

Universidad de las Ciencias Informáticas
Facultad 7



**Título: Análisis y diseño del módulo Agenda Médica del
Sistema de Información Hospitalaria alas HIS**

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

Autor: Yojan Tang Veitia

Tutores: Ing. Mainoldis Fuentes Suárez
Ing. Arianna Devesa Navarro

Ciudad de la Habana, Enero de 2010
“Año 52 de la Revolución”

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los ___ días del mes de ___ del año ____.

Yojan Tang Veitia

DATOS DE CONTACTO

Ing. Mainoldis Fuentes Suárez (mfuentes@uci.cu): Instructor graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor de la Facultad #7. Ha impartido las asignaturas Introducción a la programación, Programación I y II, e Inteligencia Artificial. Se ha desempeñado como líder de módulo en el proyecto de gestión hospitalaria, como especialista principal de gestión de configuración y jefe de asignatura de Inteligencia Artificial.

Ing. Arianna Devesa Navarro (adevesa@uci.cu): Instructor recién graduado en el año 2009 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor vinculado a la Facultad 7 y miembro del Departamento de Sistema de Gestión Hospitalaria.

En las instituciones hospitalarias se llevan a cabo procesos de planificación de actividades, las cuales deben ser cumplidas por el personal de las mismas. Este proceso se ve a menudo afectado por la falta de comunicación entre las áreas existentes en los centros hospitalarios que de una forma u otra planifican dichas actividades. Con el fin de posibilitar una planificación óptima de estas tareas surge la Agenda Médica la cual se encarga de posibilitar la visualización de la misma. En el mundo existen numerosos sistemas automatizados con este fin. Estos productos de software son altamente costosos y en su mayoría se basan en tecnologías y herramientas propietarias, por lo que no brindan una solución factible y estándar a estas instituciones. Por dichas razones el objetivo de la investigación es diseñar el módulo Agenda Médica del Sistema de Información Hospitalaria alas HIS que posibilite la visualización centralizada de los procesos de planificación de actividades dentro de los centros hospitalarios así como el control y seguimiento de los mismos.

Para llevar a cabo el sistema anteriormente descrito se propone utilizar el Proceso Unificado de Desarrollo y estará basado en tecnologías libres, multiplataforma y sobre una arquitectura en capas. Utilizará Java como lenguaje de programación e implementará el patrón de arquitectura Modelo Vista Controlador. Como Sistema de Gestión de Bases de Datos será utilizado PostgreSQL y como servidor de aplicaciones el JBoss Server. Para obtener una interfaz visual moderna y realizar eficientemente las peticiones al servidor, se utilizarán las librerías JBoss RichFaces.

Tabla de contenido

Introducción.....	1
Capítulo 1: Fundamentación Teórica	5
1.1 <i>Sistemas de salud pública</i>	5
1.2 <i>Sistemas de Información Hospitalaria</i>	6
1.3 <i>Sistemas existentes vinculados al campo de acción</i>	6
1.4 <i>Tendencias y tecnologías actuales a considerar para el proceso de desarrollo de software</i>	10
1.5 <i>Tecnologías propuestas para el proceso de desarrollo.....</i>	12
1.6 <i>Herramientas utilizadas</i>	20
Capítulo 2: Características del sistema	22
2.1 <i>Flujo actual de los procesos involucrados en el campo de acción</i>	22
2.2 <i>Objeto de automatización.....</i>	24
2.3 <i>Modelo de dominio</i>	24
2.4 <i>Conceptos relacionados con el dominio</i>	25
2.5 <i>Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración.....</i>	33
2.6 <i>Propuesta del sistema</i>	34
2.7 <i>Modelo de casos de uso del sistema.....</i>	34
Capítulo 3: Diseño del sistema.....	41
3.1 <i>Modelo de Diseño</i>	41
3.2 <i>Diagramas de clases del diseño.....</i>	46
3.3 <i>Diagramas de secuencia</i>	48
3.4 <i>Descripción de clases del diseño.....</i>	53
3.5 <i>Modelo de datos.....</i>	59
Conclusiones	62
Recomendaciones.....	63
Referencias bibliográficas	64
Bibliografía.....	67

Glosario de términos.....71

INTRODUCCIÓN

En el transcurso de los últimos años las Tecnologías de la Información y las Comunicaciones (TIC) han alcanzado un desarrollo vertiginoso en prácticamente todas las esferas de la sociedad. Estas abarcan cada vez más procesos y se integran a nuevas ramas de la vida cotidiana. Lo que se debe a una de sus principales ventajas: la optimización considerable en cuanto al tiempo de las tareas y actividades una vez que son automatizadas.

Es por ello que muchas instituciones han integrado las TIC a su desarrollo, ya que les ha permitido un marcado incremento en su eficiencia. La esfera de la salud no se encuentra ajena a estos cambios por lo que ha integrado progresiva y sistemáticamente estas nuevas tecnologías, donde obtiene grandes beneficios en la calidad de sus servicios. Se cuenta, de esta forma, con métodos novedosos para la gestión administrativa en consultas, en los hospitales y centros de investigación biomédica. De esta manera se dispone de sistemas automatizados que apoyen al diagnóstico, tratamiento y rehabilitación de los problemas de salud.

Los mismos son los llamados Sistemas de Información Hospitalario (HIS por sus siglas en inglés), que satisfacen problemas como el procesamiento de datos médicos así como el almacenamiento de la información resultante de los procesos hospitalarios médico-administrativos de las distintas áreas que abarca. Además, constituyen un apoyo para las actividades en los niveles operativos, estratégicos y tácticos de cualquier institución hospitalaria.

Permiten 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 en dependencia del área o servicio para el cual se requiera, dando lugar a la retroalimentación de la calidad de la atención de los servicios de salud.

Es por ello que hay empresas que se dedican a su implementación y comercialización en todo el mundo, obtienen de esta forma grandes ingresos monetarios puesto que los HIS ostentan precios excesivamente elevados.

No obstante, dadas las sustanciosas ventajas que ofrecen los HIS al sector hospitalario, en países europeos se encuentran acoplados a muchas instituciones de este tipo, que brindan y gestionan sus servicios exitosamente. En los Estados Unidos de América también se pueden encontrar HIS, que despliegan exitosamente sus funciones en centros de atención a la salud.

En Latinoamérica la informatización del sector hospitalario es una necesidad. Para ello se debe llevar cabo la actualización y modernización de las instituciones existentes desde el punto de vista tecnológico así como la construcción de nuevas instituciones que llevarán a cabo procesos clínicamente más complejos, por lo que su equipamiento tecnológico deberá ser avanzado. De manera que la tecnología suministrada a los centros hospitalarios sea equitativa a los servicios que brindan.

Para la explotación de los recursos tecnológicos existentes se necesita disponer de recursos humanos capacitados para su operatividad. A su vez, se debe contar con soluciones informáticas que posibiliten la gestión de la información de los procesos asistenciales con la calidad, eficiencia y eficacia que estos demandan y que sean adaptables a cualquier institución de salud de cualquier nivel.

Actualmente en la Universidad de las Ciencias Informáticas, específicamente en la Facultad 7 se está encuentra en desarrollo uno de estos sistemas de información hospitalaria, denominado alas HIS. El cual cuenta con numerosos módulos, uno de ellos, es Agenda Médica. En él, es posible visualizar la planificación de manera centralizada de las tareas que realiza el personal médico de un centro, a partir de la información obtenida a través de las distintas áreas del mismo, así como el seguimiento y control de las mismas.

El proceso de la planificación centralizada se ve afectado por varias causas. Cada área dentro de un centro hospitalario lleva de forma independiente la planificación de sus actividades. El personal encargado de realizar esta función, solo tiene acceso a la planificación que existe en su área o servicio, por lo que desconoce si algún miembro del personal médico de su servicio tiene planificadas actividades dentro de otras áreas. Esto trae como consecuencia que a menudo haya incongruencias entre las planificaciones de distintas áreas.

Una vez que se decide planificar una actividad donde intervenga personal médico de diferentes áreas, como no se dispone de la planificación de las áreas implicadas es muy difícil llevar a cabo este proceso sin cometer errores. Por otra parte, existen integrantes del personal médico que deben impartir clases a estudiantes de medicina tanto dentro como fuera de la instalación hospitalaria, por lo que estas actividades también deben tomarse en cuenta a la hora de planificar el tiempo de dichos miembros. Estas funciones también pueden coincidir con otras previamente planificadas.

Por todo lo anteriormente descrito se designa como **problema a resolver** ¿Cómo llevar a cabo la planificación centralizada de las tareas del personal médico en las instituciones hospitalarias?

El **objeto de estudio** de la investigación en curso es el proceso de gestión de la información en las instituciones hospitalarias. De manera que, el **campo de acción** estaría enmarcado en el proceso de planificación en las instituciones hospitalarias. Para dar solución al problema anteriormente identificado se propone el siguiente **objetivo general**: Diseñar el módulo Agenda Médica del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de la planificación en las instituciones hospitalarias. Para dar cumplimiento al objetivo planteado se definen las siguientes **tareas de la investigación**:

- Evaluar las tendencias actuales en el mundo de los Sistemas de Información Hospitalaria.
- Analizar los principales procesos asociados a las planificaciones en las instituciones hospitalarias.

- Analizar todos los procesos de negocio asociados al módulo Agenda Médica del Sistema de Información Hospitalaria alas HIS.
- Definir las funcionalidades del módulo Agenda Médica del Sistema de Información Hospitalaria alas HIS.
- Asimilar la arquitectura definida por el Departamento de Sistema de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
- Obtener mediante el Proceso Unificado de Desarrollo los artefactos correspondientes a los flujos de trabajo de “Modelo de Negocio”, “Gestión de Requerimientos” y “Análisis y Diseño”.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se expone la base teórica y conceptual para el posterior desarrollo del módulo Agenda Médica del Sistema de Información Hospitalaria alas HIS. Se exponen conceptos significativos relacionados con el campo de acción y el dominio del problema, así como antecedentes existentes de sistemas o subsistemas similares. Se analizan las tecnologías, metodologías y herramientas que deben estar presentes en el proceso de desarrollo del software y con las cuales se llevará a cabo el proceso de desarrollo.

1.1 Sistemas de salud pública

Atención primaria de la Salud (APS): La atención primaria de salud es la asistencia sanitaria esencial, accesible a todos los individuos y familias de la comunidad a través de medios aceptables para ellos, con su plena participación y a un costo asequible para la comunidad y el país. Es el núcleo del sistema de salud del país y forma parte integral del desarrollo socioeconómico general de la comunidad. Constituye el primer contacto del usuario con el sistema sanitario. Por ella se accede a niveles asistenciales más complejos. Atendida por los Centro de Asistencia Primaria (CAP), debería solucionar el 70 u 80% de los problemas asistenciales cotidianos. Incluye el médico de cabecera, pediatra, asistencia social, radiología, laboratorio y urgencias. [1]

Asistencia secundaria: Segundo escalón asistencial, que incluye operaciones sencillas, exploraciones y otras tareas de policlínica que no implican larga hospitalización, aunque sí en algunos casos tratamiento en camas hospitalarias de este nivel A y B, según el personal y la tecnología requeridos por la enfermedad o atención médica domiciliaria. Este nivel da cobertura a cerca del 15 % de los problemas de salud. [2]

Atención terciaria: El nivel terciario debe abarcar alrededor del 5 % de los problemas de salud. Trata enfermedades que requieren hospitalización con atención especializada y de alta tecnología en camas de nivel C. [3]

1.2 Sistemas de Información Hospitalaria

Un Sistema de Información es un conjunto de instrucciones organizadas, sistematizadas y lógicas que se relacionan entre sí por medio de un lenguaje informático con el fin de obtener información, analizarla, relacionarla y generar nueva información para satisfacer las necesidades de las áreas administrativas, operativas de una organización en general.

Un sistema de información hospitalario está orientado a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria. Así como a la optimización de los recursos humanos y materiales.

Todo sistema de información hospitalaria genera reportes e informes en dependencia del área o servicio para el cual se requiera, dando lugar a la retroalimentación de la calidad de la atención de los servicios de salud. La principal función de un HIS es la de apoyar las actividades en los niveles operativos, tácticos y estratégicos dentro de un Hospital.[4]

1.3 Sistemas existentes vinculados al campo de acción

Los Sistemas de Información Hospitalaria (HIS por sus siglas en inglés) cuentan con la presencia de varios módulos, muchos de los cuáles se identifican con áreas específicas de atención dentro de las instituciones hospitalarias. En el capítulo anterior quedó evidenciado que la Agenda Médica no es un área física dentro de una institución hospitalaria. Aún por la importancia de sus funciones forma parte de sistemas de información hospitalaria como uno de sus módulos o se desempeña como un complemento de aplicaciones automatizadas destinadas al sector de la salud. Puesto que con su ayuda se organiza de manera eficiente tanto el tiempo del personal

Capítulo 1: Fundamentación Teórica

médico en cada una de las tareas que desempeñan, así como cualquier actividad que se desee planificar en una institución hospitalaria.

Sistema Informático			Licencia		Tipo de Aplicación	
	Multiplataforma	País Productor	Propietario	Libre	Web	Escritorio
Agenda Médica DEMOSYS	no	Guatemala	X			X
Agenda Médica Electrónica	si	Chile	X		X	
DENDRITAS	si	México	X		X	
Metis-Ontho (His-Ris)	si	Uruguay	X		X	

Tabla 1.1 Sistemas de Información Hospitalaria existentes.

Características

Agenda Médica DEMOSYS: Sistema que se encarga de tramitar la planificación de citas médicas. Es utilizada por especialistas médicos para llevar una planificación de las citas médicas que deben realizar en un período de tiempo. Maneja las citas en forma de notas, y sus dimensiones son adaptables según las especificaciones del usuario. Es un complemento del sistema Clínica Médica Demosys. Es un software creado y distribuido por Demosys, una empresa productora de software, guatemalteca, que se ha desarrollado en el mercado del software desde 1998. Es una aplicación de escritorio, por lo que el proceso de instalación se debe aplicar de forma manual en cada una de las estaciones de trabajo, lo cual resulta un

Capítulo 1: Fundamentación Teórica

inconveniente desde el punto de vista de accesibilidad ya que no se puede acceder al sistema sin una previa instalación.

Agenda Médica Electrónica: Este sistema permite la creación de citas a pacientes por Internet, así como asignar horarios para diferentes sucursales y médicos. Posee una interfaz de fácil uso e interacción.

El mismo es capaz de asignar hasta cuatro mil citas diarias, busca en cada caso la disponibilidad de tiempo antes de agregar una nueva cita. Permite al usuario visualizar las características del procedimiento médico para el cual ha realizado una cita. Es un sistema desarrollado en .Net de Microsoft, por el grupo MedSalud.

SIIS Dendritas (Sistema Integral de Información en Salud): El sistema tiene como punto de partida y fuente primaria de información al individuo, toma en consideración que este es el principal centro de actividades de cualquier sistema de salud, de tal manera que a partir de la atención médica preventiva o curativa dirigida al paciente, se genera de manera automática la mayor parte de los registros necesarios para orientar el funcionamiento de los sistemas de salud, incluyendo los procesos de facturación y cobro asociados a la prestación del servicio de salud. Permite la gestión completa de la información relacionada con las personas de la población objetivo, mediante el registro de todos los datos socio- demográfico y las características de condiciones de vida de los individuos que se requieren para la identificación plena del paciente en cualquiera de los módulos que se utilice para generar información.

Este SIIS creado por HDS (Sistemas Digitales de Salud por sus siglas en inglés) presenta el módulo Agenda Médica, el cuál gestiona de manera integral la agenda del equipo asociado a la Institución Prestadora de Salud, apoyado en mecanismos de administración de consultorios y jornadas de trabajo. Cuenta con las funcionalidades siguientes:

- Creación y gestión de agenda para cada profesional, especialidad, consultorio o quirófano.
- Sistema de búsqueda por medio de agrupación de profesionales por especialidad.

Capítulo 1: Fundamentación Teórica

- Permite planificar las citas y usa un sistema de TURNOS (asignación de la cita por continuidad del horario disponible) o CITAS (asignación de la cita en el horario de conveniencia disponible) en las agendas seleccionadas.
- Sistema de re-programación de citas médicas, mediante la ubicación fechas y médicos disponibles.
- Permite reservar turnos o días para actividades diferentes tales como: reuniones, capacitaciones y labores administrativas.

La aplicación está programada en Java y disponible en ambiente WEB multiplataforma. Maneja estándares internacionales de información para diagnósticos, procedimientos (CIE-10, CIE-9, CPT), y medicamentos. Maneja estándares internacionales de comunicación, TCP/IP, SMTP. Maneja el estándar internacional de intercambio y acceso a datos HL7.

Metis HIS - Ontho RIS: His-Ris desarrollado por Apreful software. Entre los numerosos módulos que presenta podemos encontrar el módulo Agenda Médica, donde cada agenda puede ser asociada a un profesional (médico, especialista) y a un recurso del centro asistencial (tomógrafo, quirófano, rayos). Es configurable por el usuario. Metis HIS – Ontho RIS fue desarrollado con GeneXus y Pattern PXTTools. Fue generado en Java 3 capas. Los reportes dinámicos y datawarehouse fueron desarrollados con GXplorer y utiliza la prestación FULL WEB. El producto ha sido probado y funciona perfectamente en los navegadores Internet Explorer y Mozilla Firefox.

Luego de haber realizado un análisis con profundidad sobre las principales características de los sistemas encontrados como parte de los resultados de la investigación realizada, se puede concluir que a pesar de que algunos de estos presentan funcionalidades de relevancia, cuentan con las desventajas de ser muy costosos, están basados en su mayoría en tecnologías y herramientas de carácter propietario y no están integrados con el Sistema de Información Hospitalaria ala HIS. Es por ello que ninguno constituye una solución viable y factible a la problemática existente. De manera que se hace necesaria la obtención de un sistema que permita la visualización centralizada de los procesos de planificación de actividades a cumplir por los miembros del personal médico dentro de las instituciones hospitalarias.

1.4 Tendencias y tecnologías actuales a considerar para el proceso de desarrollo de software

Una vez analizado el estado del arte de los Sistemas de Información Hospitalaria, así como de otras aplicaciones automatizadas dedicadas al sector de la salud, que poseen o se complementan con el producto a desarrollar, se determina que las tecnologías a utilizar, para el desarrollo del sistema deben ser libres y multiplataforma. En lo que resta del capítulo se hará alusión a definiciones de las tecnologías escogidas, así como sus ventajas.

Patrones de arquitectura y diseño

Los patrones arquitectónicos expresan una organización estructural fundamental o esquema para los sistemas de software. Proporcionan un conjunto de subsistemas predefinidos, especifica sus responsabilidades, e incluye las normas y directrices para la organización de las relaciones entre ellos. Es por ello, que el logro del mejoramiento y la calidad del diseño de un sistema, depende en gran medida de la utilización de patrones que proporcionen esquemas para refinar subsistemas o componentes de un sistema. Estos son los denominados patrones de diseño.

Un patrón de diseño expresa esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas automatizados, es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades. Es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, un proyecto o estructura de implementación que logra una finalidad determinada, son en fin, una manera más práctica de describir ciertos aspectos de la organización de un programa.[5]

Arquitectura en capas:

Patrón Modelo-Vista-Controlador (MVC)

La arquitectura en capas es un estilo cuyo objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, para ello se definen capas o niveles.

La capa de presentación reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios. Estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menús, gráficos y elementos multimedia en general. La capa del Dominio de la Aplicación reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican. Esta capa también recibe el nombre de capa de la Lógica de la Aplicación. La capa del Repositorio reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes, por lo que también se le denomina capa de Bases de Datos.[6]

El patrón Modelo – Vista – Controlador permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. Esto proporciona múltiples vistas sobre un mismo modelo de datos. Este patrón se usa frecuentemente en aplicaciones web donde se utilicen diferentes interfaces de usuario y el código que provee los datos a la página es dinámico. Los tres elementos esenciales de este patrón son los siguientes: [7]

Modelo: Es el núcleo de la aplicación. Este mantiene el estado y los datos que la aplicación representa. Cuando se produzcan cambios significativos en el modelo a partir de especificaciones de cambio formuladas normalmente por el controlador, se actualiza todas las vistas asociadas al modelo.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

Controlador: Recibe o interpreta la información que proporciona el usuario y se inicia una respuesta. Hace llamadas de los objetos del modelo, provoca cambios en este y en caso de ser necesario en la vista.

Ventajas:

- **Vistas múltiples:** La aplicación puede mostrar el estado del modelo en una variedad de formas, y la creación y diseño de una manera escalable y modular. Esto es posible gracias al aislamiento o independencia que presenta el modelo con respecto a la vista.
- **Adaptable a cambios:** Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Desventajas:

- Costo de actualizaciones frecuentes.
- Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una gran cantidad de requerimientos de actualización.

1.5 Tecnologías propuestas para el proceso de desarrollo

Vista

JavaServer Faces (JSF)

Framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. **JSF** usa Java Server Pages (JSP) como la tecnología que

permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

RichFaces

RichFaces es un framework de código abierto que añade capacidad Ajax dentro de aplicaciones JSF existentes sin recurrir a JavaScript. RichFaces incluye ciclo de vida, validaciones, conversores y la gestión de recursos estáticos y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax (Ajax4jsf) y un alto grado de personalización del “look-and-feel” que puede ser fácilmente incorporado dentro de las aplicaciones JSF. Permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes listos para usar, altamente configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia, lo que mejora grandemente la experiencia de usuario.[8][9]

Seam UI

Serie de controles JSF altamente integrables con JBoss Seam. Están dirigidos a complementar los controles JSF incorporados, y los controles de otras bibliotecas externas.

Facelets

Framework de código abierto bajo la licencia de Apache y de tecnología alternativa de controlador de vista de la Java Server Faces. Soporta todos los componentes de interfaz de usuario JSF y construye su propio árbol de componentes, lo que refleja el punto de vista de una aplicación JSF, por lo que aporta mayor libertad a los desarrolladores pues permite definir vistas JSF mediante la utilización de plantillas del tipo HTML, reduce así el código innecesario para agregar componentes en la vista y que no necesariamente sea un contenedor web. Las principales ventajas de Facelets son:

- Construcción de interfaces basadas en plantillas.
- Rápida creación de componentes por composición.

- Fácil creación de funciones y librerías de componentes.

Lenguaje de Marcado de Hipertexto Extensible (XHTML)

Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0, combina la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar, una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados, etc.

Extended EL

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

Controlador

JBoss Seam

JBoss Seam 2.0 es un framework que integra y unifica los distintos standards de la plataforma Java EE 5.0. Puede trabajar con todos ellos y sigue el mismo modelo de programación. Simplificar al máximo el desarrollo de aplicaciones, y basa el diseño en Plain Old Java Objects (POJOs) con anotaciones. Estos componentes se usan desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa. El núcleo principal de Seam

está formado por las especificaciones, Enterprise JavaBeans 3 (EJB3) y JavaServer Faces (JSF). Seam también se integra perfectamente con otros frameworks como: RichFaces, ICEFaces (soportan AjaX) MyFaces, Hibernate y Spring. Realiza aportes 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 y aporta un nuevo concepto, la administración de espacios de trabajo. Esta permite al usuario tener en varias pestañas o ventanas del navegador actividades del negocio con contextos completamente aislados.[10][11][12]

Modelo

Java Persistence API (JPA)

Java Persistence API (JPA) es una especificación para la persistencia de objetos Java en una base de datos relacional. Proporciona un estándar para gestionar datos relacionales en aplicaciones Java SE o Java EE, de forma que además se simplifique el desarrollo de la persistencia de datos. Permite al desarrollador fácilmente mapear, almacenar, actualizar y recuperar datos de tablas de bases de datos en objetos Java y viceversa.[13]

Enterprise JavaBeans (EJB3)

Arquitectura componente del lado del servidor para la plataforma Java. Permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos. Incorpora el estándar JPA como el principal API de persistencia para aplicaciones EJB3.

Hibernate

Framework open source que provee una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. Permite mapear o relacionar el

diseño de datos a ficheros XML. Desde estos podremos generar el código de nuestros objetos persistentes en clases Java y también crear bases de datos independientemente del entorno escogido.[14]

Tecnologías y metodologías horizontales

Proceso Unificado de Desarrollo (RUP)

RUP por su nombre en inglés, Rational Unified Process, es un proceso de desarrollo de software que junto a UML constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Es en realidad un refinamiento realizado por Rational Software del más genérico proceso unificado.

Sus principales características son:

- Define una manera de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar prácticas en Ingeniería de Software de la mejor forma posible.
- Presenta un desarrollo iterativo.
- Hace posible la administración de requisitos.
- Hace uso de arquitectura basada en componentes.
- Posibilita el control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

RUP está caracterizado por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Además incluye artefactos que son los productos tangibles obtenidos mediante su aplicación y trabajadores que no son más que las personas que juegan un rol determinado para la obtención de estos artefactos.

RUP divide el proceso de desarrollo en ciclos, se obtiene así un producto al culminar cada ciclo, estos se dividen en 4 fases que finalizan con un hito donde se debe tomar una decisión importante.[15]

- **Fase de Inicio:** Tiene como objetivo determinar la visión del proyecto.
- **Fase de Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Fase de Construcción:** El objetivo es obtener la capacidad operacional inicial.
- **Fase de Transición:** Obtener el release del producto.

Lenguaje Unificado de Modelado (UML por sus siglas en inglés)

El lenguaje unificado de modelado (UML, por su nombre en inglés), es el lenguaje de modelado de sistemas de software más conocido y utilizado actualmente. Es un lenguaje gráfico que permite visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como: procesos de negocios, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Dentro de sus características se pueden mencionar algunas tales como:

- Tecnología orientada a objetos.
- Viabilidad en la corrección de errores.
- Desarrollo iterativo e incremental.
- Participación del cliente en todas las etapas del proyecto.
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).

Capítulo 1: Fundamentación Teórica

- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

Este puede ser utilizado en sistemas desarrollados en varios lenguajes de implementación y plataformas, incluyendo lenguajes de programación y bases de datos. Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como: objetos, clase, componentes y nodos. Está especialmente pensado para apoyar un estilo de desarrollo iterativo e incremental.[16]

PostgreSQL

Es un sistema gestor de bases de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD (Berkeley Software Distribution). Esta licencia tiene menos restricciones en comparación con otras como la GPL y está muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. Puede ser ejecutado sobre la mayoría de los sistemas operativos que existen hoy en día. Posee protección de transacciones u operaciones de Atomicidad, Consistencia, Aislamiento, Durabilidad (ACID, por sus siglas en inglés).

Soporta nativos para los lenguajes más populares del medio: PHP, C, C++, Perl, Java, Python, Ruby, además de su propio lenguaje PL/pgSQL. Soporta todas las características de una base de datos profesional (triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por los usuarios, vistas, vistas materializadas).

Capítulo 1: Fundamentación Teórica

Además soporta tipos de datos de SQL92 y SQL99. Soporta el protocolo de comunicación encriptado por SSL. Además de extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos. Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases de datos, se elimina la necesidad del uso de bloqueos explícitos. Esto se denomina alta concurrencia y es una de las características esenciales de PostgreSQL. Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST que presenta. [18][19]

Java

Lenguaje de programación multiplataforma que adopta el paradigma orientado a objetos, diseñado por la compañía Sun Microsystems Inc., salió a la luz en 1991, Licencia de software: GNU GPL / Java Community Process.

Incorpora parte de la sintaxis de C, C++, Smalltalk, 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. Tiene una arquitectura neutral, portátil y robusta. Posee las estructuras mínimas de un lenguaje de programación tradicional, sin añadir ninguna estructura más. Una de las características más importantes es que los programas “ejecutables”, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos. La plataforma Java consta de las siguientes partes:

- El lenguaje de programación, Java.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.

- El API Java, una biblioteca estándar para el lenguaje.

Con Java se pueden programar aplicaciones web dinámicas, con acceso a bases de datos, utiliza XML, con cualquier tipo de conexión de red entre cualquier sistema. Este lenguaje es utilizado de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación. [20]

JBoss Server

Servidor de aplicaciones Java más utilizado actualmente en el mercado. Cientos de profesionales y desarrolladores de código abierto han contribuido a su creación y desarrollo. Es una plataforma certificada que cumple con las especificaciones de J2EE para el despliegue y desarrollo de aplicaciones empresariales Java, aplicaciones web, y portales. Provee servicios extendidos de almacenamiento de datos en memoria y de manera persistente. Permite la integración de todas las tecnologías y herramientas utilizadas por JBoss Seam. Es actualizado e integrado constantemente con lo último del estado del arte de las aplicaciones web.[21][22]

1.6 Herramientas utilizadas

Por todo lo anteriormente expuesto, como resultado de la investigación realizada se designa como herramientas a conformar el Ambiente de Desarrollo del modulo Agenda Medica del Sistema Hospitalario alas HIS.

- **Visual Paradigm 6.0** como herramienta CASE (Computer – Aided Software Engineering), dado que es una herramienta UML profesional, además soporta el ciclo de vida completo del desarrollo de software. Y acoplado a este UML 2.0 para la creación de diagramas de casos de uso del flujo de requerimientos, así como los correspondientes a los de los flujos de diseño e implementación.

Capítulo 1: Fundamentación Teórica

- **pgAdmin III** para administrar la base de datos, debido a que es un ambiente gratuito, relacional, muy poderoso y puede ser ejecutado sobre la mayoría de los sistemas operativos.
- **Eclipse**, un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) que cuenta con la posibilidad de añadir nuevas funcionalidades al editor, mediante nuevos módulos, lo que es beneficioso para la utilización de Jboss Tools, que no es más que un conjunto de herramientas que posibilitan el desarrollo desde el IDE Eclipse con RichFaces, Seam e Hibernate, además de realizar la administración y configuración del servidor Jboss AS.

En este capítulo se realizó un estudio sobre los sistemas de información hospitalaria así como de aplicaciones informáticas destinadas al sector de la salud que presentan el módulo Agenda Médica existentes. A raíz de lo cual se demostró que no ofrecen una solución estándar o viable a fin de eliminar la problemática existente referente a la planificación de forma centralizada. Además, se enumeraron y describieron las tecnologías escogidas para el desarrollo del producto y las ventajas de cada una de ellas.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El Sistema Nacional de Salud (SNS) incluye, dentro de sus niveles de atención, como parte de la atención secundaria, los hospitales. En ellos, muchas de las actividades que realiza el personal médico son planificadas por una persona que es la encargada de esta tarea en un área determinada del hospital. Estas planificaciones se realizan de forma independiente una de otra, por lo que cada encargado de realizar la planificación dentro de su área no conoce la disposición real de tiempo que tienen los médicos a los que les asigna una tarea dentro de su área.

Por esta razón muchas veces puede ocurrir incongruencia entre planificaciones de diferentes áreas, incluso puede coincidir una actividad dentro de un área determinada con una reunión a la que deba asistir parte de su personal médico. Con el principal propósito de mitigar estos problemas, en este capítulo se propone la elaboración de un sistema informático capaz de contener la planificación de un centro hospitalario de manera centralizada.

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

En cuanto a planificación se refiere podemos definir los siguientes procesos:

En primer lugar la planificación de las guardias médicas dentro del área de Hospitalización, la cual es responsabilidad del Jefe de Servicio. Este tiene a su cargo la asignación de horarios de guardias médicas para los médicos especialistas, residentes e internos del área. Este proceso consiste en determinar o definir el horario, especificar en cada caso hora de inicio, fin, meses del año y días de la semana en que los miembros del personal médico deben presentarse a la guardia médica.

Por otra parte encontramos la planificación referente a los equipos de anestesiólogos y de los quirófanos dentro del área de Bloque Quirúrgico. La planificación de los equipos de anestesiólogos que asistirán a una intervención quirúrgica es responsabilidad de la secretaria de anestesiología. Una vez que esta recibe el plan quirúrgico y consulta la rotación de estos grupos dentro de los distintos servicios en los cuales están especializados, y realiza la distribución

Capítulo 2: Características del Sistema

pertinente. En caso de que existan inconvenientes con dicha rotación, ella es la encargada de realizar las modificaciones adecuadas. Una vez que la secretaria conforma los equipos de anestesiólogos por los diferentes servicios envía toda esta información a la secretaria de quirófanos y completa así este proceso de planificación.

A su vez el Jefe de quirófanos es el encargado de llevar a cabo la planificación de los mismos. Debe determinar la disponibilidad de los quirófanos, para ello consulta y analiza los datos de cada uno de ellos. Concluido este proceso busca los servicios quirúrgicos del hospital que serán asignados a los quirófanos disponibles, que cuentan con el equipamiento necesario. Una vez realizada esta asignación o administración de los quirófanos, el Jefe de Quirófanos envía esta distribución, a los cirujanos de los distintos servicios quirúrgicos, para que tengan en cuenta esta información a la hora de solicitar las intervenciones quirúrgicas programadas.

En el área de Consulta Externa, el Jefe de Servicio es el encargado de planificar el tiempo de los médicos, así como al tipo de consulta que deben realizar. Estas pueden ser consulta de primera, de control, interconsulta o de triaje. Para ello el Jefe de Servicio debe determinar o definir el horario, en el cual debe especificar la hora de inicio y fin, meses el año y días de la semana en que los miembros del personal médico deben ofrecer dichas consultas. Este período de consultas puede contener un horario de trabajo en días excepcionales (días feriados o no laborables), en este caso el Jefe de Servicio se encargará de realizar los cambios pertinentes a dicho horario a fin de mitigar estas imprecisiones.

Otro proceso que se encuentra en el sistema, tiene lugar en el área de Enfermería. Donde el coordinador se encarga de planificar el horario del personal de enfermería. Mediante el sistema el coordinador es capaz de planificar el servicio que va a ser asistido por cualquier equipo del personal de enfermería, así como el período de tiempo que durará esta asistencia. Define para ello fecha de inicio y de fin, días de la semana y meses del año que trabajará el equipo, además de los turnos que realizará (turno de trabajo de la mañana o de la tarde, o los definidos por la institución). El coordinador tiene la posibilidad de eliminar enfermeros de un equipo previamente elaborado.

2.2 Objeto de automatización

Como resultado de los procesos descritos anteriormente, se generan una serie de planificaciones en formato digital, ya que todos estos procesos se realizan a través de una aplicación automatizada. Por la importancia que se confiere a los procesos que se mencionan y explican a continuación y con el propósito de llevar a cabo una planificación de forma centralizada a fin de evitar el solapamiento de actividades programadas por las distintas áreas de un centro hospitalario, se ha determinado la necesidad de su automatización.

Se deben automatizar todos los procesos referentes a la centralización de los procesos del Sistema de Información Hospitalario alas HIS.

A raíz de identificar los principales procesos que apoyan una exitosa planificación de las actividades del personal médico de un centro hospitalario de manera centralizada, surge la necesidad de crear un sistema que automatice estos procesos. Debe concebirse una aplicación a fin con los conocimientos del personal que se encargará de gestionar toda la información del sistema. Para esto el software debe ser desarrollarse de manera que permita su total entendimiento, sea de fácil manejo y con una interfaz gráfica amigable, capaz de guiar a sus usuarios hacia la resolución exitosa de las dificultades que puedan presentar.

2.3 Modelo de dominio

Un Modelo de Dominio es un artefacto de la disciplina de análisis. Es una representación de los conceptos de importancia en el área de la aplicación, así como de las relaciones entre estos, construido con las reglas de UML simplificadas. Es presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. Se utiliza para capturar y expresar el entendimiento ganado en un área bajo análisis, como paso previo al diseño de un sistema. El modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. Convencionalmente la construcción del Modelo de Dominio es responsabilidad del Analista de

Capítulo 2: Características del Sistema

Sistema, quien ejecuta esta tarea con ayuda de la Visión del Sistema y del Glosario de Requisitos.

2.4 Conceptos relacionados con el dominio

Tarea: Clasificación que reciben aquellas actividades laborales y/o personales que son creadas por los propios usuarios. De manera que se denomina tarea a toda aquella actividad que los usuarios podrán crear, modificar, y eliminar.

Evento: Clasificación que reciben aquellas actividades del sistema que son creadas en las diferentes áreas de atención a los pacientes, dentro de los centros hospitalarios. De manera que se denomina evento a toda aquella actividad que los usuarios solo pueden consultar.

Funcionario: Tipo de usuario que tiene acceso a la Agenda Médica. Se corresponde con todos los miembros del personal médico existente en una institución hospitalaria que tienen actividades planificadas en las diferentes áreas de dicha institución. Pueden gestionar tareas en su beneficio propio, y consultar eventos que le son previamente planificados.

Gestor de la agenda médica: Tipo de usuario que tiene acceso a la Agenda Médica. Se corresponde con todos los miembros del personal médico existente en una institución hospitalaria que tienen como responsabilidad la planificación de actividades. Pueden gestionar tareas en su beneficio propio y planificar tareas para un grupo de funcionarios. Puede por otra parte consultar la planificación de los funcionarios asociados al área de la cual es encargado de la planificación.

Planificación: Contiene la administración del fondo de tiempo para los miembros del personal médico. Especifica las actividades asignadas a estos en un lapso de tiempo finito.

Capítulo 2: Características del Sistema

Guardia médica: Actividad a cumplir por el personal de la salud dentro de los centros hospitalarios. Especifica hora y fecha en que se realiza así como el período de tiempo en que tendrá validez además del funcionario que la desempeña.

Plan quirúrgico: Contiene información referente a la planificación de intervenciones quirúrgicas a realizar en el área de Quirófanos. Presenta información de estas, tales como: Fecha para la cual se realiza, numeración de las intervenciones a realizar, hora de las intervenciones, quirófano en el cual se llevara a cabo cada intervención quirúrgica, nombre del paciente, equipos de médicos que intervendrán, tipos de cirugía así como el procedimiento que se desarrollará.

Planificación de consulta externa: Contiene la administración del fondo de tiempo para los miembros del personal médico que debe proporcionar consultas a pacientes. Lleva implícita la especialidad, el servicio y el nombre del médico que interviene en la consulta. Incluye también el tipo de consulta a realizar así como hora de inicio y fin de la misma, meses del año y días de la semana en que se realizará y un listado de los pacientes a tratar.

Planificación de enfermeras por paciente: Contiene los datos de una asociación enfermera – paciente. Por lo que recoge datos tales como rango de fechas en que se realiza esta asociación, así como el rango de horarios que cubre, nombres y apellidos de los implicados, servicio en que se lleva a cabo y el encargado de esta asignación.

Diagrama de Modelo de Dominio

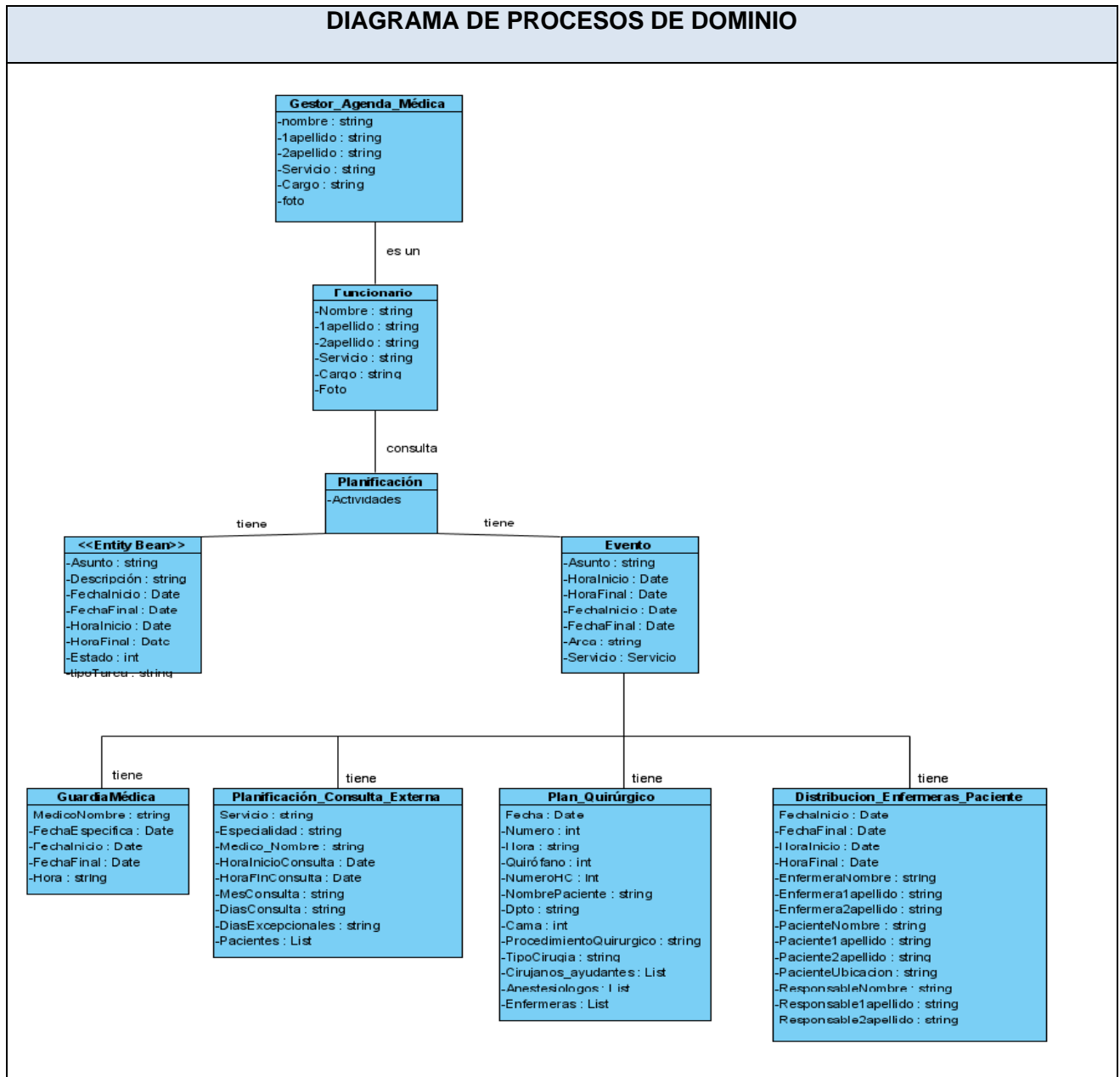


Figura 2.1 Diagrama de modelo de dominio.

Especificación de requerimientos de software

La especificación de requerimientos de software constituye la piedra angular dentro del proceso de elaboración de un software de alta calidad. El Glosario Estándar de Terminologías de Ingeniería de Software de la IEEE define los requerimientos de software como condiciones o capacidades que deben estar presentes en un sistema, para satisfacer un contrato, estándar, especificación u otro documento formal. Dada su importancia en el proceso de producción de software se debe realizar dicha tarea con la mayor precisión posible ya que está enfocada principalmente a un área: la definición del producto que se desea obtener, mediante una descripción más clara del comportamiento del sistema, por lo que se minimizan los problemas derivados de su desarrollo.

Requerimientos funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales.[26]

Requerimientos funcionales

Buscar funcionario: Funcionalidad que permite a los usuarios incluidos dentro del rol gestor de Agenda Médica realizar una búsqueda de un funcionario de su servicio con el fin de visualizar la planificación de sus actividades.

Crear tarea: Funcionalidad que permite a los usuarios adicionar una nueva tarea a las ya existentes. En el caso de los usuarios incluidos dentro del rol gestor de agenda médica, tienen la particularidad de poder crear tareas tanto para todos los funcionarios de su servicio o bien para grupo de ellos.

Capítulo 2: Características del Sistema

Ver detalles de tarea: Funcionalidad que permite a los funcionarios luego de seleccionar una tarea dentro una lista, visualizarla de forma individual a las demás de manera que podrá ver todos los datos que la componen.

Eliminar tarea: Funcionalidad que presentan todos los funcionarios. Mediante esta podrán eliminar una tarea dentro de la lista de sus tareas, ya sea porque le dio cumplimiento o cualquier razón que tenga. En el caso que la tarea sea creada por el jefe de servicio él será el único que podrá realizar esta acción.

Modificar tarea: Funcionalidad que presentan todos los funcionarios. Permite a estos gestionar cambios en una tarea previamente creada. En el caso que la tarea sea creada por el jefe de servicio él será el único que podrá realizar esta acción.

Consultar planificación: Funcionalidad que presentan todos los funcionarios. Esta les permite consultar en el momento que así lo decidan, la planificación de sus actividades de forma general o parcial, ya que la planificación se podrá visualizar en tres vistas diaria, semanal o mensual según las necesidades del usuario.

Buscar tarea: Funcionalidad que presentan todos los funcionarios. Les permite efectuar la búsqueda de una tarea específica a partir de un criterio de búsqueda establecido.

Buscar evento: Funcionalidad que presentan todos los funcionarios. Les permite efectuar la búsqueda de un evento específica a partir de un criterio de búsqueda establecido.

Requerimientos no funcionales

Los requerimientos no funcionales son cualidades o propiedades que el producto debe tener, es decir, restricciones en el producto que está en desarrollado. Estos no describen lo que el software hará, sino cómo lo hará. Son muy importantes debido a que permiten a los clientes valorar las características no funcionales del producto como: usabilidad, rendimiento y

Capítulo 2: Características del Sistema

portabilidad, que junto a las funcionalidades esperadas del software ayudarán a marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. Es preciso tener en cuenta que la definición de los requerimientos no funcionales cobra un valor adicional porque los errores que puedan existir en ellos son difíciles y caros de resolver.[27]

Usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido: Usuarios normales: 20 días, Usuarios avanzados: 30 días.

Seguridad

Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantiza sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión. Se registrarán todas las acciones que se realizan, y lleva el control de las actividades de cada usuario en todo momento. Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude. El sistema proporcionará un registro de actividades (log) de cada usuario en el sistema. Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD. El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

Rendimiento

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

Capítulo 2: Características del Sistema

Soporte

Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos por direcciones IP así como la administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación. Se podrán 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. Va a ser posible el chequeo de las operaciones y acceso de los usuarios al sistema. Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

Hardware

Estaciones de trabajo

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

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:

Capítulo 2: Características del Sistema

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

Software

El sistema debe correr en sistemas operativos Windows, Unix y Linux, utiliza la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL). El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

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

Requerimientos para la documentación de usuarios en línea y ayuda del sistema

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

Interfaz

Interfaces de usuario

Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de función y menús desplegados 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 el idioma previamente configurado.

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

El módulo de Agenda Médica es un servicio de apoyo a las instituciones hospitalarias. Para su funcionamiento se nutre de la información referente a los procesos de planificación de actividades que son generados internamente en el hospital por los servicios Consulta externa, Hospitalización, Bloque quirúrgico y Enfermería.

El módulo es diseñado para satisfacer la necesidad de centralizar la información de la planificación de actividades existentes. Contará con un proceso esencial: Realizar planificación. Actualmente se encuentran implementadas las funcionalidades que recogen la planificación de actividades en las diferentes áreas de las instituciones hospitalarias, siendo estas la base fundamental de la información a manipular. Esta es reutilizada a la hora de conformar los horarios o agendas de trabajo de los médicos del centro durante el proceso Realizar planificación.

Se reutilizan además las funcionalidades que brinda el EJB Bitácora del módulo Configuración para llevar a cabo el registro de las acciones que realiza el usuario como son: actualizar, modificar, ver, liberar, aceptar y crear.

2.6 Propuesta del sistema

El Proceso Unificado de Desarrollo (RUP por sus siglas en inglés) propone, como una de sus disciplinas, que una vez concluida la elaboración del modelado de negocio, continúe el ciclo de desarrollo y proporciona el paso al flujo de trabajo Gestión de Requerimientos, que será el encargado de enumerar los requisitos que debe presentar el sistema, a partir de los cuales se construye el Diagrama de Casos de Uso del Sistema y posteriormente se identifican aquellos que son arquitectónicamente significativos.

2.7 Modelo de casos de uso del sistema

El Modelo de Casos de Uso del Sistema es un artefacto de Ingeniería de Software que describe un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. Este modelo está formado por actores, casos de usos y las relaciones que se establecen entre estos. Por tanto representa de manera gráfica, los procesos que tendrán lugar en un sistema, así como la interacción con los actores por lo que constituye una entrada invaluable para las siguientes fases de construcción de un software.

Definición de actores

Un actor del sistema es una persona o la abstracción de un software que interactúa de alguna manera con el sistema: puede intercambiar información con él, ser un recipiente pasivo de información, representar el rol que juegan una o varias personas, un equipo o un sistema automatizado. A continuación se definen los actores del sistema en desarrollo. [28]

Capítulo 2: Características del Sistema

ACTORES	JUSTIFICACION
Gestor de la agenda médica	Actor encargado de gestionar la planificación de las actividades de cada funcionario del sistema.
Funcionario	Actor encargado de la gestión de tareas personales, así como la consulta de eventos y tareas que le han sido previamente planificadas.

Tabla 2.1 Definición de actores.

Vista global de actores



Figura 2.2 Vista Global Actores del Sistema

Diagrama de casos de uso

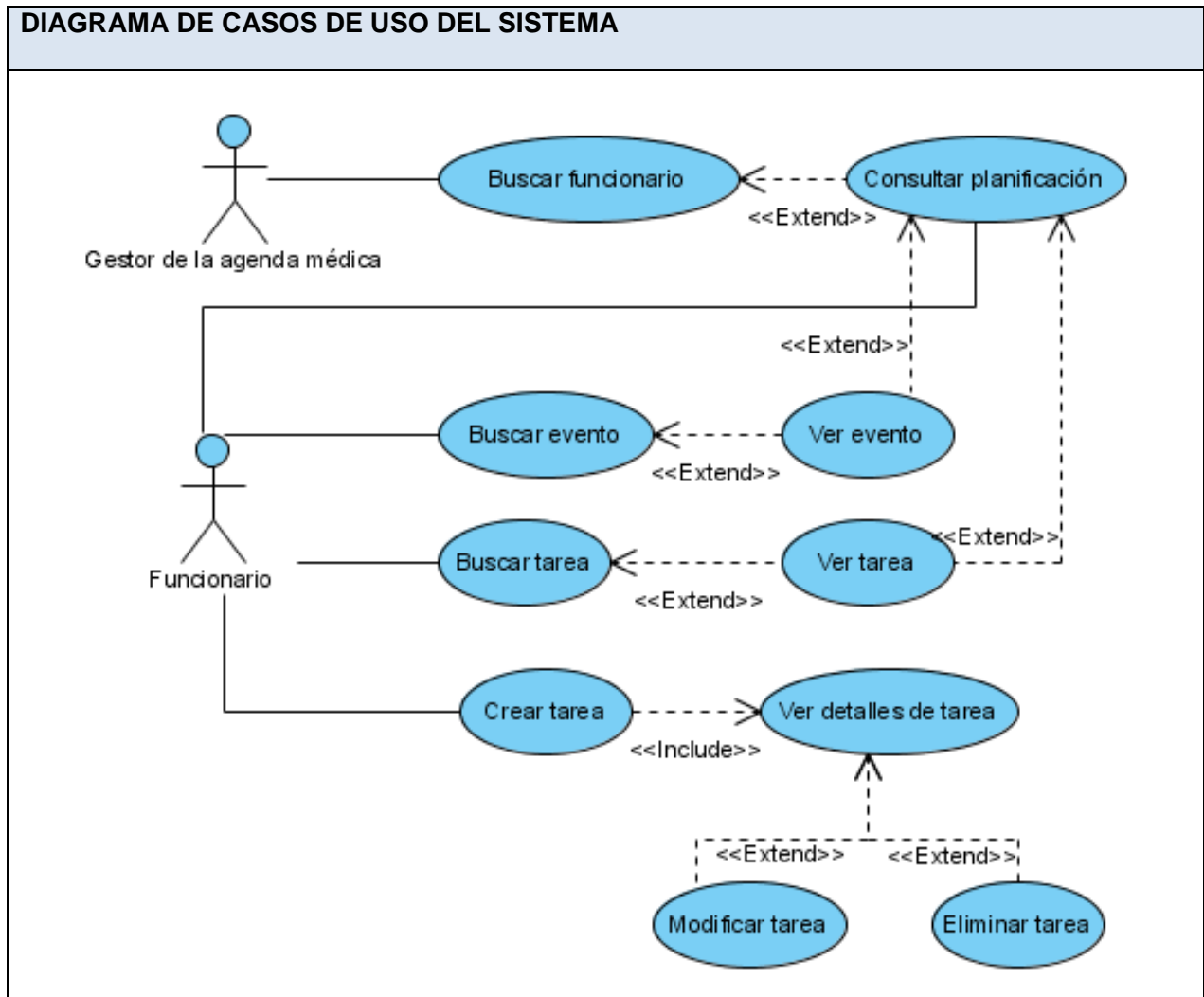


Figura 2.3 Diagrama de Casos de Uso Proceso Realizar planificación.

Capítulo 2: Características del Sistema

Descripción Textual de los Casos de Uso

CUS_1	Buscar funcionario
Actores:	Gestor de la agenda médica
Descripción:	El caso de uso inicia cuando el actor accede a la opción Buscar funcionario. El sistema muestra los funcionarios que tienen actividades planificadas que pertenecen al servicio del cual el actor es el encargado de realizar la planificación de actividades, y termina así el caso de uso.
Referencias:	RF1

Tabla 2.2 Descripción textual CUS 1 Buscar funcionario.

CUS_2	Buscar evento
Actores:	Gestor de la agenda médica, funcionario
Descripción:	El caso de uso inicia cuando el actor accede a la opción Buscar evento. El sistema muestra los campos que se deben llenar, para ello establece un criterio de búsqueda. Busca el evento que coincida con este criterio y lo muestra, en caso que la búsqueda no tenga resultados notifica al usuario, termina así el caso de uso.
Referencias:	RF2

Tabla 2.3 Descripción textual CUS 2 Buscar evento.

CU_3	Consultar planificación
Actores:	Gestor de la agenda médica, funcionario
Descripción:	El caso de uso inicia cuando el actor accede a la opción Consultar planificación. El sistema muestra la planificación de las actividades correspondientes al actor, permite además ver cada actividad de forma independiente, termina así el caso de uso.
Referencias:	RF3

Tabla 2.4 Descripción textual CUS 3 Consultar planificación.

Capítulo 2: Características del Sistema

CUS_4	Ver evento
Actores:	Funcionario
Descripción:	El caso de uso inicia cuando el actor selecciona un evento y accede a la opción de Ver datos de evento, el sistema muestra los datos del evento, el caso de uso termina.
Referencias:	RF4

Tabla 2.5 Descripción textual CUS 4 Ver evento.

CU_5	Buscar tarea
Actores:	Gestor de la agenda médica, funcionario
Descripción:	El caso de uso inicia cuando el actor accede a la opción Buscar tarea. El sistema muestra los campos que se deben llenar, para ello establece un criterio de búsqueda. Busca la tarea que coincida con este criterio y lo muestra, en caso que la búsqueda no tenga resultados notifica al usuario, termina así el caso de uso.
Referencias:	RF5

Tabla 2.6 Descripción textual CUS 5 Buscar tarea.

CU_6	Ver tarea
Actores:	Funcionario
Descripción:	El caso de uso inicia cuando el actor selecciona una tarea y accede a la opción de Ver datos de tarea, el sistema muestra los datos de la tarea, el caso de uso termina.
Referencias:	RF6

Tabla 2.7 Descripción textual CUS 6 Ver tarea.

Capítulo 2: Características del Sistema

CUS_7	Crear tarea
Actores:	Funcionario
Descripción:	El caso de uso inicia cuando el actor accede a la opción Crear tarea, el sistema brinda la posibilidad de introducir los datos para crear la tarea, el actor introduce los datos de la tarea, el sistema crea la tarea, el caso de uso termina.
Referencias:	RF7

Tabla 2.8 Descripción textual CUS 7 Crear tarea.

CUS_8	Ver detalles de tarea
Actores:	Funcionario
Descripción:	El caso de uso inicia cuando el actor selecciona una tarea y accede a la opción de Ver detalles de tarea, el sistema muestra los datos de la tarea, el caso de uso termina.
Referencias:	RF8

Tabla 2.9 Descripción textual CUS 8 Ver detalles de tarea.

CUS_9	Modificar tarea
Actores:	Funcionario
Descripción:	El caso de uso inicia cuando el actor selecciona una tarea y accede a la opción Modificar tarea, el sistema muestra los datos de la tarea y brinda la posibilidad de cambiar sus valores ya sea al introducir nuevos o seleccionar diferentes, el actor modifica los datos que necesita, el sistema actualiza los datos de la tarea, el caso de uso termina.
Referencias:	RF9

Tabla 2.10 Descripción textual CUS 9 Modificar tarea.

Capítulo 2: Características del Sistema

CUS_10	Eliminar tarea
Actores:	Funcionario
Descripción:	El caso de uso inicia cuando el actor selecciona una tarea y accede a la opción Eliminar tarea, el sistema elimina la tarea, el caso de uso termina.
Referencias:	RF10

Tabla 2.11 Descripción textual CUS 10 Eliminar tarea

A lo largo del desarrollo de este capítulo han quedado expuestos y especificados los requerimientos tanto funcionales como no funcionales del sistema, así como los actores que estarán presentes e intervendrán en el mismo. Por otra parte se detallaron todos los casos de uso que describen las funcionalidades previamente especificadas. El desarrollo de este flujo de trabajo y los artefactos que se obtuvieron a partir del mismo permiten dar comienzo al Flujo de Trabajo de Diseño, que requiere de mayor esfuerzo por parte del equipo de desarrollo para la construcción de la solución de software propuesta.

CAPÍTULO 3: DISEÑO DEL SISTEMA

Teniendo como objetivo principal la traducción de los requisitos, el diseño debe ser capaz de describir la futura implementación del sistema sin ambigüedades. Motivo por el cual el diseño se puede considerar la piedra angular en cuanto a calidad del sistema se refiere, ya que materializa en funcionalidades los requisitos de los usuarios. En el presente capítulo se pretende asimilar los aspectos relacionados con los requerimientos, tecnologías de distribución y concurrencia así como con los lenguajes de programación. Por otra parte se abarcarán y mostrarán los artefactos Modelo de Diseño, Diagrama de Clases y Descripción de las Clases de Diseño.

3.1 Modelo de Diseño

Es un modelo de objetos en el cual se describe la materialización de los casos de uso, enfocándose en el nivel de impacto que tienen los requisitos en el sistema. Describe las clases de mayor relevancia, así como su organización en paquetes y subsistemas. Sirve como abstracción de la implementación y se utiliza como la fundamental entrada de las actividades de implementación. Es representado por un sistema de diseño que representa el subsistema de más alto nivel del modelo. La utilización de otro subsistema es una forma de organización de este artefacto, en porciones más manejables. Los casos de uso son realizados por las clases del diseño y sus objetos mediante colaboraciones en el Modelo de Diseño, de manera que se denota la realización de casos de uso del diseño.

Estructuración

En la conformación del Modelo de Diseño, para lograr la optimización de esta actividad se hace necesario su estructuración u organización en entidades que posibiliten un mayor grado de manipulación por lo que se fragmenta en subsistemas o paquetes. Estos contienen sus interfaces y las dependencias entre cada una de ellas, se conforma de esta manera la estructura principal del sistema. A continuación se muestra el diagrama con los paquetes del diseño, estructurados según el paradigma de la Arquitectura en tres capas.

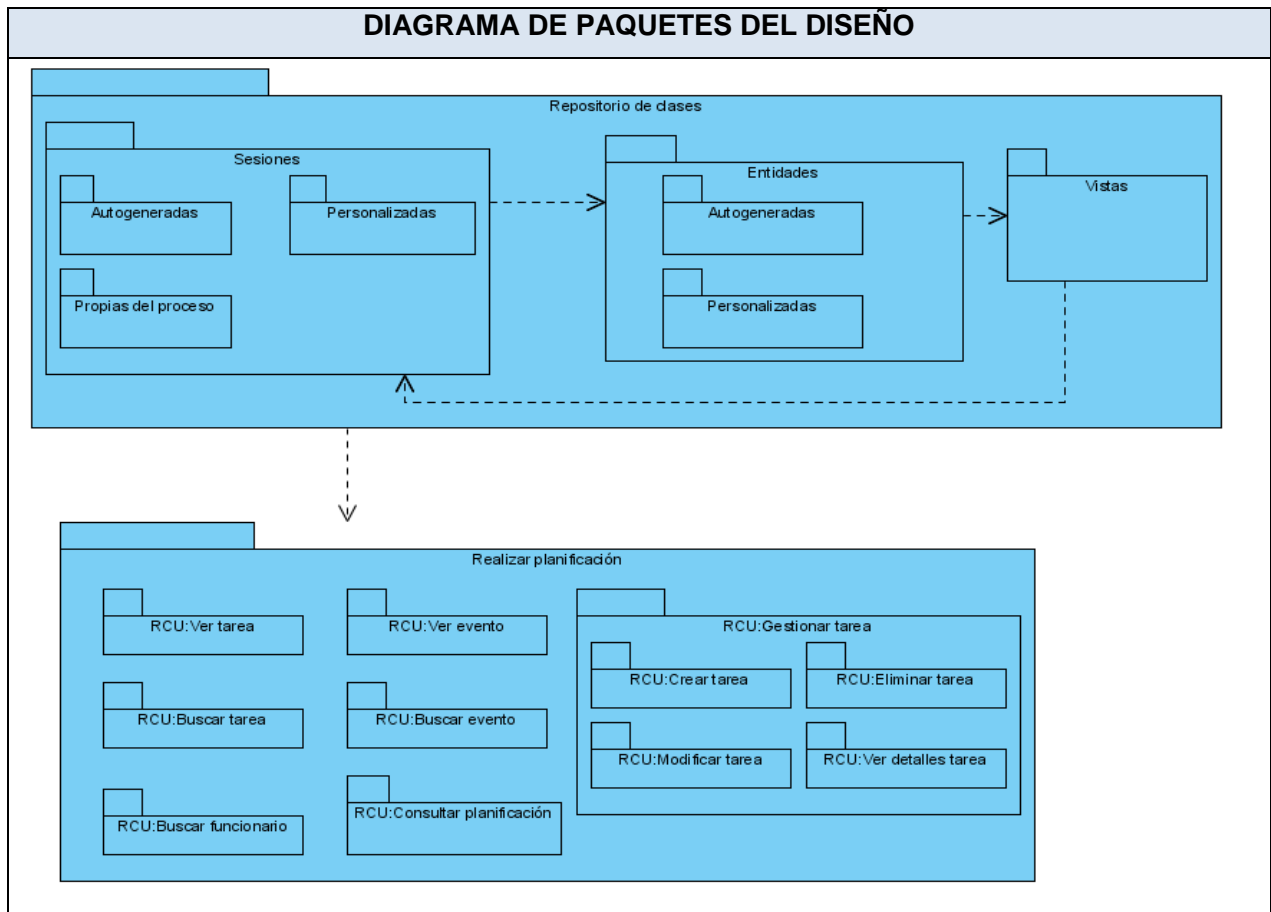


Figura 3.1 Diagrama de paquetes del diseño.

Para hacer posible un mejor entendimiento del Diagrama de paquetes del Diseño, se describe a continuación el contenido de cada paquete contenido en el mismo.

El paquete Repositorio de clases contiene todas las clases definidas en el diseño de acuerdo a las tecnologías, que serán utilizadas en la implementación del módulo, estructuradas en tres paquetes:

Entidades: Contiene las entidades Autogeneradas y Personalizadas de la base de datos. Las Autogeneradas son aquellas obtenidas mediante el proceso de ingeniería inversa de la base de

datos, utilizan para ello las herramientas ORM de Hibernate. Las Personalizadas son aquellas que son modificadas y, en algunos casos, pueden presentar relaciones de herencia o composición con las Autogeneradas.

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

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

Definición de elementos de diseño

Para realizar el Diagrama de Clase del Diseño se utiliza la extensión de UML que permite la utilización de los estereotipos web necesarios. Esta extensión presenta entre sus elementos tres clases UML que serán utilizadas para el código servidor, el código cliente y los formularios existentes estas son las "Sever Pages", "Client Pages" y los "Form" respectivamente.

El código servidor tiene a su cargo la construcción del resultado XHTML que conforma el contenido del código cliente, por su parte los formularios envían sus datos al código servidor para que sean procesados los pedidos (<<submit>>), además forman parte del código cliente o resultado XHTML. Es por ello que la relación que se establece entre la clase empleada para el código cliente y la clase empleada para el formulario es de agregación. Entre las páginas clientes pueden existir relaciones de vínculos (<<link>>) o redireccionamientos (<<redirect>>).

Es importante resaltar que una página cliente es construida por una sola página servidora; esta a su vez, puede incluir código existente en otra página de este mismo tipo, utilizan para ello una relación de inclusión (<<include>>), que aunque no es propia de la extensión de UML, las herramientas de modelado la consideran para representar todas las relaciones existentes en el modelo.

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

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

En la estructura de los Diagramas de Clases del Diseño también se pone de manifiesto la aplicación de los patrones de diseño, así como las restricciones que establecen sobre la arquitectura definida. Estos diagramas están estructurados según el patrón MVC, el cual permite la separación en capas de los objetos que componen el diseño del sistema: vistas, controladores y modelos.

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

La capa de datos en el patrón MVC también es parte del modelo. En esta capa está presente la utilización los patrones de diseño Abstract Factory, Singleton y los patrones de arquitectura de aplicaciones empresariales; más específicamente los patrones de persistencia o mapeo objeto relacional. Entre estos se encuentran los patrones Active Record, Identity map, Identity field, Foreign key mapping, Association table mapping, Lazy load y Query object.

Otros de los patrones de diseño que se utilizan son los patrones GRASP (General Responsibility Assignment Software Patterns), los cuales tienen como objetivo describir los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Dentro de estos, los patrones Experto, Creador, Alta cohesión y Bajo acoplamiento. Mediante la asignación a cada clase de las tareas o responsabilidades que estas pueden realizar en dependencia de la información que contienen se evidencia el uso de los patrones Experto y Creador. Definen quién será el responsable de crear una instancia de una clase. Al utilizar los patrones Alta cohesión y Bajo acoplamiento se permite la colaboración entre clases o elementos del diseño sin que se afecte su reutilización y entendimiento cuando se encuentren aislados.

3.2 Diagramas de clases del diseño

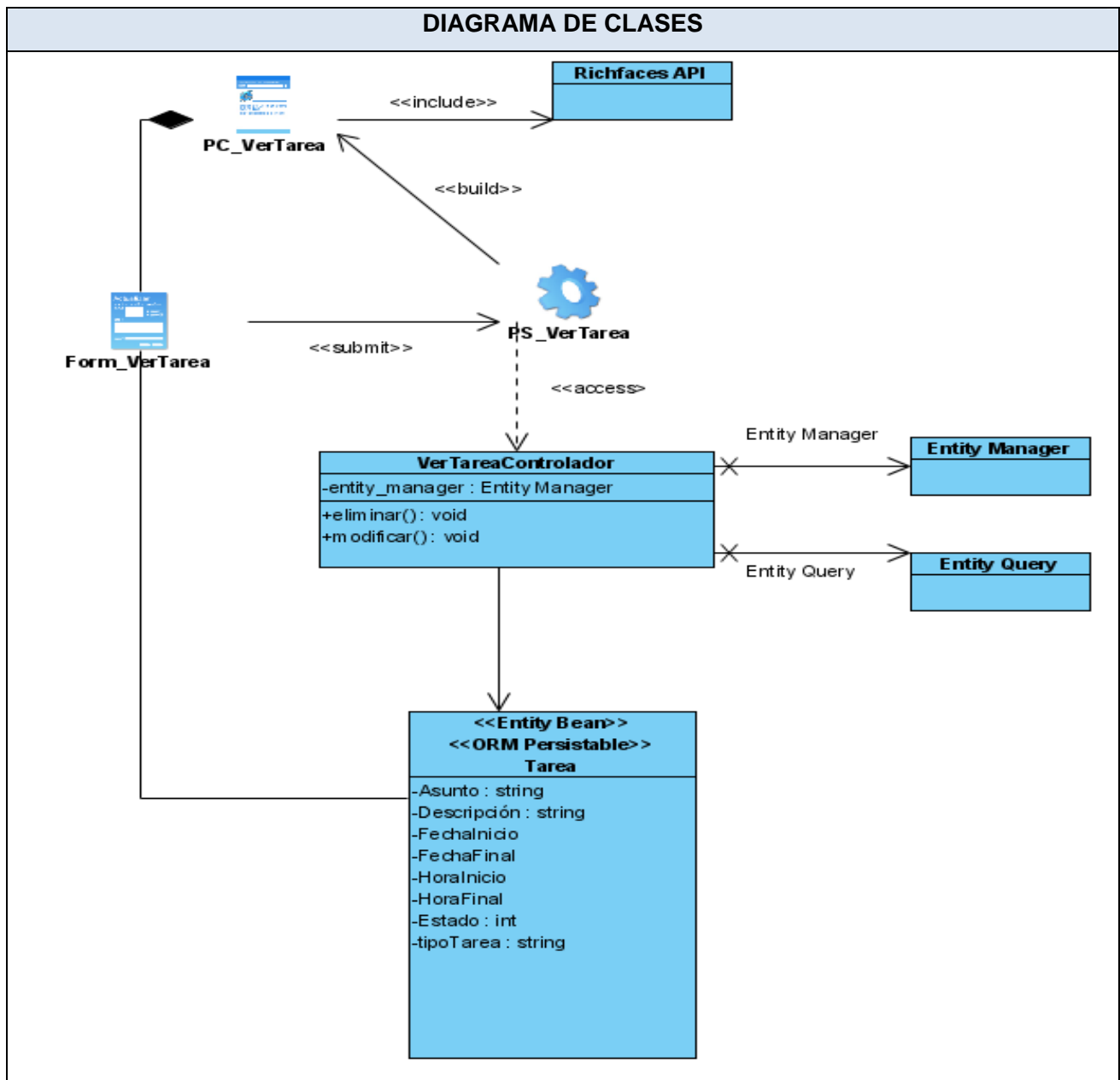


Figura 3.2 Diagrama de Clases del Diseño DCD_Ver Tarea.

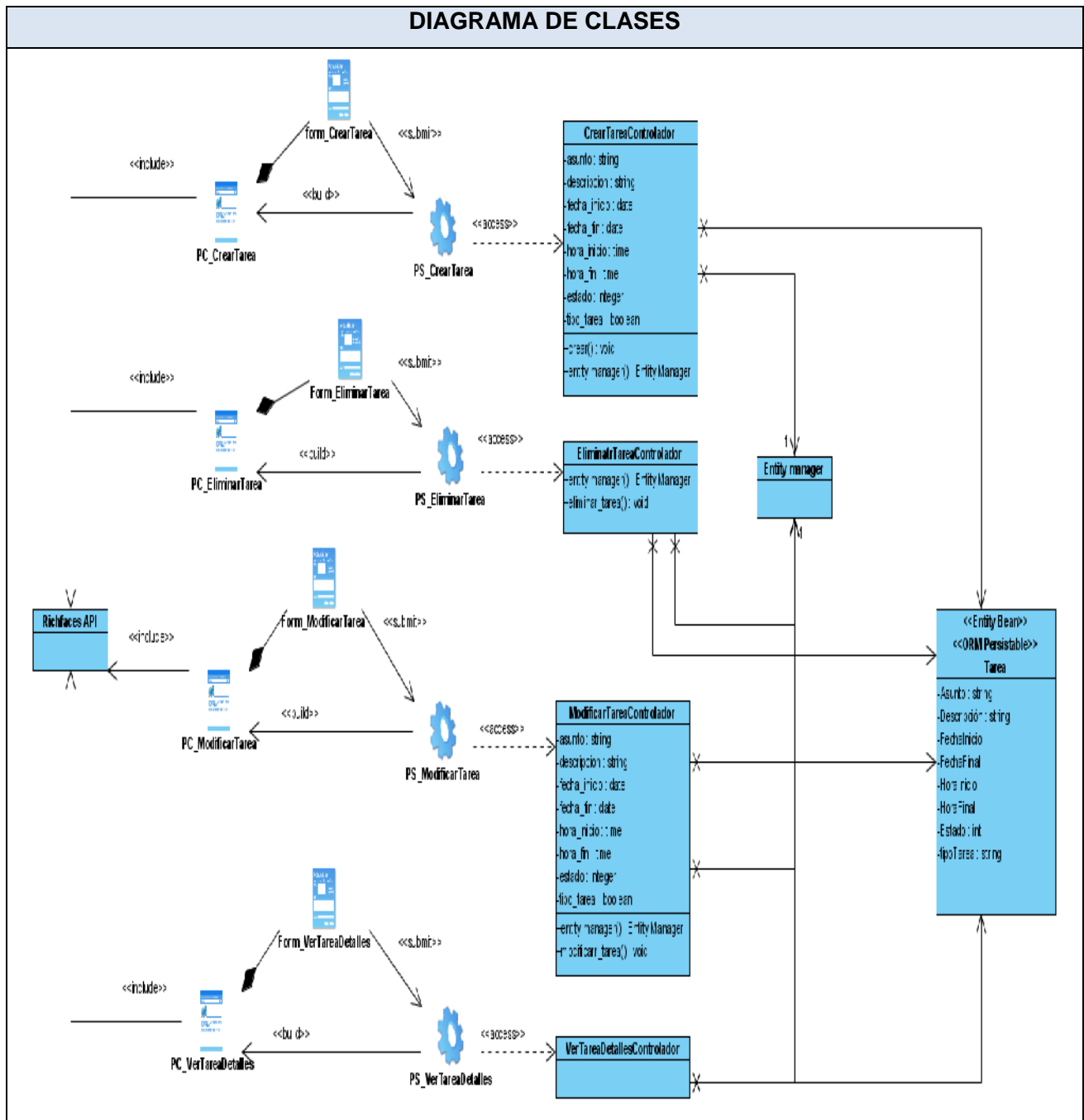


Figura 3.3 Diagrama de Clases del Diseño DCD_Gestionar Tarea.

3.3 Diagramas de secuencia

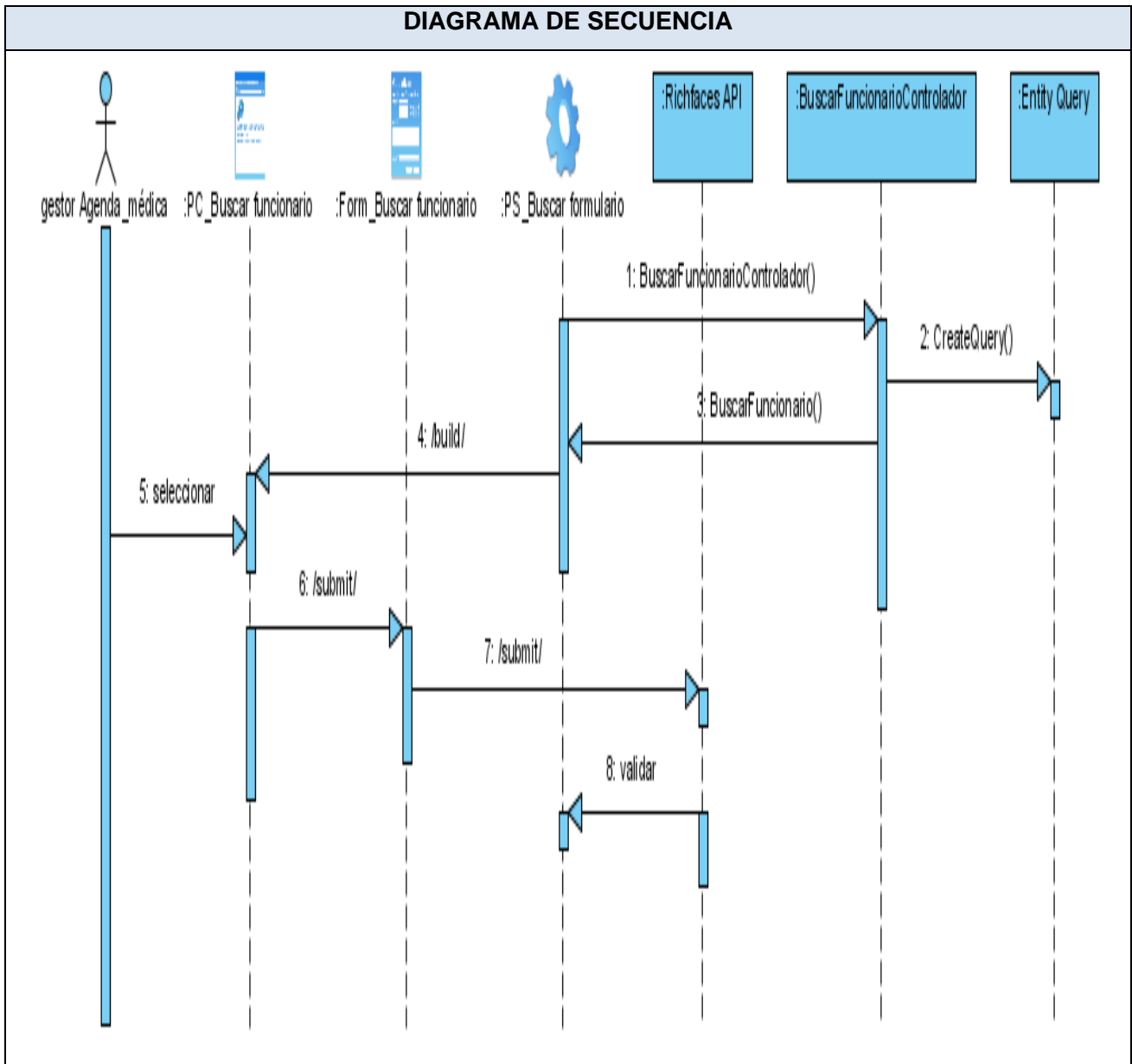


Figura 3.4 Diagrama de Secuencia DS_Buscar funcionario.

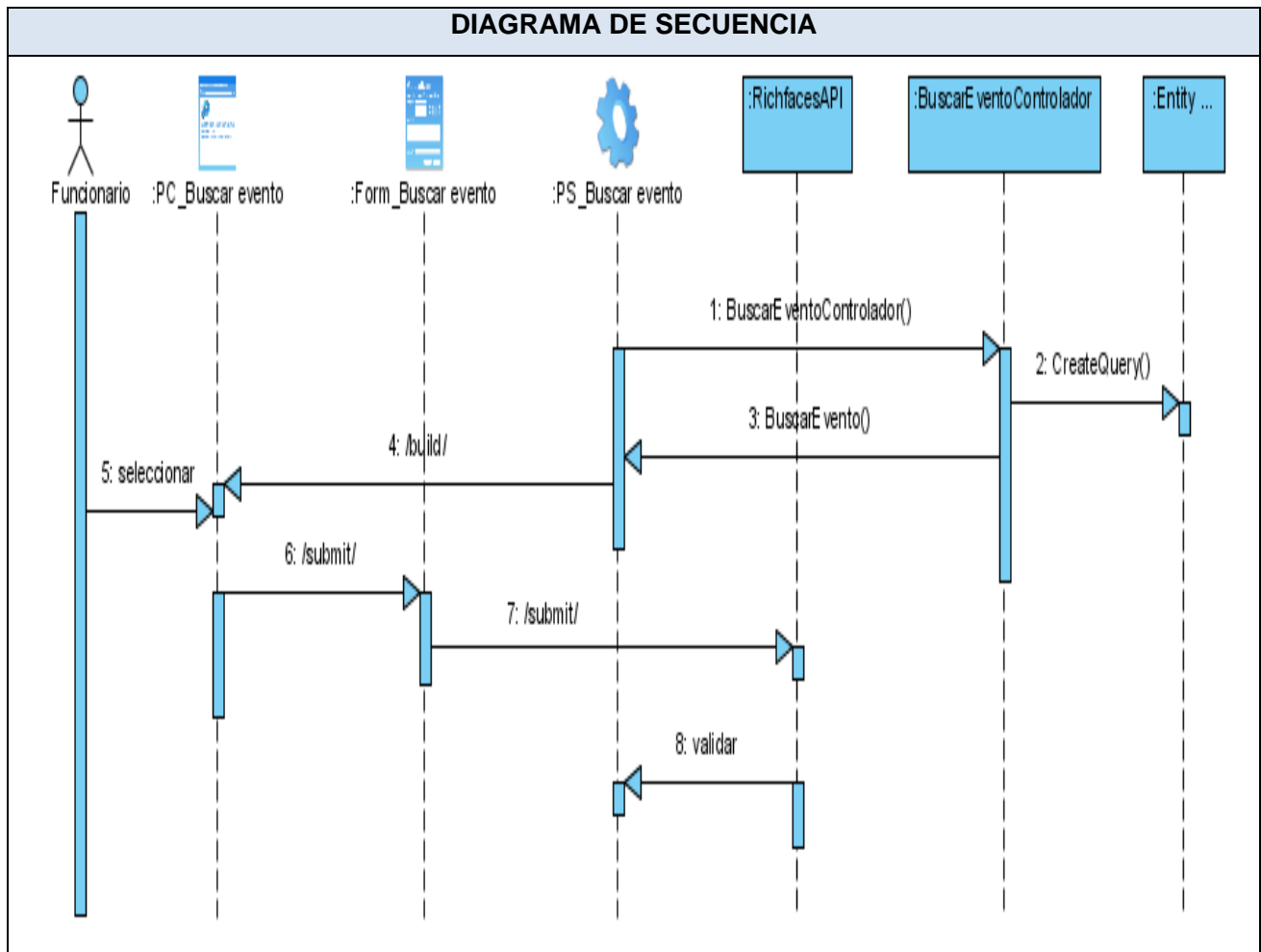


Figura 3.5 Diagrama de Secuencia DS_Buscar Evento.

Descripción de clases del diseño

A continuación se explican algunas de las clases identificadas para su futura implementación. De igual modo son descritas algunas de las responsabilidades que realizarán las páginas servidoras que responden a la lógica del negocio. De esta manera se alcanzará una comprensión mayor del funcionamiento que tendrá el sistema en desarrollo.

Descripción de páginas clientes

Nombre: PC_BuscarFuncionario
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, le permite al Gestor de la agenda médica buscar un funcionario de su servicio.

Tabla 3.1 Descripción página cliente PC_BuscarFuncionario.

Nombre: PC_ConsultarPlanificación
Tipo de Clase: Página Cliente
Descripción General: Página web que se ejecuta del lado del cliente, a los funcionarios consultar la planificación de sus actividades.

Tabla 3.2 Descripción página cliente PC_ConsultarPlanificación.

Nombre: PC_BuscarEvento
Tipo de Clase: Página Cliente
Descripción General: Es una página que se ejecuta del lado del cliente. Le permite a los funcionarios introducir los criterios de búsqueda para buscar un evento determinado dentro de su planificación de actividades

Tabla 3.3 Descripción página cliente PC_BuscarEvento.

Nombre: PC_BuscarTarea
Tipo de Clase: Página Cliente
Descripción General: Es una página que se ejecuta del lado del cliente. Les permite a los funcionarios introducir los criterios de búsqueda para buscar una tarea determinada dentro de su planificación de actividades.

Tabla 3.4 Descripción página cliente PC_BuscarTarea.

Nombre: PC_VerTarea
Tipo de Clase: Página Cliente
Descripción General: Es una página que se ejecuta del lado del cliente. Les permite a los funcionarios visualizar la planificación de sus tareas.

Tabla 3.5 Descripción página cliente PC_VerTarea.

Nombre: PC_VerEvento
Tipo de Clase: Página Cliente
Descripción General: Es una página que se ejecuta del lado del cliente. Les permite a los funcionarios visualizar la planificación de sus eventos.

Tabla 3.6 Descripción página cliente PC_VerEvento.

Nombre: PC_Crear Tarea
Tipo de Clase: Página Cliente
Descripción General: Es una página que se ejecuta del lado del cliente. Les permite a los funcionarios crear una nueva tarea y añadirla a las ya existentes.

Tabla 3.7 Descripción página cliente PC_Crear Tarea.

Descripción de páginas servidoras

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

Tabla 3.8 Descripción página servidora PS_BuscarFuncionario.

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

Tabla 3.9 Descripción página servidora PS_BuscarTarea.

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

Tabla 3.10 Descripción página servidora PS_BuscarEvento.

3.4 Descripción de clases del diseño

Nombre: ConsultarPlanificaciónControlador					
Tipo de clase : Controladora					
Atributo	Tipo				
entity_manager	Entity_manager				
Para cada responsabilidad:					
<u>Nombre:</u>	<u>listar_tarea()</u>				
Descripción:	Listar las tareas existentes				
<u>Nombre:</u>	<u>listarEvento()</u>				
Descripción:	Listar los eventos existentes				
<table border="1" style="background-color: #e0f0ff; width: 100%;"> <tr> <td style="text-align: center;">ConsultarPlanificación Controlador</td> </tr> <tr> <td>-entity manager : Entity Manager</td> </tr> <tr> <td>+listarEvento() : List</td> </tr> <tr> <td>+listar_tarea() : List</td> </tr> </table>		ConsultarPlanificación Controlador	-entity manager : Entity Manager	+listarEvento() : List	+listar_tarea() : List
ConsultarPlanificación Controlador					
-entity manager : Entity Manager					
+listarEvento() : List					
+listar_tarea() : List					

Tabla 3.11 Descripción página Controladora ConsultarPlanificaciónControlador.

Nombre: VerTareaControlador				
Tipo de clase : Controladora				
Atributo	Tipo			
entity_manager	Entity_manager			
Para cada responsabilidad:				
<u>Nombre:</u>	<u>listarTarea()</u>			
Descripción:	Listar las tareas existentes			
<table border="1" style="background-color: #ADD8E6; width: 100%;"> <tr> <td style="text-align: center;">VerTareaControlador</td> </tr> <tr> <td>-entity_manager : Entity Manager</td> </tr> <tr> <td>+listarTarea() : List</td> </tr> </table>		VerTareaControlador	-entity_manager : Entity Manager	+listarTarea() : List
VerTareaControlador				
-entity_manager : Entity Manager				
+listarTarea() : List				

Tabla 3.12 Descripción página Controladora VerTareaControlador.

Nombre: VerEventoControlador				
Tipo de clase : Controladora				
Atributo	Tipo			
entity_manager	Entity_manager			
Para cada responsabilidad:				
<u>Nombre:</u>	<u>listarEvento()</u>			
Descripción:	Listar los eventos existentes			
<table border="1" style="background-color: #ADD8E6; width: 100%;"> <tr> <td style="text-align: center;">VerEventoControlador</td> </tr> <tr> <td>-entity manager : Entity manager</td> </tr> <tr> <td>+listarEvento() : List</td> </tr> </table>		VerEventoControlador	-entity manager : Entity manager	+listarEvento() : List
VerEventoControlador				
-entity manager : Entity manager				
+listarEvento() : List				

Tabla 3.13 Descripción página Controladora VerEventoControlador.

Nombre: BuscarFuncionarioControlador		
Tipo de clase : Controladora		
Atributo	Tipo	
entity_manager	Entity_manager	
nombre	string	
Apellido1	string	
Apellido2	string	
cargo	string	
servicio	string	
Para cada responsabilidad:		
Nombre:	<u>listarFunc()</u>	
Descripción:	Listar los funcionarios existentes	
Nombre:	<u>BuscarFuncionario()</u>	
Descripción:	Busca todos los funcionarios existentes	
<table border="1"> <tr> <td> <p>BuscarFuncionarioControlador</p> <p>-nombre : string -apellido1 : string -apellido2 : string -cargo : string -servicio : string -entity manager : Entity Manager</p> <p>+BuscarFuncionario() +listarFunc(): List</p> </td> </tr> </table>		<p>BuscarFuncionarioControlador</p> <p>-nombre : string -apellido1 : string -apellido2 : string -cargo : string -servicio : string -entity manager : Entity Manager</p> <p>+BuscarFuncionario() +listarFunc(): List</p>
<p>BuscarFuncionarioControlador</p> <p>-nombre : string -apellido1 : string -apellido2 : string -cargo : string -servicio : string -entity manager : Entity Manager</p> <p>+BuscarFuncionario() +listarFunc(): List</p>		

Tabla 3.14 Descripción página Controladora BuscarFuncionarioControlador.

Nombre: BuscarTareaControlador				
Tipo de clase : Controladora				
Atributo	Tipo			
entity_manager	Entity_manager			
asunto	string			
Fecha_inicio	date			
Fecha_fin	date			
Para cada responsabilidad:				
<u>Nombre:</u>	<u>listarTarea()</u>			
Descripción:	Listar todas las tareas existentes.			
<u>Nombre:</u>	<u>buscarTarea()</u>			
Descripción:	Busca todas las tareas existentes.			
<table border="1" style="background-color: #e0f0ff;"> <tr> <td style="text-align: center;">BuscarTareaControlador</td> </tr> <tr> <td>-asunto : string -fecha_inicio : date -fecha_final : date -entity manager : Entity Manager</td> </tr> <tr> <td>+buscarTarea() +listarTarea() : List</td> </tr> </table>		BuscarTareaControlador	-asunto : string -fecha_inicio : date -fecha_final : date -entity manager : Entity Manager	+buscarTarea() +listarTarea() : List
BuscarTareaControlador				
-asunto : string -fecha_inicio : date -fecha_final : date -entity manager : Entity Manager				
+buscarTarea() +listarTarea() : List				

Tabla 3.15 Descripción página controladora BuscarTareaControlador.

Nombre: BuscarEventoControlador		
Tipo de clase : Controladora		
Atributo	Tipo	
entity_manager	Entity_manager	
asunto	string	
Fecha_inicio	date	
Fecha_fin	date	
Para cada responsabilidad:		
<u>Nombre:</u>	<u>listarEvent()</u>	
Descripción:	Listar todos los eventos existentes.	
<u>Nombre:</u>	<u>buscar()</u>	
Descripción:	Busca todos los eventos existentes.	
<table border="1" style="background-color: #e0f0ff;"> <tr> <td> BuscarEventoControlador -fecha_inicio : date -fecha_fin : date -asunto : string -entity manager : Entity Manager +buscar() +listarEvent() : List </td> </tr> </table>		BuscarEventoControlador -fecha_inicio : date -fecha_fin : date -asunto : string -entity manager : Entity Manager +buscar() +listarEvent() : List
BuscarEventoControlador -fecha_inicio : date -fecha_fin : date -asunto : string -entity manager : Entity Manager +buscar() +listarEvent() : List		

Tabla 3.15 Descripción página controladora BuscarEventoControlador.

Nombre: CrearTareaControlador													
Tipo de clase : Controladora													
Atributo	Tipo												
entity_manager	Entity_manager												
asunto	string												
Fecha_inicio	date												
Fecha_fin	date												
Descripción	string												
Hora_inicio	time												
Hora_fin	time												
Estado	integer												
Tipo_tarea	bool												
Para cada responsabilidad:													
<u>Nombre:</u>	<u>crear()</u>												
Descripción:	Crea una nueva tarea (adiciona una tarea a la lista de tareas existentes).												
<u>Nombre:</u>	<u>validar()</u>												
Descripción:	Valida que los datos de la nueva tarea no existan y sean correctos.												
<table border="1"> <thead> <tr> <th>CrearTareaControlador</th> </tr> </thead> <tbody> <tr> <td>-asunto : string</td> </tr> <tr> <td>-descripcion : string</td> </tr> <tr> <td>-fecha_inicio : date</td> </tr> <tr> <td>-fecha_fin : date</td> </tr> <tr> <td>-hora_inicio : time</td> </tr> <tr> <td>-hora_fin : time</td> </tr> <tr> <td>-estado : integer</td> </tr> <tr> <td>-tipo_tarea : boolean</td> </tr> <tr> <td>-entity manager : Entity Manager</td> </tr> <tr> <td>+crear() : void</td> </tr> <tr> <td>+validar() : void</td> </tr> </tbody> </table>		CrearTareaControlador	-asunto : string	-descripcion : string	-fecha_inicio : date	-fecha_fin : date	-hora_inicio : time	-hora_fin : time	-estado : integer	-tipo_tarea : boolean	-entity manager : Entity Manager	+crear() : void	+validar() : void
CrearTareaControlador													
-asunto : string													
-descripcion : string													
-fecha_inicio : date													
-fecha_fin : date													
-hora_inicio : time													
-hora_fin : time													
-estado : integer													
-tipo_tarea : boolean													
-entity manager : Entity Manager													
+crear() : void													
+validar() : void													

Tabla 3.20 Descripción página controladora CrearTareaControlador.

3.5 Modelo de datos

Modelo que describe de manera abstracta la representación de los datos de una aplicación o sistema de información. Consiste en una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. El Modelo de Datos tiene gran importancia en el ciclo de desarrollo de software, y de manera particular para la fase de implementación, pues define formalmente las estructuras permitidas y las restricciones que se aplican con el fin de representar los datos del dominio de la aplicación. Está compuesto por objetos: entidades que existen y se manipulan; y atributos: características básicas de dichos objetos y relaciones: forma en que se enlazan los objetos entre sí.

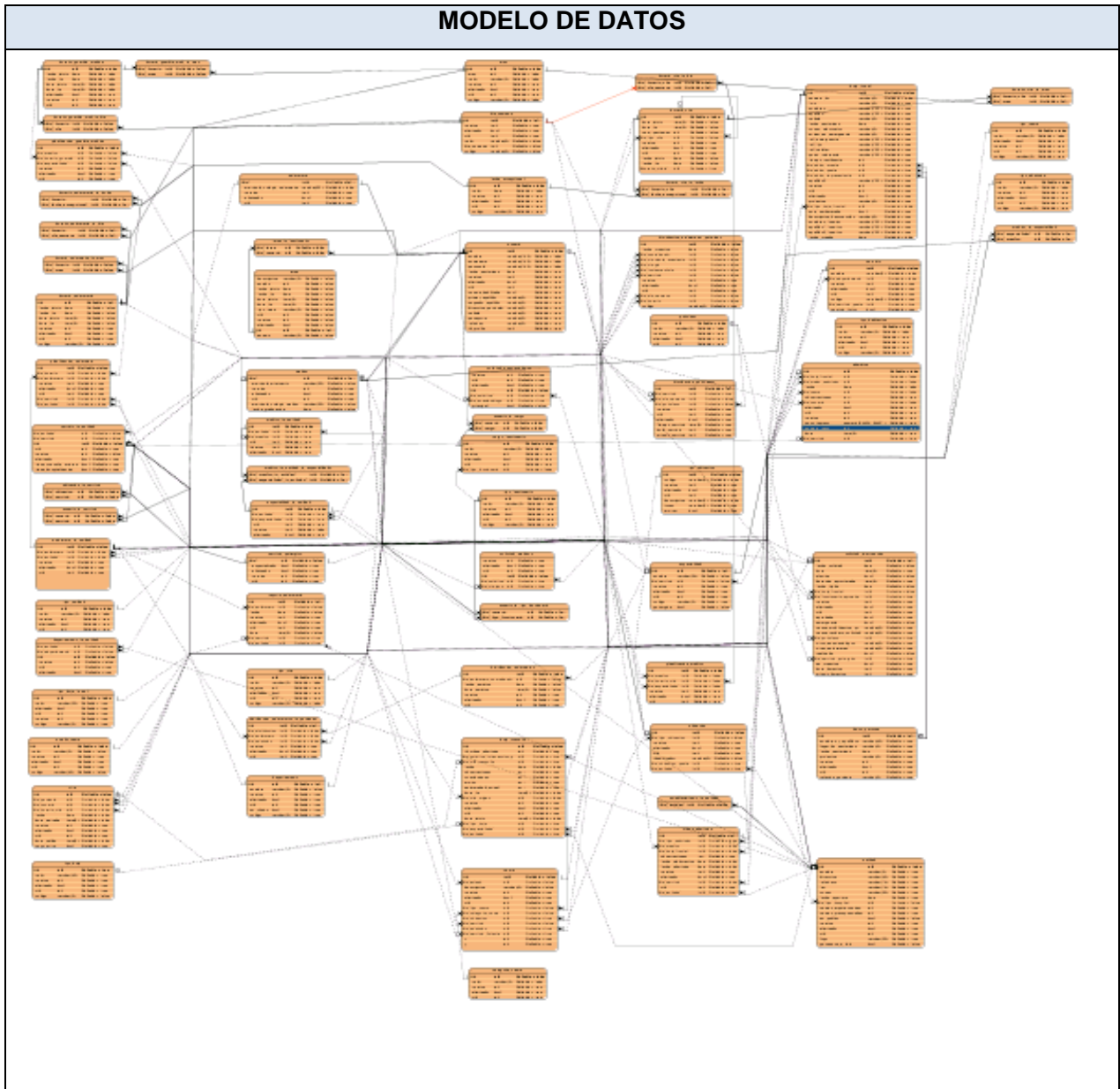


Figura 3.6 Diagrama Modelo de Datos.

A lo largo del desarrollo de este capítulo se obtuvo el Modelo de Diseño y se describió la realización física de los casos de uso. Centrándose en cómo los requerimientos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a desarrollar. Lo que constituyó una abstracción para la implementación y es utilizado como una entrada fundamental en las actividades que se realizan en el Flujo de Trabajo de Implementación.

CONCLUSIONES

Con la realización del presente trabajo y el cumplimiento de las tareas propuestas se pueden plantear las siguientes conclusiones:

- ✓ Durante el estudio de sistemas de información hospitalaria que presentan módulo Agenda Médica, se demostró que los sistemas analizados cuentan con las desventajas de estar basados en tecnologías y herramientas de carácter propietario, son muy costosos y no están integrados al Sistema de Información Hospitalaria alas HIS, por lo que no solucionan la problemática actual.
- ✓ Se evidenció la necesidad de gestionar el proceso de planificación en el módulo de Agenda Médica del Sistema de Información Hospitalaria alas HIS, con el objetivo de mejorar la calidad de la gestión de las actividades planificadas en el sistema.
- ✓ Mediante el Proceso Unificado de Desarrollo y el empleo de las tecnologías y herramientas, se obtuvo como resultado el prototipo no funcional, el cual constituye el punto de partida para la implementación del módulo Agenda Médica del sistema de información hospitalaria alas HIS.
- ✓ Se definió la estructura de diseño a utilizar para la futura implementación del módulo Agenda Médica que facilita un mejor entendimiento a los diseñadores de base de datos e implementadores.

RECOMENDACIONES

Una vez concluido el prototipo no funcional del módulo de Agenda Médica del Sistema de Información Hospitalaria alas HIS, se recomienda:

- ✓ La implementación de las funcionalidades definidas en el mismo.
- ✓ Definir en el módulo Agenda Médica una sección de Configuración, donde se pueda configurar para cada área los eventos que realiza. Haciendo posible que cada servicio pueda crear eventos.
- ✓ Agregar un reporte estadístico al módulo Agenda Médica que posibilite a los jefes de cada servicio llevar un control del estado en que se encuentran las tareas asignadas al personal del mismo.
- ✓ Implementar un sistema de alarmas configurable para los eventos y tareas en el módulo Agenda Médica.

REFERENCIAS BIBLIOGRÁFICAS

[1] Ramírez Márquez, Abelardo Dr.; Castell-Florit Serrate, Pastor Dr.; Mesa, Guillermo Dr. El Sistema Nacional de Salud de Cuba. Escuela Nacional de Salud Pública (ENSAP). La Habana, Cuba. 2003. Disponible en: http://www.sld.cu/galerias/doc/sitios/infodir//09_el_sistema_nacional_de_salud.doc

[2] IDEM 1

[3] IDEM 1

[4] Sistema de Información Hospitalaria. México D.F.: Universidad Autónoma de México. D. R. Facultad de Medicina, 2003. p 7. Disponible en: <http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf>

[5] Departamento de Lenguajes y Sistemas Informáticos. Tema 1: Patrones Arquitectónicos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Disponible en: <http://www.lsi.us.es/docencia/get.php?id=1891>

[6] Pattern-Oriented Software Architecture, Volume 1: A System of Patterns by Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal John Wiley & Son Ltd; ISBN 0471958697 (alternate, search); 1 edition (August 8, 1996)

[7] Reynoso, Carlos; Kicillof, Nicolás. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Universidad de Buenos Aires. 2004. Disponible en:

[8] Richfaces Developer Guide. Red Hat. 2008. Disponible en: <http://www.jboss.org/jbossrichfaces/docs/>

[9] Katz Max. Practical Richfaces. Apress. 2008.

- [10] Seam - Contextual Components. Introduction to JBoss Seam. Disponible en:
<http://www.seamframework.org/Documentation>
- [11] Allen Dan. Seam in action. Manning Early Access Program. Manning Publications Co. 2008. p 4, p 5, p6.
- [12] Documento de arquitectura de Software alas HIS. ALBET. 2008. p 69.
- [13] Java Persistence API FAQ. San Microsystems. Disponible en:
<http://java.sun.com/javaee/overview/faq/persistence.jsp>
- [14] Bauer Christian, King Gavin. Java Persistence with Hibernate. Manning Publications Co. 2007. p 67.
- [15] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.
- [16] Rumbaugh James, Jacobson Ivar, Booch Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. Pearson Education. 2007
- [17] Business Process Modeling Notation Specification. Object Management Group. Febrero 2006. p 1.
Disponible en: <http://www.omg.org/technology/agreement.htm>
- [18] IDEM 17
- [19] PostgreSQL Awards. Disponible en:
<http://www.postgresql.org/about/awards>
- [20] García de Jalón Javier, Rodríguez Iñigo Mingo José Ignacio, Alfonso Brazález Aitor Imaz, Larzabal Alberto, Calleja Jesús, García Jon. Aprende Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España. Enero 2000.

[21] Dr. Juntao Yuan Michael. On the road to simplicity.

Disponible en: <http://www.javaworld.com/javaworld/jw-02-2005/jw-0221-jboss4.html>

[22] JBoss Server Manager Reference Guide. Red Hat 2008.

Disponible en: <http://www.jboss.org/jbossas/docs/>

[23] Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.

[24] IDEM 23

[25] IDEM 23

[26] IDEM 23

[27] IDEM 23

[28] IDEM 23

BIBLIOGRAFÍA

- Allen Dan. Seam in action. Manning Early Access Program. Manning Publications Co. 2008.
- Bauer Christian, King Gavin. Java Persistence with Hibernate. Manning Publications Co. 2007.
- Bemmel JH Van. Medical Informatics. Art or Science? Meth Inform Me 1996.
- Business Process Modeling Notation Specification. Object Management Group. Febrero 2006. Disponible en: <http://www.omg.org/technology/agreement.htm>
- Canós José H., Letelier Patricio, Penadés M^a Carmen. Metodologías Ágiles en el Desarrollo de Software. DSIC. Universidad Politécnica de Valencia. Enero 2007. Disponible en <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- Conceptos generales de Sistemas Distribuidos. Departamento de Arquitectura de Computadoras. Universidad Politécnica de Catalunya. España. 2007. Disponible en: <http://studies.ac.upc.edu/EPSC/FSD/FSD-ConceptosGenerales.pdf>
- Departamento de Lenguajes y Sistemas Informáticos. Tema 1: Patrones Arquitectónicos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Disponible en: <http://www.lsi.us.es/docencia/get.php?id=1891>
- Documento de arquitectura de Software alas HIS. ALBET. La Habana, Cuba. 2008.

- Dr. Juntao Yuan Michael. On the road to simplicity. Disponible en:
- García de Jalón Javier, Rodríguez Iñigo Mingo José Ignacio, Alfonso Brazález Aitor Imaz, Larzabal Alberto, Calleja Jesús, García Jon. Aprende Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España. Enero 2000.
- Health Level Seven Spain. Disponible en: <http://www.hl7spain.org/VerPagina.asp?IDPage=0>
- Java Code Conventions. Sun Microsystems. California, Estados Unidos de América. 1997.
- Java Persistence API FAQ. Sun Microsystems. Disponible en: <http://java.sun.com/javaee/overview/faq/persistence.jsp>
- JBoss Server Manager Reference Guide. Red Hat 2008. Disponible en:
- Katz Max. Practical Richfaces. Apress. 2008.
- Marín Díaz, Miguel E. Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. La Habana. Cuba. 2006.
- Mandirola Brioux Humberto Fernán. La crisis en el Área Salud en la era de la información. Buenos Aires: BIOCUM. Disponible en: http://www.biocom.com/informatica_medica/La%20crisis%20en%20la%20%20era%20de%20la%20informaci%C3%B3n%20en%20el%20Area%20Salud.pdf
- Microsoft patterns and practices. Model-View-Controller. MSDN. 2009. Disponible en: <http://msdn.microsoft.com/en-us/library/ms978748.aspx>
- Navarro Franco, Ángel José. UML en acción. Modelando Aplicaciones Web. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba, Mayo 2005.

- Owen Martin, Raj Jog. BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. Popkin Software. 2003. Disponible en: http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf
- Pattern-Oriented Software Architecture, Volume 1: A System of Patterns by Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal John Wiley & Son Ltd; ISBN 0471958697 (alternate, search); 1 edition (August 8, 1996)
- Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Parte 1. La Habana, Cuba. Editorial Félix Varela. 2005
- PostgreSQL About. Disponible en: <http://www.postgresql.org/about>
- PostgreSQL Awards. Disponible en: <http://www.postgresql.org/about/awards>
- Propuesta para la informatización de Centros de Atención Médica. Softel. La Habana, Cuba. 2007. Disponible en: <http://www.softel.cu/doc/CentrosAtencionMedica.pdf>.
- Richfaces Developer Guide. Red Hat. 2008. Disponible en: <http://www.jboss.org/jbossrichfaces/docs/>
- Reynoso, Carlos Billy. Introducción a la Arquitectura de Software. Universidad de Buenos Aires. Versión 1.0. Junio 2006. Disponible en: <http://www.willydev.net/descargas/prev/IntroArq.pdf>
- Reynoso, Carlos; Kicillof, Nicolás. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Universidad de Buenos Aires. 2004. Disponible en: <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/estiloypatron.pdf>

- Rumbaugh James, Jacobson Ivar, Booch Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. Pearson Education. 2007.
- Seam - Contextual Components. Introduction to JBoss Seam. Disponible en: <http://www.seamframework.org/Documentation>
- Sistema de Información Hospitalaria. México D.F.: Universidad Autónoma de México. D. R. Facultad de Medicina, 2003. Disponible en: <http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf>
- Ramírez Márquez, Abelardo Dr.; Castell-Florit Serrate, Pastor Dr.; Mesa, Guillermo Dr. El Sistema Nacional de Salud de Cuba. Escuela Nacional de Salud Pública (ENSAP). La Habana, Cuba. 2003. Disponible en: http://www.sld.cu/galerias/doc/sitios/infodir//09_el_sistema_nacional_de_salud.doc.

GLOSARIO DE TÉRMINOS

Tarea

Clasificación que reciben aquellas actividades del sistema que son creadas por los propios usuarios. De manera que se denomina tarea a toda aquella actividad que los usuarios podrán crear, modificar, ver y eliminar.

Evento

Clasificación que reciben aquellas actividades del sistema que son creadas en las diferentes áreas de atención a los pacientes, dentro de los centros hospitalarios. De manera que se denomina evento a toda aquella actividad que los usuarios solo pueden consultar.

Funcionario

Tipo de usuario que tiene acceso a la Agenda Médica. Se corresponde con todos los miembros del personal médico existente en una institución hospitalaria que tienen actividades planificadas en las diferentes áreas de dicha institución. Pueden gestionar tareas en su beneficio propio, y consultar eventos que le son previamente planificados.

Gestor de la agenda médica

Tipo de usuario que tiene acceso a la Agenda Médica. Se corresponde con todos los miembros del personal médico existente en una institución hospitalaria que tienen como responsabilidad la planificación de actividades. Pueden gestionar tareas en su beneficio propio y planificar tareas para un grupo de funcionarios. Puede por otra parte consultar la planificación de los funcionarios asociados al área de la cual es encargado de la planificación.

Planificación

Contiene la administración del fondo de tiempo para los miembros del personal médico. Especifica las actividades asignadas a estos en un lapso de tiempo finito.

Guardia médica

Actividad a cumplir por el personal de la salud dentro de los centros hospitalarios. Especifica hora y fecha en que se realiza así como el periodo de tiempo en que tendrá validez además del funcionario que la desempeña.

Plan quirúrgico

Contiene información referente a la planificación referente a intervenciones quirúrgicas a realizar en el área de Quirófanos. Presenta información de estas, tales como: Fecha para la cual se realiza, numeración de las intervenciones a realizar, hora de las intervenciones, quirófano en el cual se llevara a cabo cada intervención quirúrgica, nombre de los pacientes, equipos de médicos que intervendrán, tipos de cirugía así como el procedimiento que se desarrollará.

Planificación de consulta externa

Contiene la administración del fondo de tiempo para los miembros del personal médico que debe proporcionar consultas a pacientes. Lleva implícita la especialidad, el servicio y el nombre del médico que interviene en la consulta. Incluye también el tipo de consulta a realizar así como hora de inicio y fin de la misma, meses del año y días de la semana en que se realizará y un listado de los pacientes a tratar.

Planificación de enfermeras por paciente

Contiene los datos de una asociación enfermera –paciente. Por lo que recoge datos tales como rango de fechas en que se realiza esta asociación, así como el rango de horarios que cubre, nombres y apellidos de los implicados, servicio en que se lleva a cabo y el encargado de esta asignación.