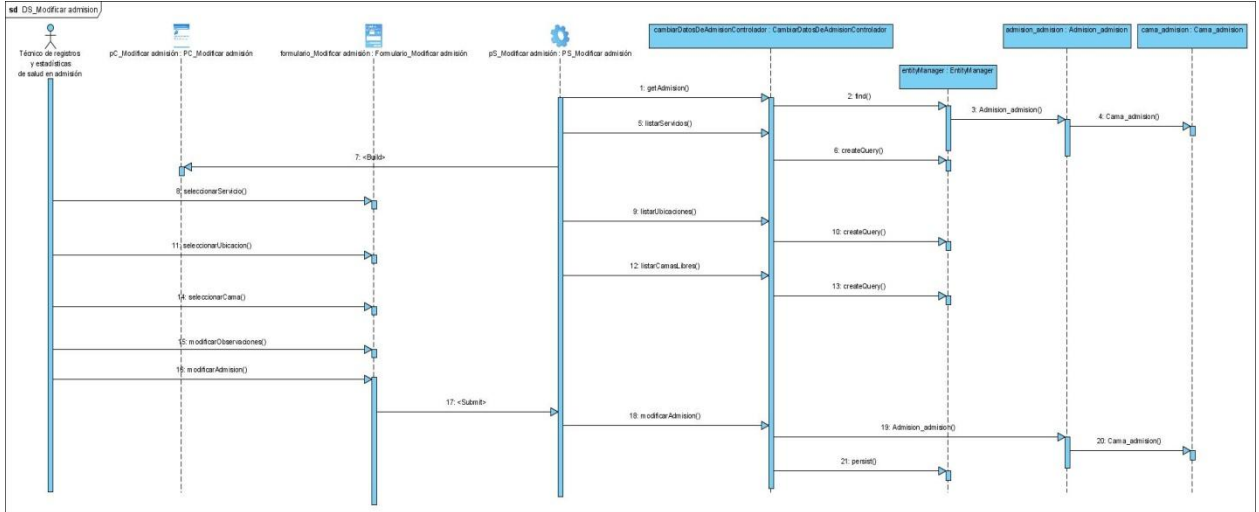
Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Exhibe los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario.

## 7 Realizar admisión

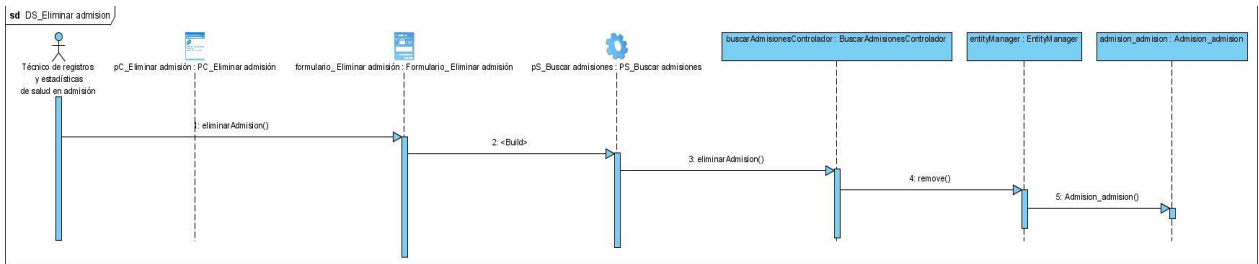


# Análisis y diseño del sistema

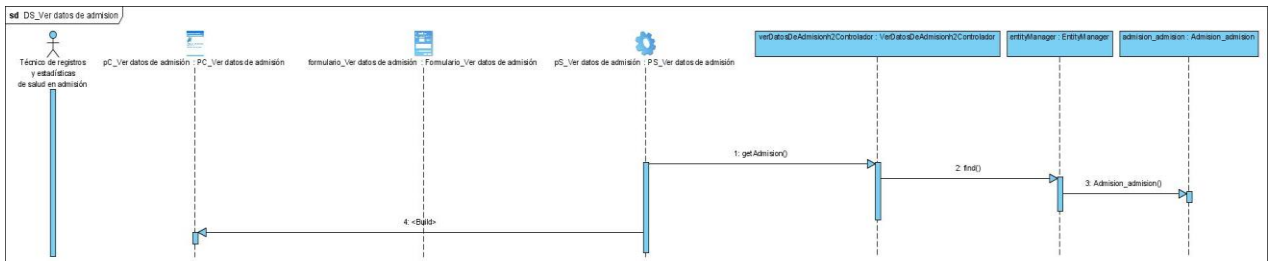
## 8 Modificar admisión



## 9 Eliminar admisión

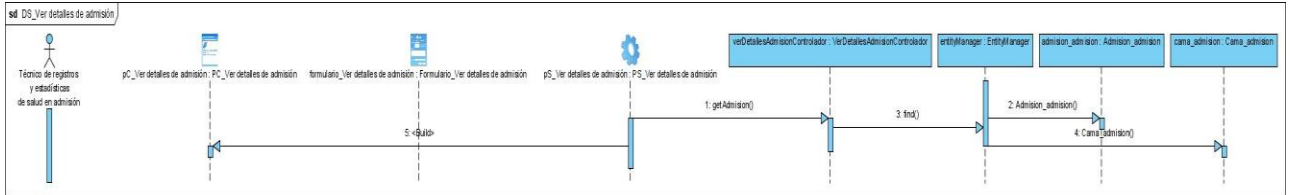


## 10 Ver datos de admisión

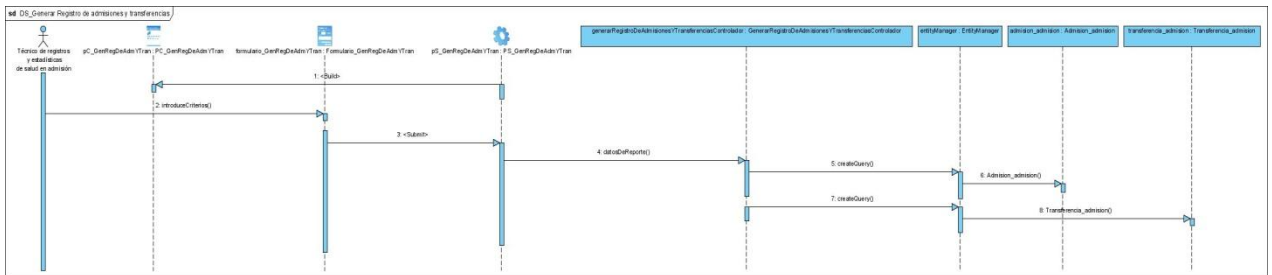


# Análisis y diseño del sistema

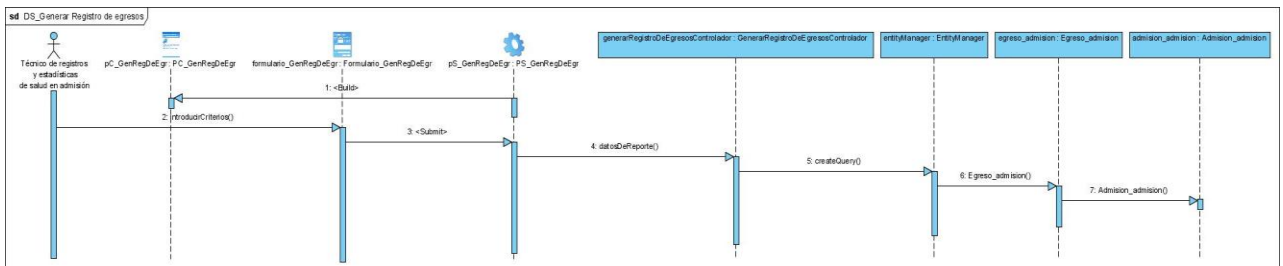
## 11 Ver detalles de admision



## 12 Generar registro de admisiones y transferencias



## 13 Generar registro de egresos



## Análisis y diseño del sistema

---

### 3.2.6 Descripción de las clases

Algunos elementos están presentes en todos los diagramas de clases pero teniendo en cuenta que sus funcionalidades son las mismas a continuación se brinda una descripción a grandes rasgos, de cada uno de ellos.

- Las páginas clientes son construidas a partir de la información que se desea brindar al usuario. Sirve como punto de entrada para la interacción de los usuarios con el sistema.
- Los formularios forman parte de las páginas clientes y se utilizan para solicitar la entrada de datos al usuario y enviarlos a las páginas servidoras
- Las páginas servidoras reciben y procesan los datos enviados por los formularios, construyen las páginas clientes a partir de la información que le es solicitada.

<b>Nombre:</b> AsignarCamaControlador	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
idOrden	Integer
servicio	String
ubicacion	String
fechaIngreso	Date
Observaciones	String
<b>Para cada responsabilidad:</b>	
Nombre:	listarServicios()
Descripción:	Devuelve los servicios que existen en la base de datos
Nombre:	listarCamasLibres()
Descripción:	Devuelve las camas libres que pertenecen a una ubicación en particular
Nombre:	ubicacionesDeServicio()
Descripción:	Retorna las ubicaciones que pertenecen a un servicio previamente seleccionado
Nombre:	realizarAdmision()
Descripción:	Ingresa a un paciente, para ello le asigna una cama de un servicio y cambia el estado de la cama a ocupada
Nombre:	edad()
Descripción:	Calcula la edad de un paciente a partir de su fecha de nacimiento y la fecha actual
Nombre:	diagnosticoPreliminar()
Descripción:	Obtiene el diagnóstico preliminar de un paciente



## Análisis y diseño del sistema

---

<b>Nombre:</b> CambiarDatosDeAdmisionControlador	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
servicio	String
ubicacion	String
<b>Para cada responsabilidad:</b>	
Nombre:	listarCamasLibres()
Descripción:	Devuelve las camas libres que pertenecen a una ubicación en particular
Nombre:	listarServicios()
Descripción:	Devuelve los servicios que existen en la base de datos
Nombre:	listarUbicaciones()
Descripción:	Obtiene las ubicaciones de un servicio
Nombre:	modificar()
Descripción:	Modifica la admisión
Nombre:	edad()
Descripción:	Calcula la edad de un paciente a partir de su fecha de nacimiento y la fecha actual
Nombre:	diagnosticoPreliminar()
Descripción:	Obtiene el diagnóstico preliminar de un paciente

<b>Nombre:</b> BuscarAdmisionesControlador	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
idAdmision	Integer
admisionCustomList_admision	
<b>Para cada responsabilidad:</b>	
Nombre:	listarServicios()
Descripción:	Devuelve los servicios que existen en la base de datos
Nombre:	listarUbicaciones()
Descripción:	Obtiene las ubicaciones de un servicio
Nombre:	listarCamas()
Descripción:	Devuelve todas las camas de una ubicación seleccionada
Nombre:	Eliminar()
Descripción:	Elimina la admision

<b>Nombre:</b> BuscarOrdenDeAdmisionControlador	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	listarSexo()
Descripción:	Devuelve los posibles valores para sexo en la base de datos

## Análisis y diseño del sistema

---

<b>Nombre:</b> VerDatosDeAdmisionControlador	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre	edad()
Descripción	Calcula la edad de un paciente a partir de su fecha de nacimiento y la fecha actual
Nombre	diagnosticoPreliminar()
Descripción	Obtiene el diagnóstico preliminar de un paciente

<b>Nombre:</b> GenerarRegistroDeEgresosControlador	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
fechaMenor	Date
fechaMayor	Date
servicio	String
<b>Para cada responsabilidad:</b>	
Nombre:	listarServicios()
Descripción:	Devuelve los servicios que existen en la base de datos
Nombre	datosDeReporte()
Descripción	Devuelve los egresos que cumplen con los criterios de búsqueda

<b>Nombre:</b> GenerarRegistroDeAdmisionesYTransferenciasControlador	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
fechaMenor	Date
fechaMayor	Date
serv	String
<b>Para cada responsabilidad:</b>	
Nombre:	listarServicios()
Descripción:	Devuelve los servicios que existen en la base de datos
Nombre	datosDeReporte()
Descripción	Devuelve las admisiones o transferencias que cumplen con los criterios de búsqueda

### **Conclusiones**

En este capítulo se ha descrito y fundamentado la arquitectura del sistema. Se explica brevemente como fueron utilizados los patrones Modelo Vista Controlador, GRASP y los de la capa de datos. Han sido presentadas las clases del diseño que participan en la realización de los casos de uso de la aplicación propuesta, los diagramas de secuencia del diseño y la descripción textual de cada una de las clases controladoras donde se implementan las principales funcionalidades.

# Implementación

---

## CAPÍTULO 4 IMPLEMENTACIÓN

En este capítulo se muestra el modelo de datos, la descripción de las tablas de la base de datos, además de explicar brevemente el diagrama de despliegue, en el cual se muestra la ubicación física de cada componente, los mismos se exponen más adelante en el diagrama de componentes, exponiendo temas polémicos del sistema como son los estándares de codificación y los tratamientos de errores aplicados.

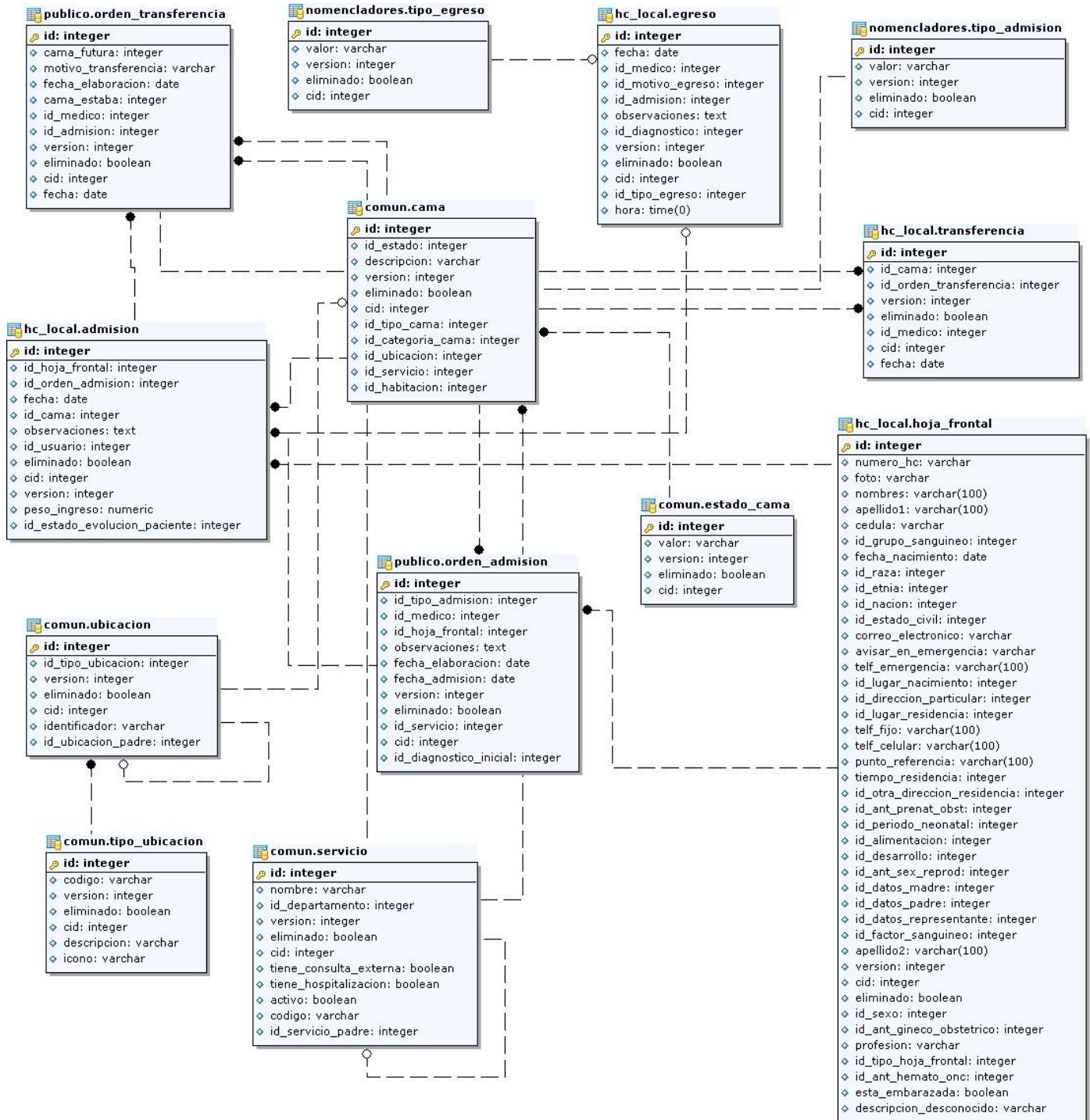
### **4.1 Modelo de datos**

Un modelo de datos es una serie de conceptos que puede utilizarse para describir un conjunto de datos y las operaciones para manipularlos. En un enfoque más amplio, un modelo de datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí.

Teniendo en cuenta las particularidades de los datos que se recolectan durante los procesos que se llevan a cabo en el área de admisión se ha definido el siguiente modelo de datos:

# Implementación

## 14 Modelo de datos



# Implementación

---

## 4.1.1 Descripción de las tablas

Existen atributos que están presentes en todas las tablas de la base de datos. Ellos son:

- **id:** identifica unívocamente cada tupla de la tabla.
- **eliminado:** la eliminación de los datos es lógica, este atributo especifica si un dato fue borrado o no, lo que permite mantener siempre disponible los datos clínicos del paciente.
- **cid:** almacena el identificador de la última acción realizada por un usuario para mantener las trazas del sistema.
- **version:** es utilizado por Hibernate para controlar las versiones de los datos.

<b>Nombre: Tipo_egreso</b>		
<b>Descripción:</b> Guarda los tipos de egresos, entre ellos defunción, por mejoría, etc.		
Atributo	Tipo	Descripción
valor	String	Valor del tipo de egreso
descripcion	String	Descripción del tipo de egreso

<b>Nombre: Admision</b>		
<b>Descripción:</b> Almacena todos los ingresos que se realizan en la institución		
Atributo	Tipo	Descripción
fecha	Date	Fecha de ingreso del paciente
observaciones	String	Observaciones que se quieran realizar
peso_ingreso	Decimal	Peso del paciente al momento del ingreso

## Implementación

<b>Nombre: Hoja_frontal</b>		
<b>Descripción:</b> Mantiene la información del paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
numero	String	Número de historia clínica
foto	String	Dirección a la foto del paciente
nombres	String	Nombre del paciente
apellido1	String	Primer apellido del paciente
Apellido2	String	Segundo apellido del paciente
cedula	String	Numero que identifica al paciente
fecha_nacimiento	Date	Fecha de nacimiento del paciente
correo_electronico	String	Dirección de correo electrónico del paciente
avisar_en_emergencia	String	Datos de la persona a avisar en caso de emergencia
telf._emergencia	String	Teléfono de la persona a avisar en caso de emergencia
telf._fijo	String	Número de teléfono del paciente
telf._celular	String	Número del teléfono celular del paciente
punto_referencia	String	Punto de referencia de la residencia
tiempo_residencia	Int	Tiempo de residencia
profesion	String	Profesión del paciente
esta_embarazada	Boolean	Si en paciente está embarazada o no
descripcion_desconocido	String	Descripción general de un paciente desconocido

<b>Nombre: Tipo_admision</b>		
<b>Descripción:</b> Mantiene los tipos de admisiones que se pueden realizar		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
valor	String	Contiene el tipo de admisión

<b>Nombre: Egreso</b>		
<b>Descripción:</b> Recolecta los egresos que se realizan en la institución		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
fecha	Date	Fecha en que se realiza el egreso
hora	Time	Hora en que se realiza el egreso
observaciones	String	Observaciones generales que se deseen especificar

# Implementación

<b>Nombre: Servicio</b>		
<b>Descripción:</b> Contiene los servicios de la institución		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nombre	String	Nombre del servicio
Tiene_consulta_externa	Boolean	Especifica si el servicio tiene consulta externa o no
Tiene_hospitalizacion	Boolean	Especifica si el servicio tiene hospitalización o no
codigo	String	Código que identifica al servicio
activo	Boolean	Indica si el servicio está activo o no

<b>Nombre: Orden_admision</b>		
<b>Descripción:</b> Almacena los datos de las ordenes de admisión		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
observaciones	String	Observaciones generales que se deseen realizar
Fecha_elaboracion	Date	Fecha en que es elaborada la orden de admisión
Fecha_admision	Date	Fecha en que el paciente debe ser ingresado

<b>Nombre: Orden_transferencia</b>		
<b>Descripción:</b> Guarda los datos de las ordenes de transferencia		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
motivo_transferencia	String	Motivo por el que se indica la transferencia al paciente
Fecha_elaboracion	Date	Fecha en que se elabora la orden de transferencia
fecha	Date	Fecha en que se debe realizar la transferencia

<b>Nombre: Ubicacion</b>		
<b>Descripción:</b> Recopila las ubicaciones de la institución		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
identificador	String	Identificador de la ubicación

<b>Nombre: Tipo_ubicacion</b>		
<b>Descripción:</b> Guarda los tipos de ubicaciones (salas, habitaciones, etc.)		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
codigo	String	Código para el tipo de ubicación
descripcion	String	Descripción del tipo de ubicación



# Implementación

---

<b>Nombre: Cama</b>		
<b>Descripción:</b> Mantiene todas las camas de la institución		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
descripcion	String	Descripción de la cama

<b>Nombre: Estado_cama</b>		
<b>Descripción:</b> Define los estados que puede tener una cama (ocupada, desocupada, etc.)		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
valor	String	Estado en que se encuentra la cama

<b>Nombre: Transferencia</b>		
<b>Descripción:</b> Recopila la fecha en que se realiza una transferencia		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
fecha	Date	Fecha en que se realiza la transferencia

## 4.2 Implementación

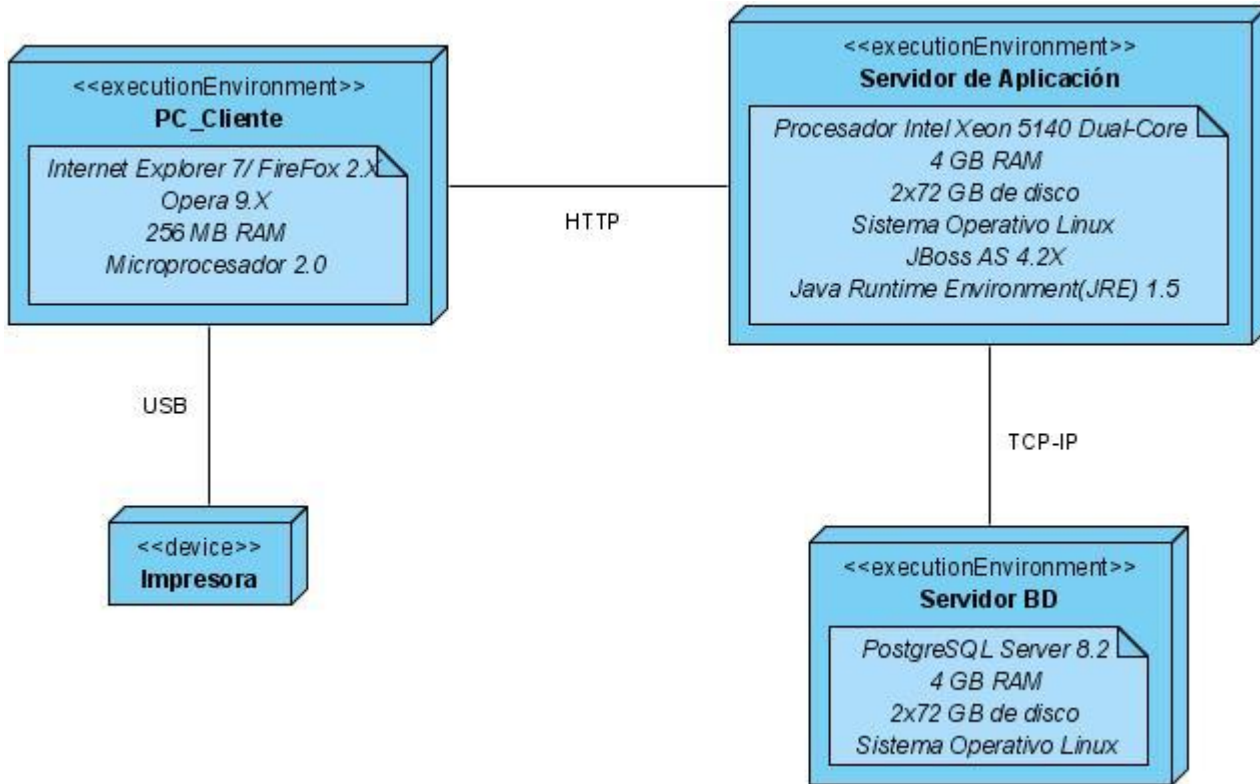
### 4.2.1 Diagrama de despliegue

El modelo de Despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

Para la implantación y la utilización de la aplicación en una institución hospitalaria el usuario debe conectarse a esta mediante una PC cliente utilizando un navegador web. Las peticiones por el protocolo HTTP serán procesadas por el servidor de aplicaciones que enviará la respuesta al cliente y en caso que sea necesario también hará peticiones mediante el protocolo TCP-IP al servidor de base de datos. El usuario puede imprimir reportes o gráficos desde la PC cliente utilizando una impresora.

# Implementación

## 15 Modelo de despliegue



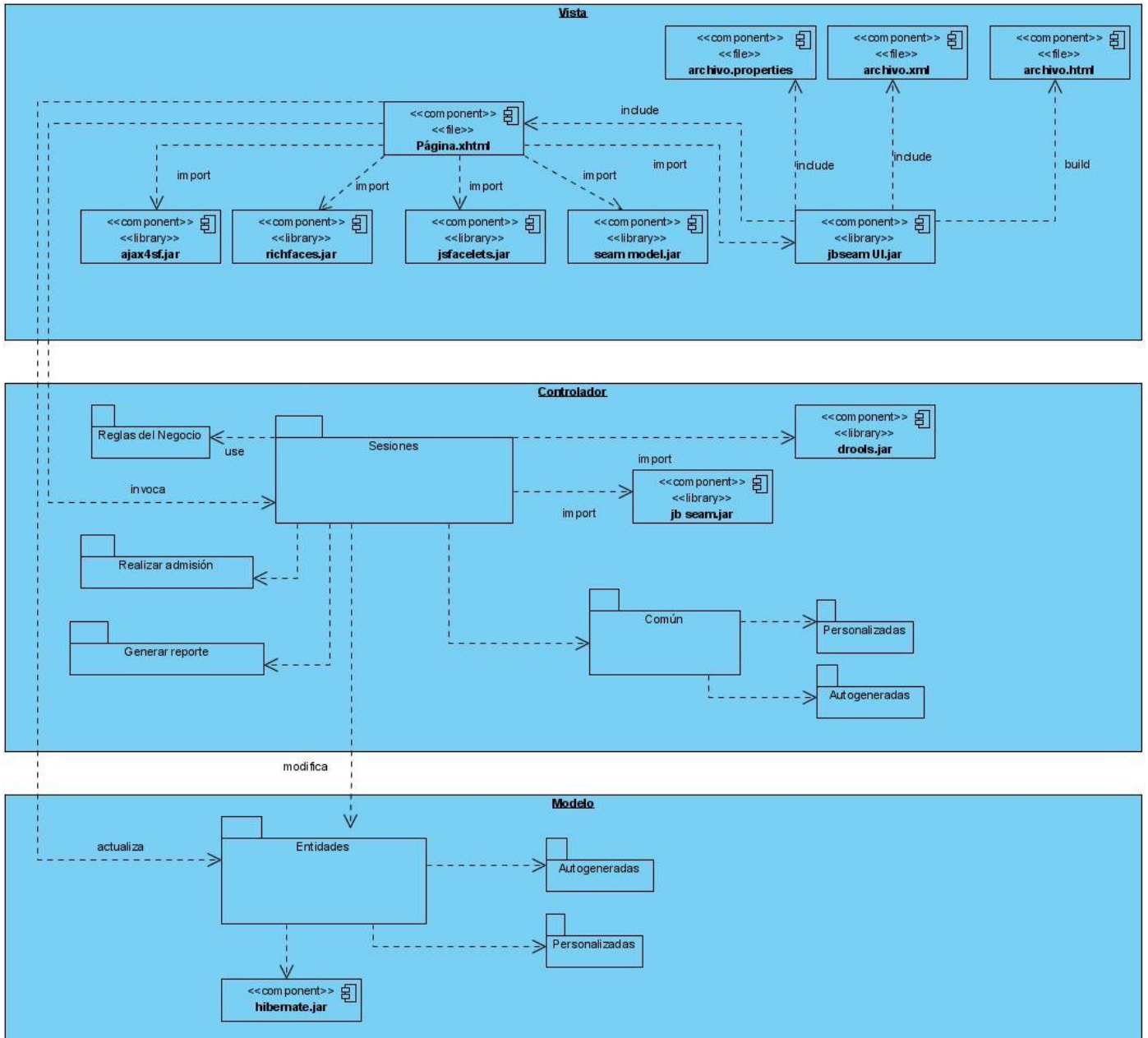
### 4.2.2 Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

# Implementación

Para mantener una trazabilidad directa desde el diagrama de paquetes de diseño al diagrama de componentes, se ha establecido un paquete de componentes por cada paquete del diseño.

## 16 Diagrama de componentes



## 4.3 Tratamiento de errores

Las excepciones son el mecanismo recomendado para la propagación de errores que se produzcan durante la ejecución de las aplicaciones (divisiones por cero, intentos de lectura de archivos dañados). Las mismas crean objetos con información acerca de ellas, que pertenecen a la clase `Exception` y son llamadas en la ejecución del programa en los puntos donde se encuentra un error, permitiendo que la ejecución del programa, aún en presencia de un error.

En el HIS se propone realizar el tratamiento de excepciones en todos los fragmentos de código donde pueda ocurrir un error, especialmente en los lugares donde se manipulan los datos son insertados o modificados en la base de datos y además en las validaciones de los datos que son insertados por los usuarios en la interfaz de usuario.

El control de la navegación, en caso de ocurrir una excepción que implique una redirección, se maneja mediante los `.pages.xml`, los mismos encargan de capturar globalmente las excepciones y ejecutar las instrucciones determinadas. Para el control de las demás excepciones es utilizado el componente `FacesMessages` del framework Seam el cual se encarga de mostrar los mensajes que se manejan a través del objeto `facesMessages` inyectado en las clases controladoras. El mismo trata los mensajes por tipo (error, alerta y notificación).

## 4.4 Estrategias de codificación. Estándares y estilos a utilizar

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Aunque la legibilidad y la mantenibilidad son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones.

Es fundamental mantener una buena indentación para lograr una estructura uniforme en los bloques de código así como para los diferentes niveles de anidamiento. Se recomienda dejar cuatro espacios en

# Implementación

---

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`.

Deben escribirse comentarios al principio de cada clase y método brindando una breve descripción de los propósitos generales de cada funcionalidad.

Los nombres de clases deben ser palabras completas, en mayúsculas y minúsculas, con la primera letra de cada palabra en mayúscula. Los nombres de clases deben ser simples y descriptivos, utilizando palabras completas y acrónimos o abreviaturas (a no ser que la abreviatura sea ampliamente conocida, como URL o HTML).

Los nombres de interfaces deberían seguir las mismas reglas indicadas para las clases.

Los métodos deberían ser verbos, en mayúsculas y minúsculas, con la primera letra en minúscula, y la primera letra de cada una de las palabras internas en mayúscula.

Todos los nombres de variables de instancia o de clase deben estar constituidos por palabras con la primera letra de la primera palabra en minúscula y la primera letra de las palabras internas en mayúscula.

Los nombres de variables deben ser cortos y significativos. La elección de un nombre de variable debe ser mnemotécnico, es decir, pensado para que un lector casual al verla comprenda su uso. Se deben evitar las variables de una sola letra, excepto en variables temporales de corto uso. Nombres comunes para este tipo de variables son: `i`, `j`, `k`, `m` y `n` para enteros; `c`, `d`, `y` e `e` para caracteres.

Los nombres de variables declaradas como constantes de clase y constantes ANSI, deberían escribirse siempre en mayúsculas, con las palabras internas separadas por el signo de subrayado ("`_`"). Las constantes ANSI se deben evitar en lo posible, para facilitar la depuración del código.

# Implementación

---

## **Conclusiones**

En el capítulo se detalla el diagrama de despliegue y componentes que muestran la distribución física del sistema y las dependencias lógicas entre los componentes de la aplicación. Fue descrita la estructura de tablas en la que se sustenta el modelo de datos de la aplicación, así como sus atributos y relaciones. Se estructuraron las clases del diseño en paquetes y subsistemas de implementación. Se describieron, además, los principios de diseño seguidos, concepción del tratamiento de errores y principios de codificación. En general, fue implementado el sistema en términos de componentes con el objetivo de dar solución a los requisitos especificados.

## Conclusiones generales

---

### CONCLUSIONES GENERALES

Durante el estudio realizado para elaborar el presente trabajo, se demostró que los sistemas estudiados no satisfacen las necesidades, que en materia de gestión de la información, tienen las instituciones hospitalarias. Además, durante el análisis de los procesos de negocio, se evidenció la necesidad de automatizar los procesos asociados al área de Admisión de las instituciones hospitalarias, con el objetivo de mejorar la calidad de los servicios que en dicho marco se brindan y lograr facilitar la gestión de la información.

Es debido a esto que fue implementado un sistema robusto y flexible, que requiere un mínimo de cambios para su adaptación a las necesidades de cualquier institución hospitalaria en particular. Con esta aplicación se informatizaron los procesos de negocio realizar admisión y generar reportes del módulo admisión que permitirán ingresar a los pacientes en los distintos servicios y obtener datos estadísticos para la toma de decisiones médicas y administrativas. Por lo que fue desarrollado el módulo Admisión del Sistema de Información Hospitalaria alas HIS, que facilita la gestión de información en esta área de las instituciones hospitalarias.

## Recomendaciones

---

### RECOMENDACIONES

Sobre el presente trabajo los autores recomiendan:

- Desarrollar los procesos relacionados con la gestión de la Historia clínica electrónica del paciente.
- Incorporar funcionalidades que permitan realizar comparaciones entre historias clínicas para minimizar la existencia de información duplicada del paciente.



## Referencias bibliográficas

---

### REFERENCIAS BIBLIOGRÁFICAS

1. OMS Organización Mundial de la Salud. [En línea] OMS. <http://www.who.int/es/>.
2. *Hospitales\_InscripcionAdmicion\_Conceptos*. 2008.
3. SIVSA. [En línea] SIVSA. <http://www.sivsa.com/>.
4. CARE2X Integrated Healthcare Environment. [En línea] <http://www.care2x.org/>.
5. C-DAC. [En línea] C-DAC. <http://www.cdac.in/html/his/sushrut.asp>.
6. **Zeldman, Jeffrey**. *Diseño con estándares web*. s.l. : Anaya.
7. Wikipedia. La enciclopedia libre. [En línea] [http://es.wikipedia.org/wiki/Aplicación\\_web](http://es.wikipedia.org/wiki/Aplicación_web).
8. **Shklar, Leon y Rosen, Richard**. *Web Application Architecture Principles, Protocols and Practices*. ISBN 0-471-48656-6.
9. **Reynoso, Carlos y Kiccillof, Nicolás**. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
10. **Smith, Michael**. *Java and Object-Oriented Language*. s.l. : McGraw-Hill, 2000.
11. **J. Homes, Barry y T. Joyce, Daniel**. *Object-Oriented programming with Java*.
12. **Becerril C., Francisco**. *Java a su alcance*. ISBN 970-10-1774-9.
13. **Geary, David y Horstmann, Cay**. *Core JavaServer™ Faces*. 2007. 978-0-13-173886-7.
14. **Red Hat**. *RichFaces Developer Guide*. 2007.
15. Seamframework.org. [En línea] Red Hat. <http://seamframework.org/>.
16. **Juntao Yuan, Michael and Heute, Thomas**. *JBoss Seam Simplicity and Power Beyond Java EE*. 2007. ISBN 0-13-134796-9.
17. Jboss Community. [En línea] <http://www.jboss.org/>.
18. JBoss. [En línea] Red Hat, 2009. <http://www.jboss.com/>.

## Referencias bibliográficas

---

19. Hibernate. [En línea] Red Hat. <https://www.hibernate.org/>.
20. *Hibernate Reference Documentation*.
21. **Bauer, Cristian y King, Gavin.** *Java Persistence with Hibernate*. 2005. ISBN 1-932394-88-5.
22. PostgreSQL. [En línea] PostgreSQL Global Development Group. <http://www.postgresql.org/>.
23. **The PostgreSQL Global Development Group.** *PostgreSQL 8.3.6 Documentation*. 2008.
24. **Cerami, Ethan.** *Web Services Essentials Distributed Applications with XML-RPC, SOAP, UDDI & WSDL*. 2002. ISBN: 0-596-00224-6.
25. **Akif, Mohammad, y otros.** *Java y XML*. s.l. : Anaya.
26. **Jacobson, Ivan, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. 2004.
27. **Fowler, Martin.** *Patterns of Enterprise Application Architecture*.

## Bibliografía

---

### BIBLIOGRAFÍA

**Akif, Mohammad, y otros.** *Java y XML*. s.l. : Anaya.

**Allen, Dan.** *Seam in Action*. s.l. : Manning Publications, 2008.

**Bauer, Cristian y King, Gavin.** *Java Persistence with Hibernate*. 2005. ISBN 1-932394-88-5.

**Becerril C., Francisco.** *Java a su alcance*. ISBN 970-10-1774-9.

CARE2X Integrated Healthcare Environment. [En línea] <http://www.care2x.org/>.

C-DAC. [En línea] C-DAC. <http://www.cdac.in/html/his/sushrut.asp>.

**Cerami, Ethan.** *Web Services Essentials Distributed Applications with XML-RPC, SOAP, UDDI & WSDL*. 2002. ISBN: 0-596-00224-6.

Drools Documentation Library. [En línea] JBoss Community.  
<http://www.jboss.org/drools/documentation.html>.

**Farley, Jim.** *Practical JBoss® Seam Projects*. 2007.

**Fowler, Martin.** *Patterns of Enterprise Application Architecture*.

**Geary, David y Horstmann, Cay.** *Core JavaServer™ Faces*. 2007. 978-0-13-173886-7.

Health Level 7. [En línea] HL7. <http://www.hl7.org/>.

Health Level Seven Spain. [En línea] <http://www.hl7spain.org/VerPagina.asp?IDPage=0>.

Hibernate. [En línea] Red Hat. <https://www.hibernate.org/>.

*Hibernate Reference Documentation*.

*Hospitales\_InscripcionAdmicion\_Conceptos*. 2008.

**Jacobson, Ivan, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. 2004.

## Bibliografía

---

**Jamae, Javid y Johnson, Peter.** *JBoss in Action*. s.l. : Manning Publications, enero, 2009. ISBN: 1933988029.

JavaHispano. [En línea] <http://www.javahispano.org/>.

JBPM Documentation Library. [En línea] JBoss. [http://www.jboss.org/jbossjbpm/jbpm\\_documentation/](http://www.jboss.org/jbossjbpm/jbpm_documentation/).

Jboss Community. [En línea] <http://www.jboss.org/>.

JBoss. [En línea] Red Hat, 2009. <http://www.jboss.com/>.

**JBOSS.** *Seam - Contextual Components A Framework for Enterprise Java*.

**J. Homes, Barry y T. Joyce, Daniel.** *Object-Oriented programming with Java*.

**Juntao Yuan, Michael and Heute, Thomas.** *JBoss Seam Simplicity and Power Beyond Java EE*. 2007. ISBN 0-13-134796-9.

**Larman, Craig.** *UML y patrones. Introducción al análisis y diseño orientado a objetos*. 2004.

OMG Object Management Group. [En línea] Object Management Group, Inc. <http://www.bpmn.org/>.

OMS Organización Mundial de la Salud. [En línea] OMS. <http://www.who.int/es/>.

**Pérez Zurita, Paúl.** *IH-SW-DR-080 ALAS-HIS\_Visión*. 2008.

PostgreSQL. [En línea] PostgreSQL Global Development Group. <http://www.postgresql.org/>.

**Red Hat.** *Ajax4jsf Developer Guide*. 2007.

**Red Hat.** *RichFaces Developer Guide*. 2007.

**Reynoso, Carlos y Kiccillof, Nicolás.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.

RichFaces developer guide. [En línea] <http://www.jboss.org/file-access/default/members/jbossrichfaces/freezone/docs/devguide/en/html/index.html>.

RichFaces live demo. [En línea] <http://livedemo.exadel.com/richfaces-demo/index.jsp>.

## Bibliografía

---

Seamframework.org. [En línea] Red Hat. <http://seamframework.org/>.

**Shklar, Leon y Rosen, Richard.** *Web Application Architecture Principles, Protocols and Practices*. ISBN 0-471-48656-6.

SIGHO Sistema de Información para la Gerencia Hospitalaria. [En línea] <http://www.sigho.gob.mx/quees.htm>.

SIVSA. [En línea] SIVSA. <http://www.sivsa.com/>.

**Smith, Michael.** *Java and Object-Oriented Language*. s.l. : McGraw-Hill, 2000.

**S. Pressman, Roger.** *Ingeniería de software, un enfoque práctico*. 2005.

**Stephen A. White, PhD y Miers, Derek.** *BPMN Modeling and Reference Guide. Understanding and Using BPMN*. 2008. ISBN-13: 978-0977752720.

**Sun microsystems.** Sun España. [En línea] <http://es.sun.com/>.

**The PostgreSQL Global Development Group.** *PostgreSQL 8.3.6 Documentation*. 2008.

*UI development with JavaServer Faces*.

**Velázquez Carralero, Alejandro Mario.** *IH-SW-DR-091 ALAS-HIS\_Documento de Arquitectura del Sistema*. 2008.

Wikipedia. La enciclopedia libre. [En línea] [http://es.wikipedia.org/wiki/Aplicación\\_web](http://es.wikipedia.org/wiki/Aplicación_web).

**Zeldman, Jeffrey.** *Diseño con estándares web*. s.l. : Anaya.

# Glosario de términos

---

## GLOSARIO DE TÉRMINOS

### **API**

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

### **Bean**

Es un componente software que tiene la particularidad de ser reutilizable.

### **Framework**

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

### **HTML**

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

### **HTTP**

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

### **JavaScript**

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

### **Java Server Pages (JSP)**

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

### **LPT**

Puerto paralelo para conectar periféricos a una computadora.

## Glosario de términos

---

### **Plain Old Java Object (pojo)**

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

### **TCP/IP**

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

### **USB**

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

### **Web**

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

### **XML**

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