Universidad de las Ciencias Informáticas

Facultad 2

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

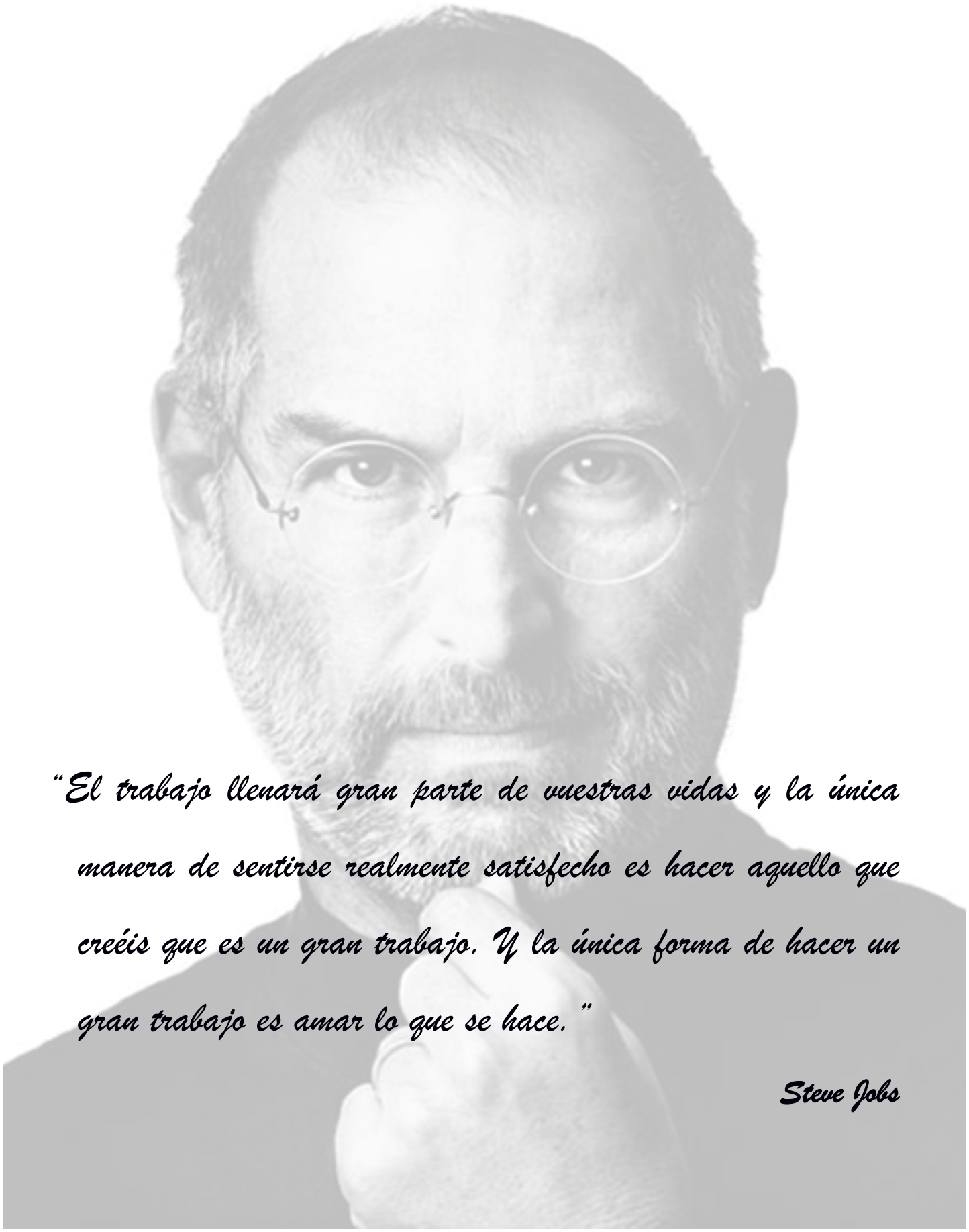
Título: Módulo Configuración entidad para el Sistema de Información Hospitalaria del Centro de Informática Médica

Autores: Bilmarys González Leal
Ivey León Muñoz

Tutores: Ing. Anny Campos Cosme
Ing. Osmín Pérez Morales

La Habana, junio de 2015.

"Año 57 de la Revolución"



"El trabajo llenará gran parte de vuestras vidas y la única manera de sentirse realmente satisfecho es hacer aquello que creéis que es un gran trabajo. Y la única forma de hacer un gran trabajo es amar lo que se hace."

Steve Jobs

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los __ días del mes de ___ del año ____.

_____
Autora: Bilmarys González Leal

_____
Autor: Ivey León Muñoz

_____
Tutor: Ing. Osmín Pérez Morales

_____
Tutora: Ing. Anny Campos Cosme

Datos de Contacto

Tutores:

Ing. Anny Campos Cosme: Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2010. Ha participado en varios proyectos de desarrollo vinculados al perfil de salud. Actualmente labora en el Departamento de desarrollo de componentes del Centro de Informática Médica (CESIM), desempeñándose como Analista del mismo.

Correo Electrónico: acampos@uci.cu

Ing. Osmín Pérez Morales: Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2010. Se ha desempeñado como Desarrollador de Software en el Centro de Informática Médica (CESIM). Ha participado en varios proyectos de desarrollo vinculados al perfil de salud y ha sido tutor de otros trabajos de diploma.

Correo Electrónico: opmorales@uci.cu

Agradecimientos:

Bilmarys:

Agradezco a mis padres por haberme dado la vida, guiarme hacia el camino correcto, brindarme amor, educación, pero sobre todo confianza.

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

A mi novio por su comprensión y apoyo.

A todos mis amigos de la universidad por compartir tantos momentos juntos, en especial a una personita que más que amiga fue hermana en estos 5 años, Dayana.

A mis tutores y profesores por la ayuda brindada.

A mi compañero de tesis porque siempre supo sobreponerse a las dificultades de la tesis y por hacer junto conmigo que este trabajo saliera adelante.

A todos aquellos que de una forma u otra estuvieron ahí para ayudarme en algún momento, gracias.

Agradecimientos

Ivey:

Quiero agradecer a mis padres Lili e Ivey por guiarme hacia el camino correcto ofrecerme amor, educación, y mucha dedicación. Sin ellos, no hubiese podido alcanzar esta meta.

A mi otra madre Ivonne por su apoyo incondicional y estar siempre presente en todos los momentos.

A mis hermanas Lilibet, Lorena y Layra por apoyarme en todo, compartir las alegrías y tristezas siempre juntos.

A mis abuelos Osvaldo, Ela, Mercedes por su cariño incondicional y siempre impulsarme a ser cada día mejor.

A todos mis tíos por cada pedacito de esfuerzo y dedicación.

A mi familia en general por estar siempre que los he necesitado.

A mi novia por su amor, y por estar a mi lado en los momentos buenos y malos.

A mis compañeros de Universidad por haber compartido con ellos tantos momentos inolvidables durante estos años.

A mi compañera de tesis por su dedicación, y por hacer junto conmigo que este trabajo saliera adelante.

A mis tutores y profesores que contribuyeron a la realización de este trabajo.

Dedicatoria:

Bilmarys:

A mis padres que han sido mis guías en todo momento, a mis hermanos por ser lo más grande que tengo.

Ivey:

Dedicado a mi padre, por ser mi meta, mi guía y por ser aquella mano que me levanta cuando más lo necesito. A mi madre, Ivonne y mis hermanas por quererme tanto, y por regalarme tanta felicidad.

Resumen

El Centro de Informática Médica de la Universidad de las Ciencias Informáticas desarrolla el Sistema de Información Hospitalaria. Para la administración del mismo se emplea el módulo Configuración, en el que se maneja información referente a las áreas existentes en una institución hospitalaria.

El módulo Configuración presenta ciertas dificultades en la gestión de las entidades[1] debido a que se encuentran mezcladas la administración del sistema y la específica de una entidad; afectando así la usabilidad, seguridad y el proceso de implantación del sistema. Por estas razones se propone la implementación del módulo Configuración entidad, que permita la administración de forma individual de cada una de las entidades del sistema.

El desarrollo del sistema está guiado por el Proceso Unificado de Desarrollo. Como plataforma de desarrollo se utiliza Java Enterprise Edition. Se usa Java como lenguaje de programación y se implementa el patrón de arquitectura Modelo Vista Controlador. Como gestor de bases de datos se tiene PostgreSQL, Hibernate como herramienta ORM para la persistencia de los datos, el framework Seam para facilitar la comunicación entre el negocio y la presentación y como servidor de aplicaciones el JBoss Server.

Una vez desarrollado el módulo se espera contribuir a la mejora del proceso de configuración de las entidades del sistema, así como la gestión de la información relacionada con: personal asociado, departamentos, servicios y especialidades, clínicos y no clínicos. Además brindar la gestión de la *permisología* sobre módulos y funcionalidades.

Palabras claves: Configuración, Entidad, Sistema de Información Hospitalaria, Gestión de la información.

[1] Parte integrada de una organización médica y social, cuya misión consiste en proporcionar a la población una asistencia médico sanitaria completa, tanto curativa como preventiva. Además es un centro de formación de personal médico sanitario.

Índice de Contenidos

Índice de Contenidos

Índice de Figuras

Índice de Tablas

Introducción

En los últimos años el desarrollo económico, científico y técnico impone nuevas metas a la sociedad. Avances vertiginosos de la ciencia, en especial los que responden a las Tecnologías de la Información y las Comunicaciones (TIC), definen la actual era tecnológica. El auge que ha tomado la informatización de los procesos involucrados en las distintas esferas de la sociedad apuesta por la inclusión de mejores métodos con el uso de la informática y las comunicaciones, para responder las necesidades existentes.

En la medicina, los avances de la ciencia han sido aplicados en numerosos campos, pero no se trata solo de la actualización tecnológica por la adquisición de equipamientos modernos, es indispensable contar con una herramienta que permita almacenar y tratar la información. Surge entonces lo que se conoce como informática médica: disciplina que aplica las metodologías desarrolladas en diferentes áreas del conocimiento científico a las múltiples tareas orientadas, recopilando datos en el momento que se generan y ofreciéndolos para la toma de decisiones administrativas, de investigación, diagnóstica y terapéutica. (1)

Desde el surgimiento de este concepto se han desarrollado disímiles sistemas informáticos aplicados a cada una de las áreas de la medicina, debido a la gran cantidad de datos que se genera constantemente en las instituciones de salud. Dentro de los más destacados se encuentran los Sistemas para el Almacenamiento, Visualización y Transmisión de Imágenes Médicas (PACS, por sus siglas en inglés), Sistema de Información Radiológica (RIS, por sus siglas en inglés), Sistemas de Información de Laboratorios (LIS, por sus siglas en inglés) y Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés). Un HIS es un sistema que centraliza toda la información generada por los distintos servicios del hospital a partir de un mismo paciente (2), lo que ha ofrecido a los profesionales de la salud una alternativa que permite: lograr la optimización de recursos humanos y materiales, así como preservar la información que se maneje acerca de los pacientes.

El Centro de Informática Médica (CESIM) perteneciente a la Universidad de las Ciencias Informáticas (UCI), en el año 2010 desarrolló un sistema de información hospitalaria. Esta aplicación está conformada por un conjunto de módulos que responden a cada una de las áreas que se pueden encontrar en una institución hospitalaria, entre estos se encuentran Admisión, Citas, Consulta Externa, Hospitalización, Laboratorio, Banco de Sangre, entre otros. Para la

administración del sistema se emplea el módulo Configuración, en el que se manejan diferentes conceptos, dígase departamentos, servicios y especialidades clínicos y no clínicos, personal asociado, roles, niveles de acceso a la información, activación de funcionalidades, módulos y otros.

La experiencia adquirida con el despliegue e implantación de la aplicación en instituciones hospitalarias y su posterior utilización por los usuarios finales de la misma, reflejan que la estructura del módulo Configuración supone ciertas dificultades pues se encuentran mezcladas las funcionalidades asociadas a la administración del sistema y las específicas de una entidad. Partiendo de la idea de obtener un sistema multientidad y el desarrollo progresivo del mismo, se recargó de funcionalidades que afectan su usabilidad y seguridad.

La organización que presenta el módulo Configuración trae consigo que todo el personal que tenga acceso a este, independientemente de la entidad a la que pertenezca, puede visualizar la información de todas las entidades del sistema. Esta situación dificulta la seguridad de la aplicación, porque queda a consideración de ese usuario, gestionar o no información de las instituciones en las que no tiene privilegios.

La gestión de la seguridad en el sistema se realiza mediante la administración de la *permisología* de usuarios y roles sobre los módulos y funcionalidades. Estos permisos se dividen en lógicos y físicos; donde los lógicos permiten que se muestren o no las funcionalidades y los físicos controlan que se ejecuten las páginas correspondientes. Para realizar la asignación de privilegios se utilizan dos vistas donde se listan todos los usuarios y roles existentes. Esto requiere que en el proceso de asignar los mismos, el administrador deba conocer a priori a que entidad pertenece cada usuario en caso contrario, debe acceder a otra funcionalidad para conocer esta información. Conocer los permisos que tiene asignado tanto un rol como un usuario es un proceso engorroso porque se debe seleccionar cada funcionalidad del sistema y buscar en los listados que se muestran, el estado de la *permisología* asociado a estos, restándole facilidad de interacción con la aplicación.

Es válido señalar además, que en el momento de creación de los usuarios es preciso especificar el tipo de funcionario. Esto trae consigo que si el usuario es un médico interno o residente, se le deba asignar una especialidad de graduación y manejarlo como especialista, para que pueda realizar la atención a los pacientes quedando registrada una atención médica tomando al facultativo como especialista sin serlo.

Otro de los inconvenientes detectados es que las funcionalidades son comunes para todo el sistema, por lo que no pueden ser excluidas de una entidad que lo solicite, de hacerlo se estaría eliminando la funcionalidad para todas las entidades.

Por otra parte el módulo contempla entre sus funcionalidades la gestión de la información referente a: departamentos, servicios y especialidades con que cuentan las instituciones hospitalarias. Los departamentos configurados tienen asociados servicios y estos a su vez especialidades. La estructura de estos conceptos es única en el sistema, lo que implica que de ser necesario realizar un cambio de esta en alguna entidad, el mismo afecta a todas por igual.

Tomando en cuenta la situación descrita anteriormente, se identifica el siguiente **problema a resolver:** La estructura del módulo Configuración del Sistema de Información Hospitalaria del CESIM, dificulta la gestión de la información referente a las entidades hospitalarias.

Teniendo como **objeto de estudio,** el proceso de configuración en sistemas de gestión de información hospitalaria.

El **campo de acción** está centrado en: el proceso de configuración de entidades, del Sistema de Información Hospitalaria del CESIM.

Para dar solución al problema planteado se propone como **objetivo general** Desarrollar el módulo Configuración entidad para el Sistema de Información Hospitalaria del CESIM, que permita la correcta gestión y administración de las entidades.

Con el fin de dar cumplimiento al objetivo planteado se concretan las siguientes **tareas de la investigación:**

1. Analizar los principales procesos de configuración de entidades hospitalarias para los sistemas de información hospitalaria.
2. Definir las funcionalidades del módulo Configuración entidad del Sistema de Información Hospitalaria del CESIM.
3. Asimilar la arquitectura y pautas definidas para el desarrollo de los artefactos, el diseño y las aplicaciones del Sistema de Información Hospitalaria del Centro de Informática Médica.
4. Desarrollar los artefactos asociados a las disciplinas: Modelado de negocio, Gestión de

requisitos, Diseño e Implementación correspondientes al Proceso Unificado de Desarrollo.

5. Implementar el módulo Configuración entidad del HIS del Centro de Informática Médica.

De este modo se espera que el desarrollo del módulo Configuración entidad para el Sistema de Información Hospitalaria del CESIM ofrezca los siguientes beneficios:

1. Configuración de la entidad para el Sistema de Información Hospitalaria del CESIM que agilice el proceso de implantación del mismo.
2. Facilidad de entendimiento de las funcionalidades encargadas de la configuración de las entidades para los administradores.
3. Mejora de la seguridad y usabilidad del Sistema de Información Hospitalaria del CESIM.

El contenido del presente documento se encuentra estructurado en 4 capítulos:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN. Contiene el marco conceptual en el que se muestran las principales definiciones utilizadas en la investigación. Incluye un estudio del estado del arte. Además se hace referencia a la metodología de desarrollo, las herramientas y lenguajes de programación que apoyen el desarrollo de la solución.

CAPÍTULO 2: CARACTERÍSTICAS DEL MÓDULO CONFIGURACIÓN ENTIDAD. Describe la propuesta de solución. Contiene el marco conceptual relativo a la información que se maneja en el sistema.

CAPÍTULO 3: DISEÑO DEL MÓDULO CONFIGURACIÓN ENTIDAD. Se describe la arquitectura del Sistema de Información Hospitalaria y cómo se enmarca dentro de la misma la propuesta de solución. Muestra el patrón arquitectónico y los de diseño que se utilizan en el proceso de desarrollo de la solución.

CAPÍTULO 4: IMPLEMENTACIÓN DEL MÓDULO CONFIGURACIÓN ENTIDAD. En este capítulo se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los requisitos de seguridad y la configuración de una entidad para el Sistema de Información Hospitalaria del Centro de Informática Médica.

Capítulo 1. Fundamentación teórica de la investigación

Este capítulo está dedicado a realizar un estudio de los principales conceptos y características que constituyen las bases para el desarrollo del módulo Configuración entidad para el Sistema de Información Hospitalaria del CESIM. Se presentan un conjunto de temas relacionados con la configuración de los sistemas de información hospitalaria así como la tecnología definida para el Sistema de Información Hospitalaria con las que se lleva a cabo el proceso de desarrollo.

1.1 Descripción específica del módulo

El módulo Configuración garantiza la seguridad de acceso al sistema, permitiendo a su vez la administración de usuarios y roles. Gestiona cada uno de los departamentos, servicios, ubicaciones, camas, así como los datos generales de la institución hospitalaria. Entre las funcionalidades principales de la configuración de las entidades del sistema se encuentran:

- Administrar seguridad: define los permisos y niveles de acceso por roles o usuarios a las funcionalidades de cada uno de los módulos. Se puede permitir o denegar el acceso a una funcionalidad en específico, en caso de haber otorgado permiso al módulo, el usuario tendrá acceso a todas las funcionalidades de este. En la vista física se muestra el directorio que contiene los módulos y funcionalidades de la entidad en la que se encuentra trabajando. Se debe administrar la seguridad en la vista lógica y en la vista física. En ambas vistas el procedimiento para permitir o denegar usuarios y roles es el mismo.
- Configurar funcionalidades: posibilita visualizar todos los módulos y submódulos existentes mostrando de cada uno de ellos sus funcionalidades que pueden estar agrupadas por categorías. Permite adicionar, modificar, eliminar, editar el orden y cambiar la visibilidad de las funcionalidades y categorías, así como ver los datos de las funcionalidades. Estas funcionalidades son comunes para todas las entidades en el sistema.
- Usuarios y roles: permite gestionar los usuarios y roles del sistema, así como los médicos, enfermeros y bioanalistas, brindando la posibilidad de crearlos, modificarlos, ver sus datos y eliminarlos.
- Departamentos clínicos: permite crear departamentos y servicios, de cada uno de ellos se puede modificar la información asociada o eliminar la misma. Los servicios creados en el sistema son asociados a los diferentes departamentos según sea la distribución de la

institución, posibilita además realizar búsquedas de un servicio en específico o de los servicios asociados a un determinado departamento.

- Otros departamentos y servicios: permite gestionar los departamentos y servicios no clínicos presentes en el sistema.
- Gestionar ubicaciones: permite definir los tipos de ubicaciones en el hospital como: Ala, Piso, Habitación o Sala, definiendo código y descripción para cada uno. De igual forma se visualiza jerárquicamente la estructura de la institución, especificando las habitaciones por sala, las salas por piso y los pisos por ala del hospital. Existe además la posibilidad de la generación del identificador de las ubicaciones.
- Gestionar camas: permite crear las camas de la institución, especificando datos como: Descripción, Tipo de cama, Estado de cama, Habitación y Categoría. Esta información puede ser modificada o eliminada según el interés del usuario. Además brinda la posibilidad de organizar las camas por ubicaciones o servicios y muestra el mapa de las ubicaciones físicas de las camas existentes.

1.2 Sistemas informáticos existentes vinculados al campo de acción

Se realiza un análisis de los diferentes sistemas informáticos vinculados al sector de la salud los cuales forman parte del estado del arte del trabajo, haciendo énfasis en los aspectos relacionados con la configuración de la entidad, con el objetivo de determinar soluciones eficaces capaces de brindar respuesta a los problemas existentes en la gestión de la información necesaria para usar el sistema.

Los principales sistemas informáticos analizados son los siguientes:

Sistema Integral de Gestión Hospitalaria (SIGH): ofrece la automatización integral en procesos hospitalarios, está compuesto por una serie de módulos o paquetes informáticos que permiten la gestión sistematizada en cada una de las áreas que conforman a la institución médica, aportando automatización en sus procesos y construyendo la información necesaria e integrada, para la toma de decisiones, tanto para las áreas médicas como para el personal administrativo de diferentes clínicas, centros médicos y hospitales.

Capítulo 1

El Módulo de Configuración y Seguridad comprende un alto control y manejo del acceso a cada uno de los módulos del Sistema Integral de Gestión Hospitalaria, garantizando mayor seguridad y control a la información Médica de la institución. (3)

Hosix-V: es un sistema de gestión e información hospitalaria flexible y modular, que abarca todas las áreas de actividad de un hospital. Es un sistema desarrollado en módulos orientados a la gestión específica de cada servicio o departamento, cubriendo los aspectos más importantes de una unidad de salud. (4)
Sus servicios relativos a seguridad se dividen en dos vertientes: (5)

1. Consultoría de Seguridad
   - Implantación de elementos de seguridad física
   - Puesta en marcha de medidas de seguridad lógica
   - Revisión de la Política de Seguridad definida en la organización

2. Auditorías de Seguridad
   - Auditoría interna
   - Auditoría perimetral
   - De aplicaciones web

SISalud: se considera un portal de aplicaciones para el Sistema Nacional de Salud en Cuba que conforma su estructura a partir de los derechos del usuario que se autentica, dándole solamente la posibilidad de acceso a los módulos o aplicaciones que éste requiera y mostrándole además los avisos que le corresponden. Su estructura integra diferentes módulos que se relacionan con las siguientes áreas:

Administración del sistema: garantiza la seguridad en el acceso a determinado nivel de profundidad y derechos de edición por medio de la autenticación, autorización y auditoría (AAA). Ofrece prestaciones según tipo de usuario y nivel de gestión por medio de contraseñas y certificados digitales.

Registros básicos y codificadores: son módulos que registran datos que son de necesidad común a otros módulos y favorecen la estandarización de la información. La información es administrada y

gestionada a nivel nacional o central. Contiene todos los componentes del registro informatizado de la salud clasificados.

Administración de salud: son los módulos que responden a otros sistemas de gestión e información de salud que no forman parte directamente de los niveles de atención médica. Por ejemplo: Medicamentos y Fármacos, Balance Material, Economía, entre otros. (6)

Phphclinica: el sistema está conformado por trece módulos donde dos responden a la configuración y administración del sistema. Además cuenta con cincos niveles de seguridad de usuarios para que estos no tengan los mismos privilegios a la hora de interactuar con el sistema.

Módulo Configuración: en este módulo se puede configurar el sistema para su uso, todos los datos que se configuran aquí son utilizados en los diferentes formularios de sus módulos. Además se podrá registrar nuevos usuarios para el sistema.

Módulo de Administración: La función principal de este módulo es controlar los datos de seguridad del sistema como son las trazas que deja el mismo para mantener la seguridad en los datos que son registrados.

Interacción de los módulos: Ochos de los treces módulos del sistema interactúan entre sí mediante dos claves únicas (Historia Clínica y Número de Identidad) que hacen que los datos de eso módulos específicos estén relacionados, los demás módulos cumplen diferentes funciones en todo el sistema. (7)

Valoración de los sistemas analizados

Una vez analizada la configuración actual del Sistema de Información Hospitalaria del CESIM y la problemática existente, se determina la necesidad de separar las configuraciones relacionadas con la administración general del sistema de las concernientes a la configuración de una entidad de salud. Además se realiza un estudio de algunos sistemas informáticos vinculados al campo de acción que sirven de base de conocimiento para la definición de las funcionalidades y características que debe tener la solución que se propone. De este modo el estudio de dichos sistemas permite tener un punto de referencia para el desarrollo propuesto.

1.3 Tecnología a utilizar

Java Enterprise Edition 6 (JEE):

Es una plataforma de programación distribuida para ejecutar y desarrollar software de aplicaciones en lenguaje de programación Java, desarrollada por SunMicrosystem. Esta es un conjunto de librerías que establecen un estándar para lograr un producto altamente calificado. Permite el manejo de diversos detalles mediante una programación simple y al no ser privativa, el sistema que se desarrolle usando Java puede ser comercializado en el mundo entero. (8)

Seam 2.1.1

Seam es un marco de trabajo desarrollado para Java EE, creado para la realización de aplicaciones Web 2.0 de forma fácil, soportando una arquitectura unificada de componentes. Permite la integración de tecnologías como AJAX, Java Server Faces (JSF), Enterprise Java Beans (EJB3), java Portlets, Business Process Management (BPM), Hibernate y Java Persistence API (JPA) en una única solución. Contextos que soporta:

- Contexto de página.
- Contexto de aplicación.
- Contexto de evento.
- Contexto de sesión.
- Contexto de conversación. **(9)**

Interfaz de Usuario Seam 2.1.1 GA (Seam UI)

Serie de controles Java Server Faces (JSF) altamente integrables con JBoss Seam. Es utilizado para complementar los controles JSF incorporados y los controles de otras bibliotecas externas. (10)

Facelets 1.1.15.B1

Es un *framework* simplificado de presentación, en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. (11)

JBoss AS 4.2.2 como servidor de aplicaciones

JBoss Application Server es el servidor de aplicaciones de código abierto. Soporta todas las especificaciones correspondientes, incluyendo servicios adicionales como clusterizar, carga en memoria caché y persistencia, por ser una plataforma con certificación JEE 5. También soporta Enterprise Java Beans (EJB) 3.0. Además, al ser desarrollado con tecnología Java, es multiplataforma. (12)

JBoss Tools 3.0.0.CR1

Es un conjunto de plug-ins de Eclipse que tiene como objetivo ayudar a los desarrolladores a crear aplicaciones webs de forma rápida y sencilla. (13)

Los módulos que presenta JBoss Tools son:

- RichFaces VE: Editor visual proporcionado por Exadel. Brinda el apoyo para la edición visual de páginas HTML, JSF, JSP y Facelets. También incluye soporte visual para las librerías de componentes JSF incluyendo JBoss RichFaces.
- Seam Tools: Incluye soporte para seam-gen, RichFaces VE.
- Hibernate Tools: Soporta el mapeo de archivos, anotaciones y JPA con la ingeniería inversa, completamiento de código, asistentes de proyecto, refactorización.
- JBoss AS Tools: Fácil de iniciar, detener y analizar ejecución paso a paso al estar integrado con Eclipse. También incluye funciones para el despliegue eficaz de cualquier tipo de proyecto en el IDE.
- JBPM (Java Business Process Management) Tools: Edición del flujo de trabajo del JBPM, motor de procesos BPM.
- JBossWS (JBoss Web Services) Tools: Desarrollo, invocación, inspección y pruebas de servicios web sobre http con la adición y soporte de características JBossWS.”

Java Server Faces (JSF) 1.2

Es una librería de interfaz de usuario para aplicaciones Web implementadas con Java. Fue diseñado para aliviar la carga de desarrollo y mantenimiento de aplicaciones que se ejecutan en Servidores de aplicaciones Java y prestar sus interfaces de usuario a un cliente objetivo. (14)

Ajax4JSF

Ajax4js es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas, dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript. Con el uso de este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, entre otras. (11)

Richfaces 3.3.0.GA

Richfaces es una librería de código abierto que añade los componentes de Ajax (Ajax4jsf) en aplicaciones existentes de JSF sin recurrir a JavaScript. Aprovecha la librería de Java Server Faces para incluir validaciones, instalaciones de conversión y la gestión de los recursos estáticos y dinámicos. De esta forma permite al desarrollador ahorrar tiempo y aprovechar las características de los componentes para crear aplicaciones Web, con mejor apariencia visual. (15)

Hibernate 3.0

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL. (11)

Java Persistence API (JPA) 3.0

Java Persistence API (JPA) proporciona un modelo de persistencia basado en POJO's para mapear bases de datos relacionales en Java. El Java Persistence API fue desarrollado por el grupo de expertos de EJB 3.0, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes; incluso fuera de la plataforma Java EE, por ejemplo, en aplicaciones Java SE. En su definición, se han combinado ideas y conceptos de las principales librerías de persistencia como Hibernate, Toplink y JDO, y de las versiones anteriores de EJB. (16)

Enterprise Java Beans 3 (EJB)

La especificación de Enterprise Java Beans, define una arquitectura para un sistema transaccional de objetos distribuidos basados en componentes. La especificación posee un modelo de programación; convenciones o protocolos y un conjunto de clases e interfaces que crean el API EJB. (8)

Java

Java es un lenguaje de programación orientado a objetos que está inspirado en la sintaxis de C++, pero su funcionamiento tiene mayor semejanza al de Smalltalk que a este. Es un lenguaje compilado e interpretado. Este lenguaje facilita la creación de aplicaciones distribuidas proporcionando una colección de clases para ser usadas en aplicaciones de red. El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos por lo que la fase de enlazado hace de Java además un lenguaje de alto rendimiento. (17)

XHTML 1.0

Los documentos XHTML están basados en XML, una familia de módulos y tipos de documentos que reproduce, engloba y extiende el HTML 4.0. Los tipos de documentos de esta familia están diseñados fundamentalmente, para trabajar en conjunto con aplicaciones de usuario basados en XML.

Ventajas que presenta:

- Son conformes a XML. Por lo que son fácilmente visualizados, editados y validados con herramientas XML estándar.
- Pueden escribirse para que funcionen igual, o mejor que lo hacían antes, tanto en las aplicaciones de usuario conformadas con HTML 4.0, así como en las nuevas aplicaciones conformes a XHTML.
- Pueden usar aplicaciones que se basen, ya sea en el Modelo del Objeto Documento de HTML o XML. **(18)**

JavaScript

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web. JavaScript se usa generalmente en su forma del lado del cliente, implementado como parte de un navegador web y permite mejoras en la interfaz de usuario y

páginas web dinámicas, aunque también presenta una forma del lado del servidor. El código JavaScript puede detectar acciones de los usuarios que HTML por sí sola no puede, ejemplo de ello son las pulsaciones de teclado. (8)

RUP (Racional Unified Process). Proceso Unificado Racional

RUP es una metodología adaptable al contexto y necesidades de cada organización cuyo fin es entregar un producto de software. Además es un proceso de desarrollo de software el cual utiliza el lenguaje unificado de modelado UML. Constituye así la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. En el desarrollo del proyecto se hace mayor énfasis en los flujos de trabajo de Modelado de Negocio, Gestión de Requisitos, Análisis y Diseño e Implementación obteniendo los artefactos correspondientes a cada uno de estos flujos.

El ciclo de vida de RUP se caracteriza por:

- Dirigido por casos de uso: los casos de uso reflejan lo que el cliente necesita y desea, lo cual se capta al modelar el negocio y se representa a través de los requisitos. A partir de aquí los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: la arquitectura muestra la visión común del sistema en la que el equipo de desarrollo y los clientes deben de estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- Iterativo e Incremental: cada fase se desarrolla en iteraciones, una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla algunos más que de otros. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos, donde cada uno de ellos es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. **(19)**

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

UML es un lenguaje de representación visual que permite combinar diversos elementos gráficos y crear diagramas. Se usa para modelar sistemas y usa tecnología orientada a objetos. El lenguaje unificado de modelado describe lo que hará un sistema pero no dice cómo implementarlo. Su

objetivo es visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo. Involucra todo el ciclo de vida del proyecto y está pensado para varios lenguajes y plataformas tales como ASP, PHP, entre otros. (20)

JBoss Developer Studio 8.1

JBoss Developer Studio es un entorno de desarrollo integrado (IDE) certificado y basado en Eclipse para desarrollar, probar e implementar aplicaciones web avanzadas, aplicaciones web móviles, aplicaciones empresariales transaccionales y aplicaciones y servicios de integración basados en la arquitectura orientada a servicios (SOA).

JBoss Developer Studio incluye un amplio conjunto de herramientas y soporte para varios modelos y marcos de programación, entre los que se incluyen:

- Java Enterprise Edition 5
- RichFaces
- Java Server Faces (JSF)
- Enterprise JavaBeans (EJB)
- Java Persistence API (JPA) e Hibernate **(21)**

PostgreSQL 8.4 como servidor de base de datos

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Entre sus principales características se encuentran:

- Extensible: Su código fuente está disponible para todos sin costo. Si un equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales.
- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable). Diseñado para ambientes de alto volumen. **(22)**

PgAdmin como aplicación cliente para manejar la base de datos

PgAdmin III es una aplicación gráfica para trabajar con el gestor de bases de datos PostgresSQL. Es la más completa y popular con licencia Open Source. Está escrita en C++. Usa la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Su interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad. (23)

Visual Paradigm 8.0 como herramienta de modelado

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML contribuye a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Se integra con IDE's como NetBeans (de Sun Microsystems), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland). Su diseño está centrado en casos de uso y se usa lo mismo para UML que para BPMN (Business Process Modeling Notation). (15)

Java virtual machine

La Máquina Virtual Java es el núcleo del lenguaje de programación Java. Se encarga de interpretar todo el código java y convertirlo al lenguaje nativo del sistema operativo en uso (8). La utilización de la máquina virtual java permite la portabilidad de la aplicación.

1.4 Conclusiones del capítulo

- Para garantizar la correcta gestión de la información de las entidades del sistema, se determina separar las configuraciones relacionadas con la administración general del sistema de las concernientes a la configuración de una entidad de salud.

- Las funcionalidades que se implementan en Configuración entidad, cumpliendo con las tendencias actuales, se basan en el proceso de configuración de las entidades de los sistemas estudiados.
- La tecnología definida para el Sistema de Información Hospitalaria del CESIM, permite la obtención de un módulo flexible y multiplataforma.

Capítulo 2. Características del módulo Configuración entidad.

En este capítulo, con motivo de especificar el contexto en el cual se desarrolla el sistema, se muestra el modelo de dominio, donde se exponen conceptos fundamentales relacionados con el sistema y las relaciones entre ellos. Se plantean además los requisitos de software funcionales, no funcionales y se describen los casos de uso del sistema.

2.1 Modelo de dominio

El Modelo de Dominio, o Modelo Conceptual, captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará embebido el sistema. Se utiliza para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. Contiene conceptos que estarán asociados tanto a su definición natural como al papel que juegan desde el punto de vista informático. (24)

En el sistema el administrador general es el encargado de gestionar las entidades las cuales son establecimientos destinados al diagnóstico y tratamiento de los individuos que padecen una determinada enfermedad. Estas entidades tienen módulos que representan las diferentes áreas de un hospital y estos a su vez contienen funcionalidades que permiten realizar una acción particular en el sistema y que pueden ser agrupadas en categorías.

El administrador general gestiona además los usuarios y roles que pueden estar permitidos o denegados en las funcionalidades las cuales, son comunes para todas las entidades. Otros de los conceptos que se manejan en el modelo de dominio son los referentes a departamentos clínicos y no clínicos que son sectores de la unidad de la organización hospitalaria que tienen asociados servicios y especialidades cuyo objetivo es prestar atención especializada a pacientes hospitalizados. A continuación se muestra el diagrama del modelo de dominio con sus respectivos conceptos y relaciones.

2.1.1 Diagrama del modelo de dominio

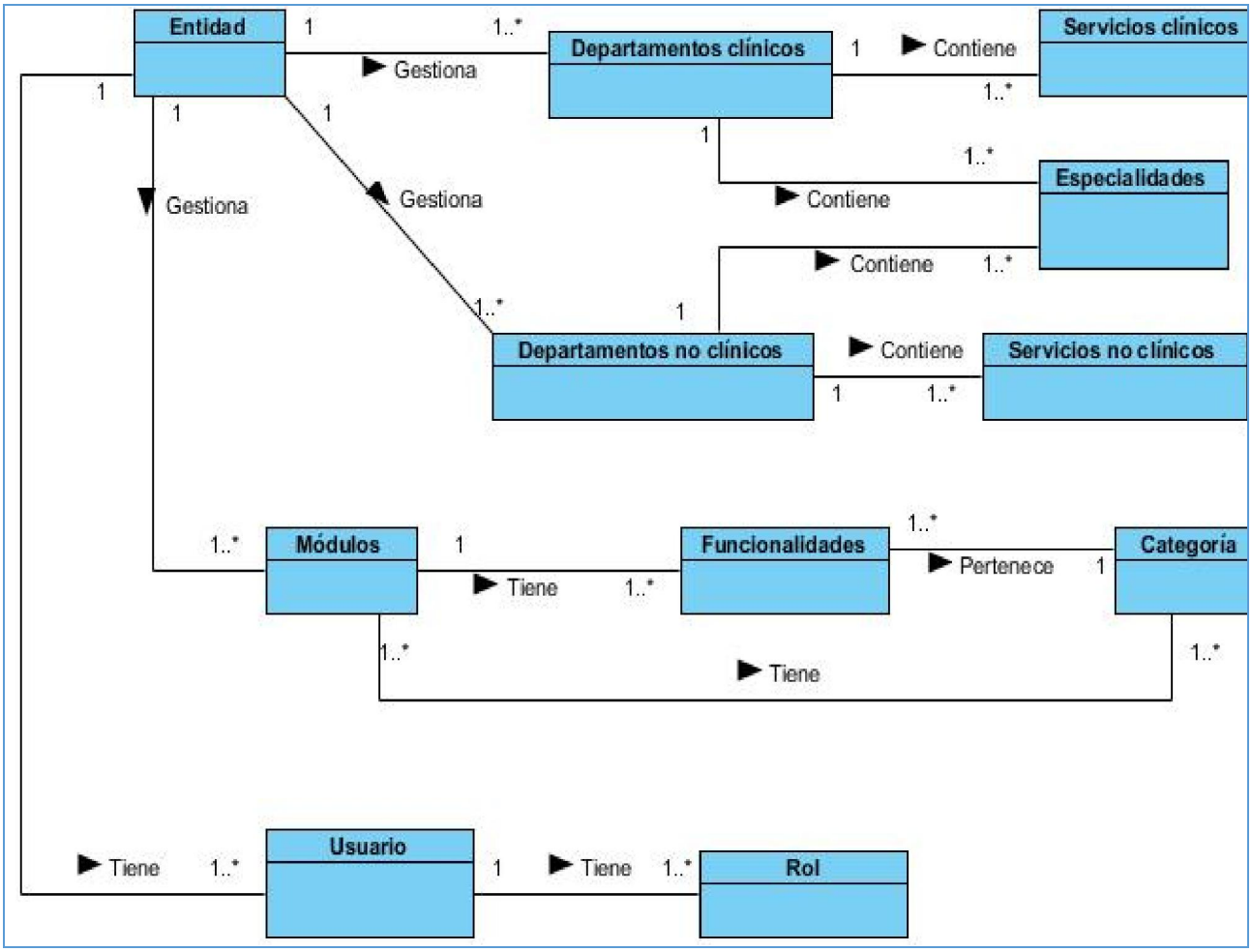


Figura 1. Modelo conceptual

2.2 Propuesta del sistema

Fueron identificadas las funcionalidades del módulo, quedando distribuidas de la siguiente manera: de un total de 227 funcionalidades que tiene el módulo Configuración del sistema, 147 responden a la configuración de las entidades. De ellas se modificaron 69 y 51 fueron agrupadas en 22 funcionalidades.

El módulo Configuración entidad tiene como objetivo principal lograr la gestión de la información previa y necesaria para la configuración de una entidad en el Sistema de Información Hospitalaria del CESIM. La configuración que proporciona el mismo, permite la gestión de la información

relacionada con la administración del personal asociado a la entidad, la cual implica la creación de roles y cuentas de usuarios que utilizan el sistema. Además se garantiza la seguridad de la entidad gestionando la permisología, sobre módulos y funcionalidades, tanto lógica como física. Otra de las funcionalidades del módulo es la gestión de la información referente a los departamentos, servicios y especialidades, clínicos y no clínicos, con que cuente la institución hospitalaria.

Configuración entidad brinda la posibilidad de configurar funcionalidades y categorías lo que permite visualizar todos los módulos y submódulos existentes en la entidad mostrando de cada uno de ellos sus funcionalidades y categorías; brindando la posibilidad de adicionar las funcionalidades previamente configuradas en el módulo Configuración general. Luego de añadidas las funcionalidades estas pueden ser ocultadas o eliminadas.

2.3 Especificación de los requisitos del software

Los requisitos de software suelen ser la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes. Tienen varias clasificaciones entre las que se encuentran, los requisitos funcionales y los requisitos no funcionales.

Requisitos funcionales: son declaraciones de los servicios que debe proporcionar el sistema. Describen lo que el sistema debe hacer.

Requisitos no funcionales: son restricciones de los servicios o funciones ofrecidos por el sistema. No se refieren directamente a las funciones específicas que propiciará el sistema sino a las propiedades emergentes del mismo. (25)

2.3.1 Requisitos funcionales

Tabla 1. Requisitos no funcionales

RF1- Configurar módulo	RF20-Ver detalles de usuario
RF2- Adicionar módulo	RF21-Buscar rol
RF3-Adicionar submódulo	RF22-Adicionar rol
RF4-Eliminar módulo	RF23-Ver detalles de rol
RF5-Modificar módulo	RF24-Modificar rol
RF6-Seleccionar módulo	RF25-Eliminar rol
RF7-Seleccionar submódulo	RF26-Administrar departamentos.
RF8-Configurar funcionalidades	RF27- Adicionar departamento clínico
RF9-Adicionar categoría	RF28- Modificar departamento
RF10-Eliminar categoría	RF29- Eliminar departamento clínico
RF11-Adicionar funcionalidad	RF30- Adicionar departamento no clínico
RF12-Eliminar funcionalidad	RF31- Eliminar departamento no clínico
RF13-Administrar seguridad	RF32- Adicionar servicio
RF14-Seleccionar funcionalidad	RF33- Modificar servicio
RF15-Buscar usuario	RF34- Eliminar servicio
RF16-Adicionar usuario	RF35- Adicionar especialidad
RF17-Modificar usuario	RF36- Modificar especialidad
RF18-Ver datos de usuario	RF37- Eliminar especialidad
RF19-Eliminar usuario	

2.3.2 Requisitos no funcionales:

Usabilidad: El módulo brinda un acceso rápido y fácil a sus funcionalidades y la representación de los textos y los íconos es comprensible para los usuarios que interactúan con el mismo facilitando su uso.

Interfaz: las ventanas del sistema contienen los datos claros y bien estructurados. La interfaz cuenta con teclas de función y menús desplegables que facilitan y aceleran su utilización. La entrada de datos incorrecta es detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecen en el idioma previamente configurado.

Seguridad: Se mantendrá la seguridad y control a nivel de usuarios y roles, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo con la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema. Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

Rendimiento: el sistema optimiza el uso de recursos tales como la memoria y respeta buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

Hardware recomendado:

Estaciones de trabajo

En la solución se recomiendan estaciones de trabajo para la interacción con el Sistema de Información Hospitalaria de 1 GB de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo Windows o Linux.

Servidores

Por otra parte los servidores deben tener 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 Aplicaciones: HDD: 80 GB, TR: 100 Mbps, RAM: 16 GB, CPU: 8 núcleos, Sistema Operativo Linux, JBoss AS 4.2.2 GA, Java Runtime Enviroment (JRE).
- Servidores de base de datos: PostgresSQL Server 8.4, HDD: 80 GB, TR: 100 Mbps, RAM: 8 GB, CPU: 8 núcleos, Sistema operativo Linux.

Software: el módulo forma parte del Sistema de Información Hospitalaria del CESIM, por lo que puede ser desplegado en los sistemas operativos Windows 7 o superior y Linux, utilizando Java Runtime Environment, JBoss Server y PostgreSQL. El usuario debe disponer de un navegador web, el cual puede ser, Google Chrome, Firefox 4.0 o versiones superiores de estos.

Portabilidad: el sistema puede ser desplegado en los sistemas operativos Windows y Linux.

2.4 Modelo de casos de uso del sistema

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un caso de uso es una secuencia de interacciones que se desarrollan entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. Los diagramas de casos de uso se utilizan para ilustrar los requisitos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo. (26)

2.4.1 Vista global de actores

Figura 2. Actores del sistema

2.4.2 Definición de actores

Un actor es una entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad.

Tabla 2. Actores del sistema

Actor	Descripción
Administrador de la entidad	Maneja los datos con que cuentan las entidades para el correcto funcionamiento de sus actividades en las distintas áreas que la componen. Gestiona cada uno de los conceptos encontrados en el negocio como: usuarios, roles, departamentos y

	servicios clínicos y no clínicos, entre otros conceptos.

2.4.3 Diagramas de casos de uso

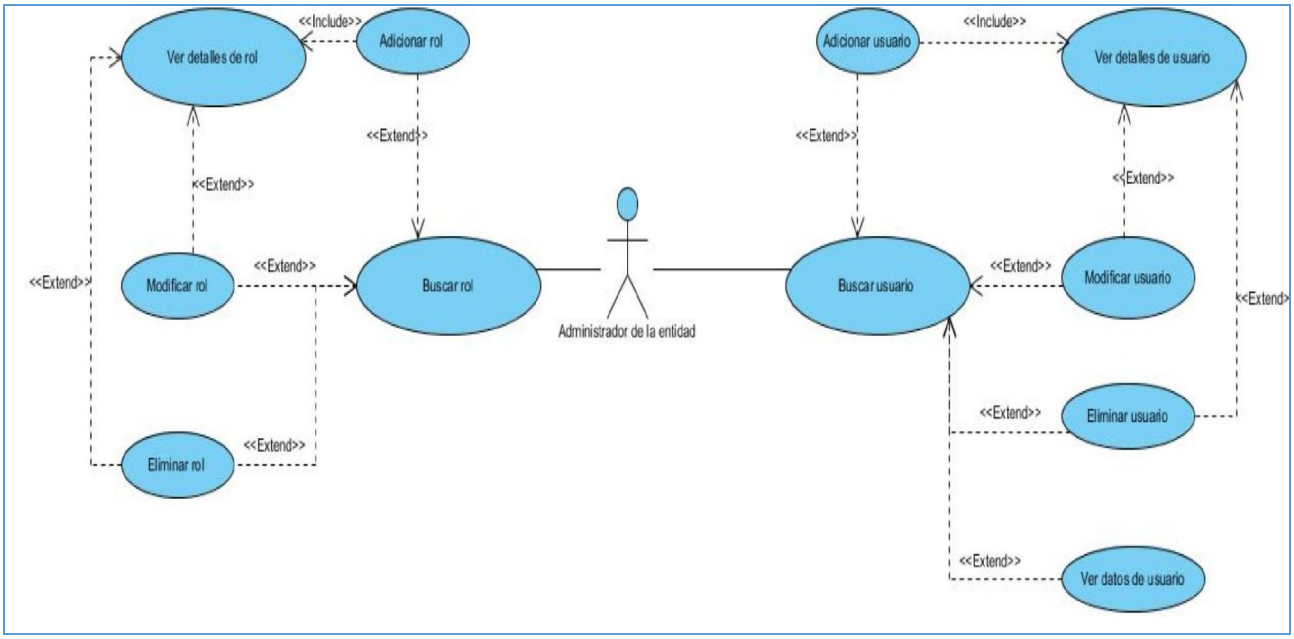


Figura 3. Diagrama de caso de uso del sistema Gestionar usuarios y roles

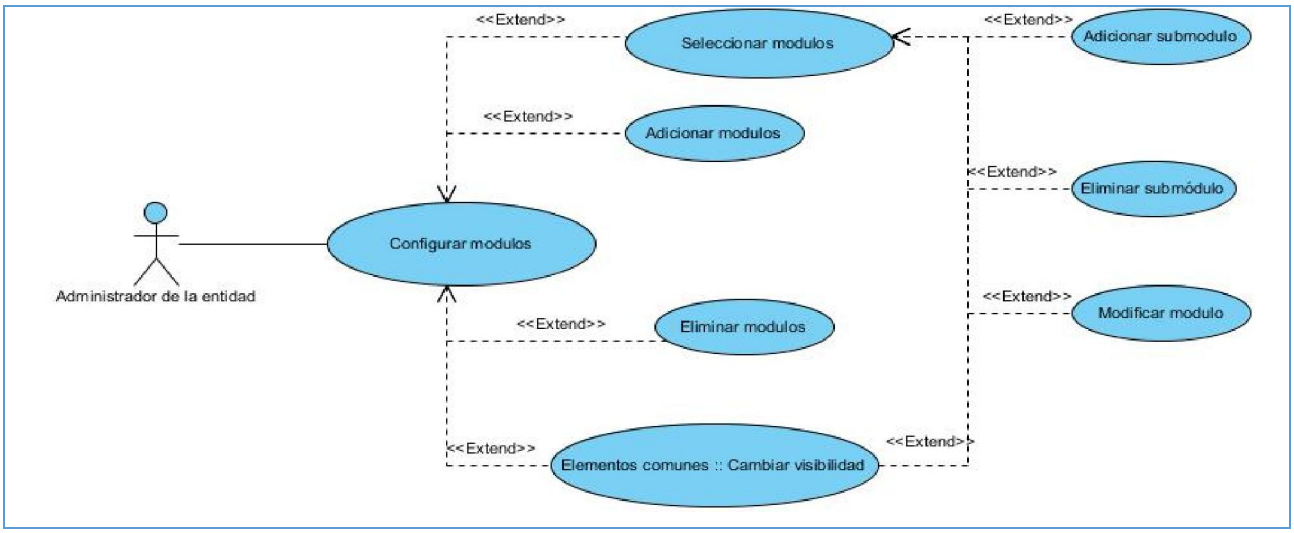


Figura 4. Diagrama de caso de uso del sistema Configurar módulos

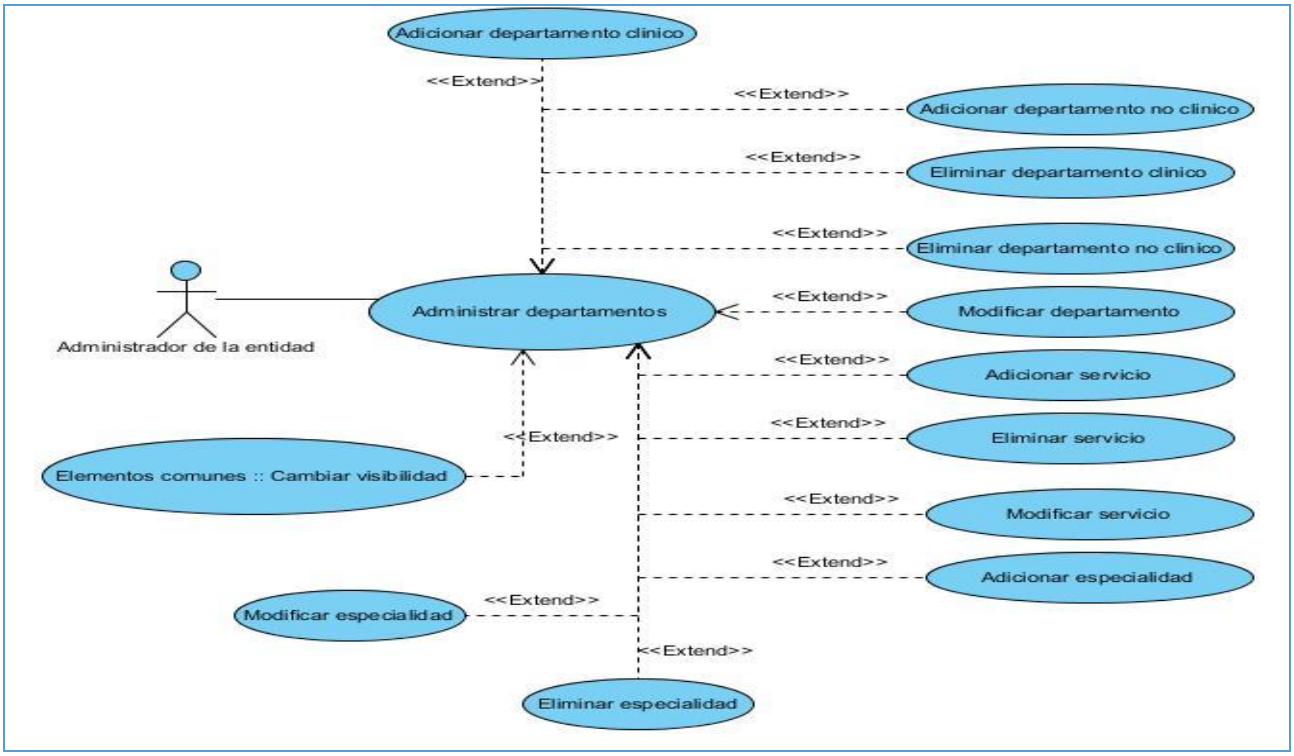


Figura 5. Diagrama de caso de uso del sistema Administrar Departamentos

2.4.4 Especificación de los casos de uso

Tabla 3. Descripción del caso de uso: Adicionar usuario

Objetivo	Permite crear usuario.
Actores	Administrador de la entidad.
Resumen	El caso de uso inicia cuando el actor accede a la opción Administrar usuarios en el menú. El sistema permite Adicionar usuario y brinda la posibilidad de introducir los datos para adicionar el nuevo usuario, el actor introduce los datos del usuario, el sistema adiciona el usuario, el caso de uso termina.
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	No existen.
Postcondiciones	Se creó un usuario.

Tabla 4. Descripción del caso de uso: Administrar departamentos.

Objetivo	Permite Administrar departamentos.
Actores	Administrador de la entidad
Resumen	El caso de uso inicia cuando el actor accede a la opción Gestionar departamentos en el menú, el sistema muestra un listado con los departamentos, servicios y

	especialidades clínicos y no clínicos de la entidad, permite seleccionarlos y salir de la vista actual, el caso de uso termina.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	No existen.
Postcondiciones	Se administró un departamento.

Tabla 5. Descripción del caso de uso: Configurar módulos.

Objetivo	Permite configurar los módulos del sistema
Actores	Administrador de la entidad
Resumen	El caso de uso inicia cuando el actor accede a la opción Configurar módulos en el menú. El sistema muestra un listado de módulos a los que el actor tiene acceso y permite adicionar, eliminar y cambiar la visibilidad de los mismos; termina el caso de uso.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	No existen
Postcondiciones	Se configuró un módulo.

Nota: Para un mayor entendimiento de los diagramas de casos de uso Ver Anexo 2: Especificación de casos de uso.

2.5 Conclusiones del capítulo

- La documentación que se obtiene permite un mejor entendimiento del proceso de negocio, a partir del cual se identifican las funcionalidades necesarias para el desarrollo del módulo propuesto.

Capítulo 3. Diseño del módulo Configuración entidad.

En el presente capítulo se describe la arquitectura y los patrones de diseño definidos para el desarrollo de las funcionalidades asociadas al módulo Configuración entidad. Se realiza el diagrama de clases de diseño y se brinda una descripción de las clases identificadas para su posterior implementación.

3.1 Descripción de la arquitectura

De acuerdo con la definición que brinda la IEEE Std 1471-2000, la Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (27)

Para el desarrollo del módulo y teniendo en cuenta la tecnología propuesta, se define como parte de la línea base de la arquitectura la implementación del patrón de diseño Modelo Vista Controlador. Este patrón es muy usado en aplicaciones web. Permite la separación de los datos de una aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos: el modelo, donde se encuentran los datos y las reglas del negocio; la vista, que muestra la información del modelo al usuario; y el controlador, que gestiona las entradas del usuario.

Con este patrón se logra realizar un diseño que separe la vista del modelo y permita la reusabilidad de los componentes. Ofrece mejor organización según la función que realizan, permitiendo que en un momento determinado un elemento de una capa pueda ser modificado o sustituido completamente causando el mínimo de alteraciones en otro elemento que lo utilice.

La capa de la vista o capa de presentación está compuesta por páginas XHTML, desarrolladas básicamente con JSF, utilizando las librerías Ajax4JSF y RichFaces, que se complementan con la plataforma de integración JBoss Seam. Además se utilizan componentes Seam de interfaz de usuario y Facelets como motor de plantillas lo que enriquece el diseño de la interfaz de usuario.

La capa de negocio está constituida por clases controladoras que se encargan de definir la lógica del negocio del módulo, así como del manejo y validación de los datos capturados en la capa de presentación. A estas clases, mediante anotaciones que provee el marco de trabajo Seam, se les puede especificar el contexto en que se encuentran, ya sea conversacional, evento, página, entre otros, los que definen el estado de los datos y las entidades que manejan.

La capa de datos o modelo se encarga principalmente de la carga, modificación, eliminación y persistencia de la información en la base de datos. Esta capa valida los datos antes de persistirlos. Todo este manejo de datos es mediante Hibernate que abstrae al desarrollador del gestor de base de datos utilizado a través del mapeo de tablas, lo que permite llevar las consultas a un lenguaje de objetos. (28)

3.2 Modelo de diseño

El modelo de diseño constituye el conjunto de diagramas que describen el diseño lógico de un sistema. Comprende los diagramas de clases de software, diagramas de interacción, diagramas de paquetes, etc., ofreciendo una perspectiva de especificación o implementación, como quiere el modelador. (25)

Por lo general, durante la elaboración del diseño se utilizan patrones que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construyen sistemas automatizados. Para la definición del diseño de la solución propuesta se tuvo en cuenta una serie de patrones, entre ellos los Patrones para la Asignación General de Responsabilidades (GRASP, por sus siglas en inglés).

El patrón bajo acoplamiento soporta el diseño de clases más independientes y reduce el impacto de las modificaciones, pues en caso de querer modificar un usuario o asociarle una nueva especialidad a un médico, solamente se llama a sus respectivas clases.

La clase UsuarioBuscarControlador_configuracionentidad tiene como objetivo seleccionar de la bases de datos todos los usuarios pertenecientes a la entidad en la que se esté trabajando, en ella se representa el patrón alta cohesión pues realiza solo funciones específicas de búsqueda de los usuarios y designa las demás operaciones a otras clases.

Por otra parte se tiene la clase UsuarioCrearControlador_configuracionentidad encargada de crear y guiar la asignación de responsabilidades relacionadas con la creación de objetos de tipo Usuario_configuracion, UserInRole, Medico_configuracion, Enfermera_configuracion, Bioanalista_configuracion y otras, de esta forma se ve reflejado el uso del patrón creador.

Además, se manifiesta el patrón experto en la clase FuncionalidadesManager_configuracionentidad, pues esta recibe de la clase ActiveModule a

través del método activeModule.getActiveModule().getEntidad() la información necesaria para conocer la entidad seleccionada y así cumplir su tarea.

3.2.1 Diagramas de clases del diseño

Los diagramas de clases de diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema y forman parte de las realizaciones de casos de usos (26). Además son de gran importancia, pues permiten visualizar, especificar y documentar modelos estructurales.

Con la definición de los principales aspectos a tener en cuenta para la realización del modelo de diseño, se establece una estructura de paquetes dividida en fragmentos manejables para su posterior implementación. Existe una relación entre los paquetes mediante los que se establecen dependencias entre las distintas clases que lo contienen. Todas las clases están agrupadas en el paquete repositorio de clases. El paquete sesiones contiene todas las clases controladoras agrupadas en paquetes, uno tiene las controladoras autogeneradas, otro las personalizaciones que se hacen sobre algunas controladoras autogeneradas y un paquete para las controladoras propias del proceso. El paquete Entidades que contiene a su vez otro paquete con las entidades autogeneradas y personalizadas. Por último todas las vistas están contenidas en el paquete Vistas. Estos paquetes se relacionan entre ellos ya que las vistas consultan y actualizan las entidades e invocan a las controladoras y estas modifican las entidades.

El paquete Configuración_entidad contiene los paquetes de realización de casos de usos (RCU) donde se agrupan los casos de uso de acuerdo a las entidades que gestionan. Estos casos de uso son implementados haciendo uso de la estructura del paquete Repositorio de clases.

A continuación se representa el diagrama de paquetes del módulo propuesto:

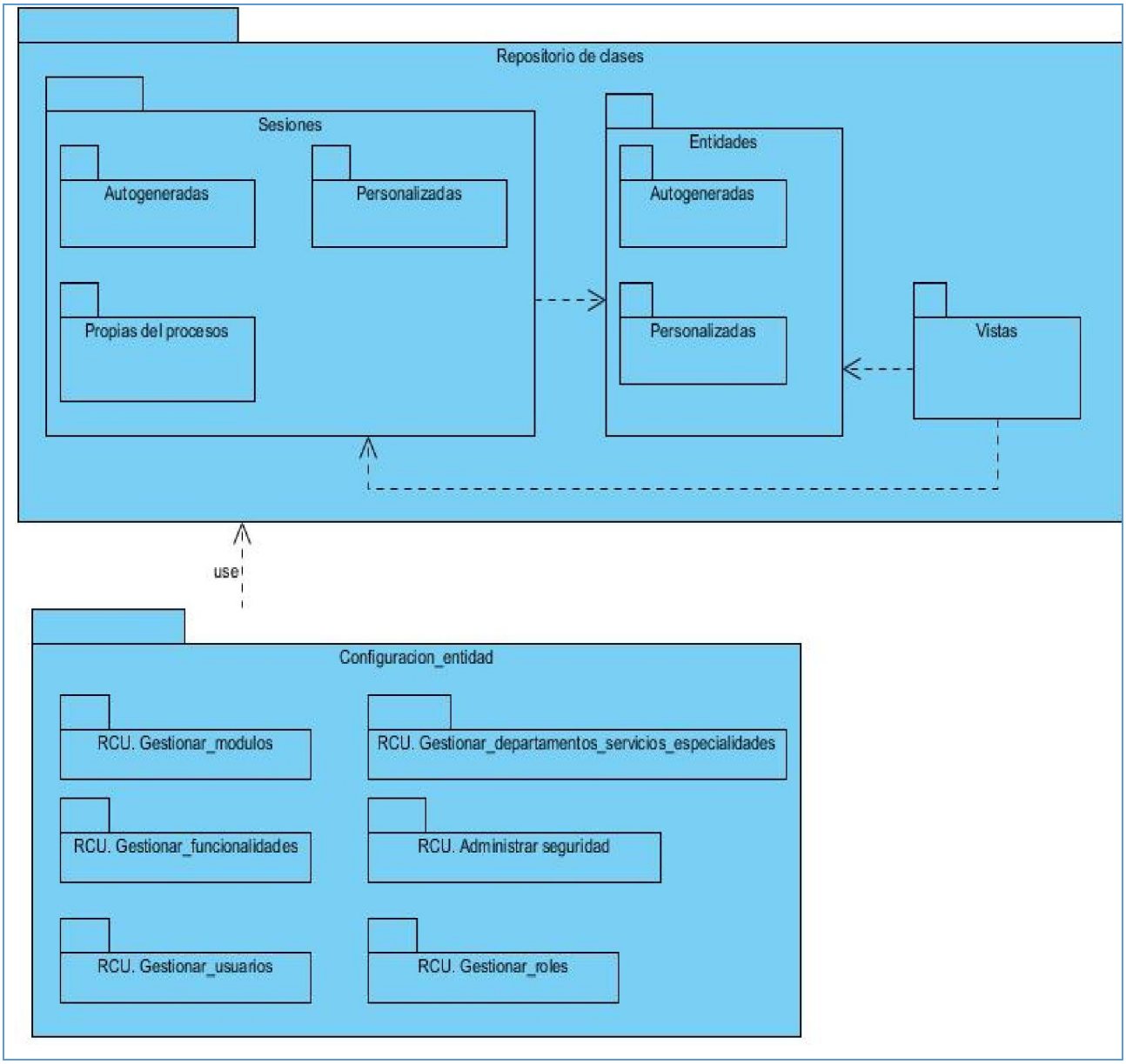


Figura 6. Diagrama de paquetes Módulo Configuración Entidad.

La estructura general de los diagramas de clases del diseño del módulo Configuración entidad están compuestos por páginas clientes que son construidas por páginas servidoras y que a su vez contienen formularios que muestran y capturan toda la información. Las páginas servidoras invocan métodos o responsabilidades en la clase controladora que según la acción solicitada pueden modificar las entidades.

A continuación se realiza la descripción de las principales clases identificadas en el diseño para su futura implementación, con el objetivo de lograr una comprensión más amplia del módulo en cuestión.

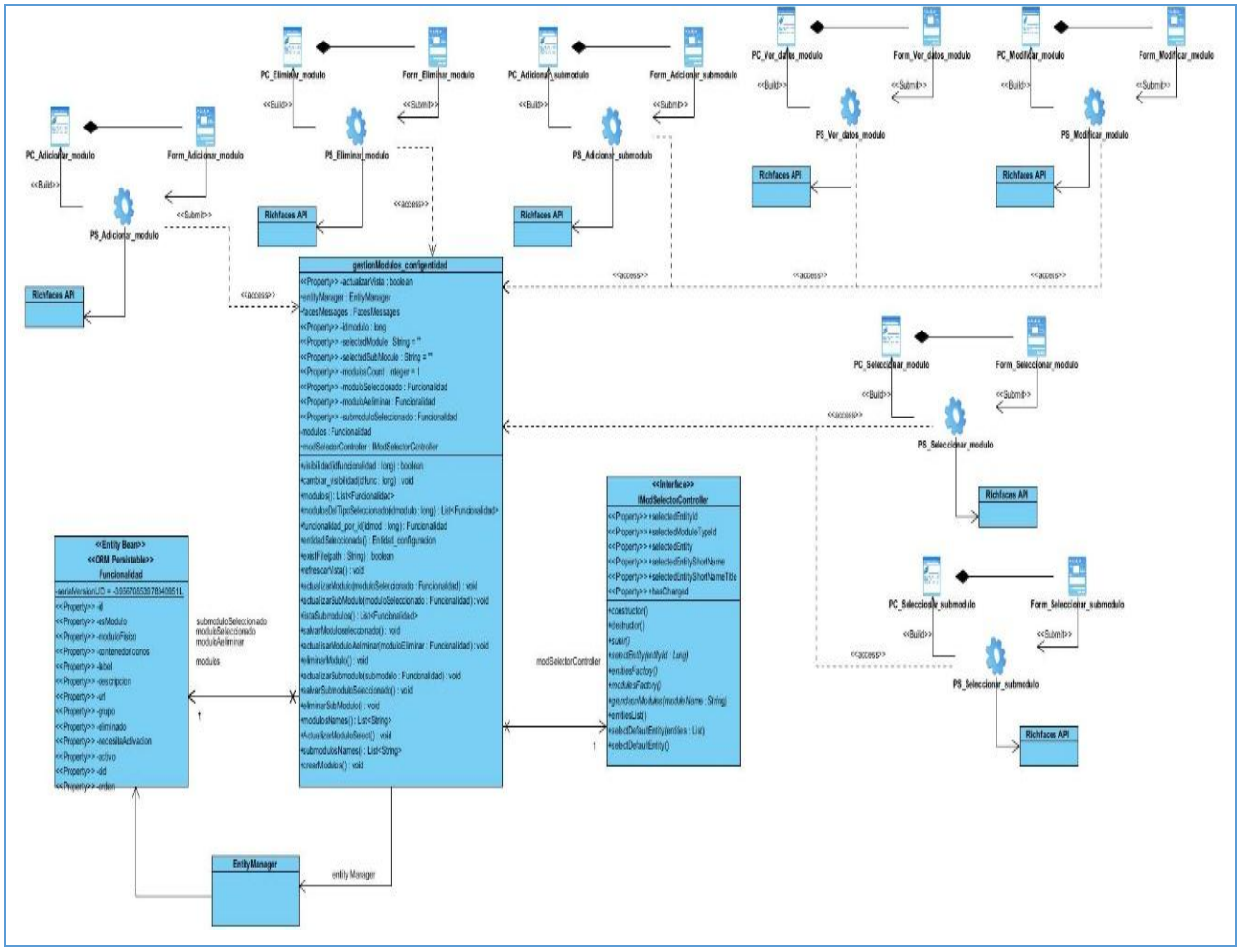


Figura 7. DCD_ Gestionar módulos.

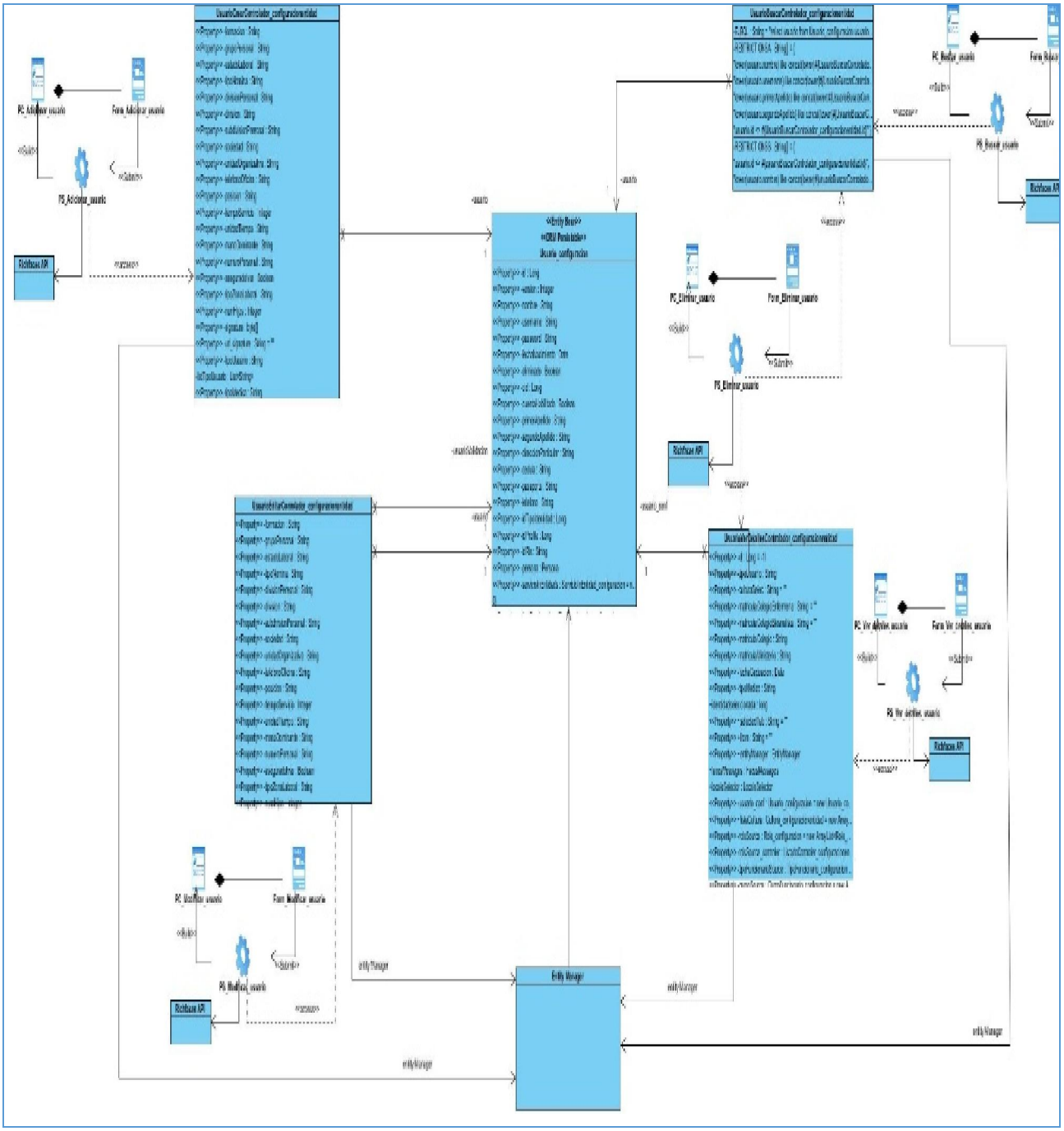


Figura 8. DCD_ Gestionar usuarios.

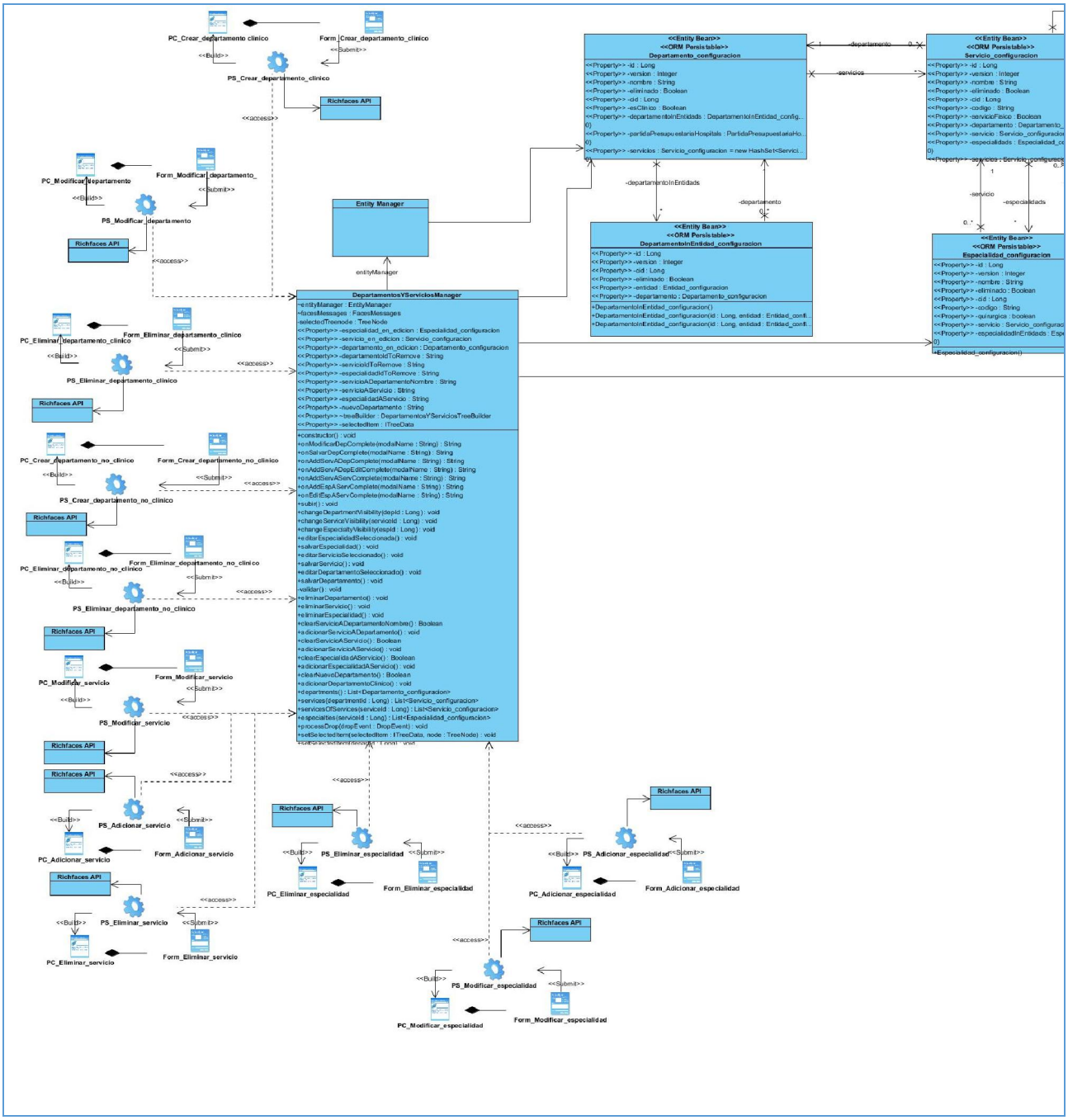


Figura 9. DCD_ Gestionar departamentos servicios especialidades.

Nota: Para un mayor entendimiento de los diagramas de clases Ver Anexo 3: Descripción de las clases.

3.3 Conclusiones del capítulo

- El diseño propuesto, a partir del uso del patrón Modelo-Vista-Controlador, permite una mejor comprensión de la estructura del módulo Configuración entidad.

Capítulo 4. Implementación del módulo Configuración entidad

Luego de realizado el análisis y el diseño de la solución propuesta, se tienen todas las condiciones para comenzar la implementación de la misma. Este capítulo muestra los principales componentes y sus relaciones, haciendo uso del Diagrama de Componentes. Además se elabora el Modelo de Despliegue y se expone la estrategia de seguridad aplicada.

4.1 Modelo de datos:

Un modelo de datos es un conjunto de conceptos utilizados para organizar los datos de interés y describir su estructura en forma comprensible para un sistema informático (25). En general un modelo de datos es la estructura o representación física de las tablas de la base de datos. En el modelo de datos del módulo Configuración entidad (Ver Anexo 4. Modelo de datos) además de realizar algunas modificaciones a las tablas del módulo Configuración, se representan dos nuevas tablas, rol_in_entidad y especialidad_rotación, en los esquemas seguridad y común respectivamente. Estas tablas se agregaron con el objetivo de almacenar la información referente a las especialidades por las que pueden rotar los médicos interno o residentes de una institución hospitalaria y la relación de roles pertenecientes a cada una de las entidades del sistema, ya que con la estructura que presenta la bases de datos existente, no permite guardar la información antes mencionada.

4.1.1 Descripción de las tablas

Tabla 6. Descripción de las tablas.

Nombre	Descripción
Funcionalidad	Contiene todos los datos de las funcionalidades del sistema.
Usuario	Almacena todos los usuarios que interactuarán con el sistema.

Campos comunes en todas las tablas.

Tabla 7. Descripción de los atributos comunes entre todas las tablas.

Atributo	Tipo	Descripción
Id	bigint	Id necesario en cada entidad para las referencias en las relaciones entre tablas.

version	integer	Indica con qué versión de la entidad se está trabajando. Es usado para garantizar que se está trabajando con la versión de la entidad más actualizada que existe en la base de datos.
eliminado	boolean	Permite la eliminación lógica con que cuenta el sistema, cuando está en verdadero indica que la entidad está eliminada.
cid	Long	Campo para tracking de los cambios en la bitácora.

Tabla 8. Descripción de la tabla Funcionalidad.

Atributos	Tipo	Descripción
Label	varchar	Nombre de la funcionalidad
url	varchar	Dirección de la página que es invocada por la funcionalidad.
Orden	integer	Orden en el que se encuentra la funcionalidad.
Imagen	varchar	Imagen del icono de la funcionalidad.
Es_modulo	boolean	Si es módulo o no.
Id_funcionalidad_padre	integer	Identificador de la funcionalidad padre.
Label_size	integer	Cantidad de caracteres que puede tener el nombre.
nombre	varchar	Nombre de la funcionalidad.
Modulo_fisico	boolean	Módulo contenedor de las funcionalidades.
Contenedor_iconos	boolean	Si contiene iconos o no.
codebase	varchar	Dirección física de la funcionalidad.
descripcion	varchar	Descripción del módulo.
Grupo	varchar	Grupo al que pertenece.
Necesita_activacion	boolean	Si necesita ser activado o no.
activo	boolean	Si está activo o no.

Tabla 9. Descripción de la tabla Usuario.

Atributos	Tipo	Descripción
Nombre	varchar	Nombre del usuario.
Username	varchar	Nombre de usuario para acceder al sistema.
Password	varchar	Contraseña del usuario para acceder al sistema.
Fecha_nacimiento	date	Fecha de nacimiento del usuario.
Cuenta_habilitada	boolean	Establece si el usuario estaba habilitado en el sistema.
Primer_apellido	varchar	Primer apellido del usuario.
Segundo_apellido	varchar	Segundo apellido del usuario.
Dirección_particular	varchar	Dirección particular del usuario.
Cedula	varchar	Cédula del usuario.
Pasaporte	varchar	Pasaporte del usuario.
Teléfono	varchar	Teléfono del usuario.
Id_profile	bigint	Identificador del perfil del usuario

4.2 Modelo de Despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (29)

Para la implantación y utilización del módulo Configuración entidad, el usuario debe conectarse al sistema mediante una computadora cliente a través del navegador. Las peticiones por el protocolo HTTPS (Hypertext Transfer Protocol Secure) serán procesadas por el servidor de aplicaciones, y este a su vez enviará las respuestas al cliente. El servidor de aplicaciones emitirá peticiones por el protocolo TCP / IP (Transfer Control Protocol / Internet Protocol) hacia el servidor de base de datos. El diagrama de despliegue del módulo quedó definido de la siguiente forma:

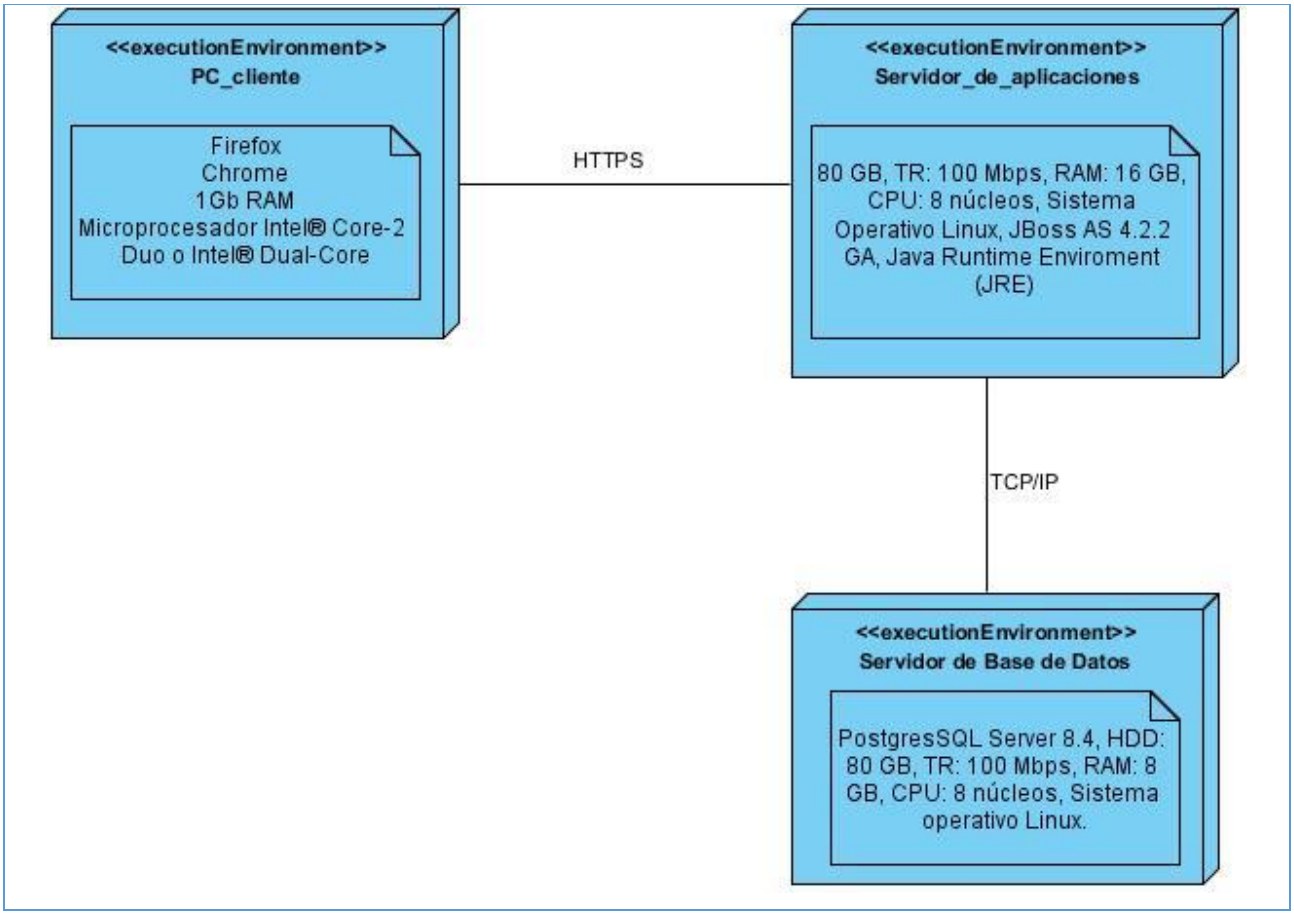


Figura 10. Modelo de despliegue

4.3 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema agrupados en paquetes lógicos y las relaciones entre ellos. Además permiten visualizar con más facilidad la estructura general del sistema mostrando los componentes del software. (25)

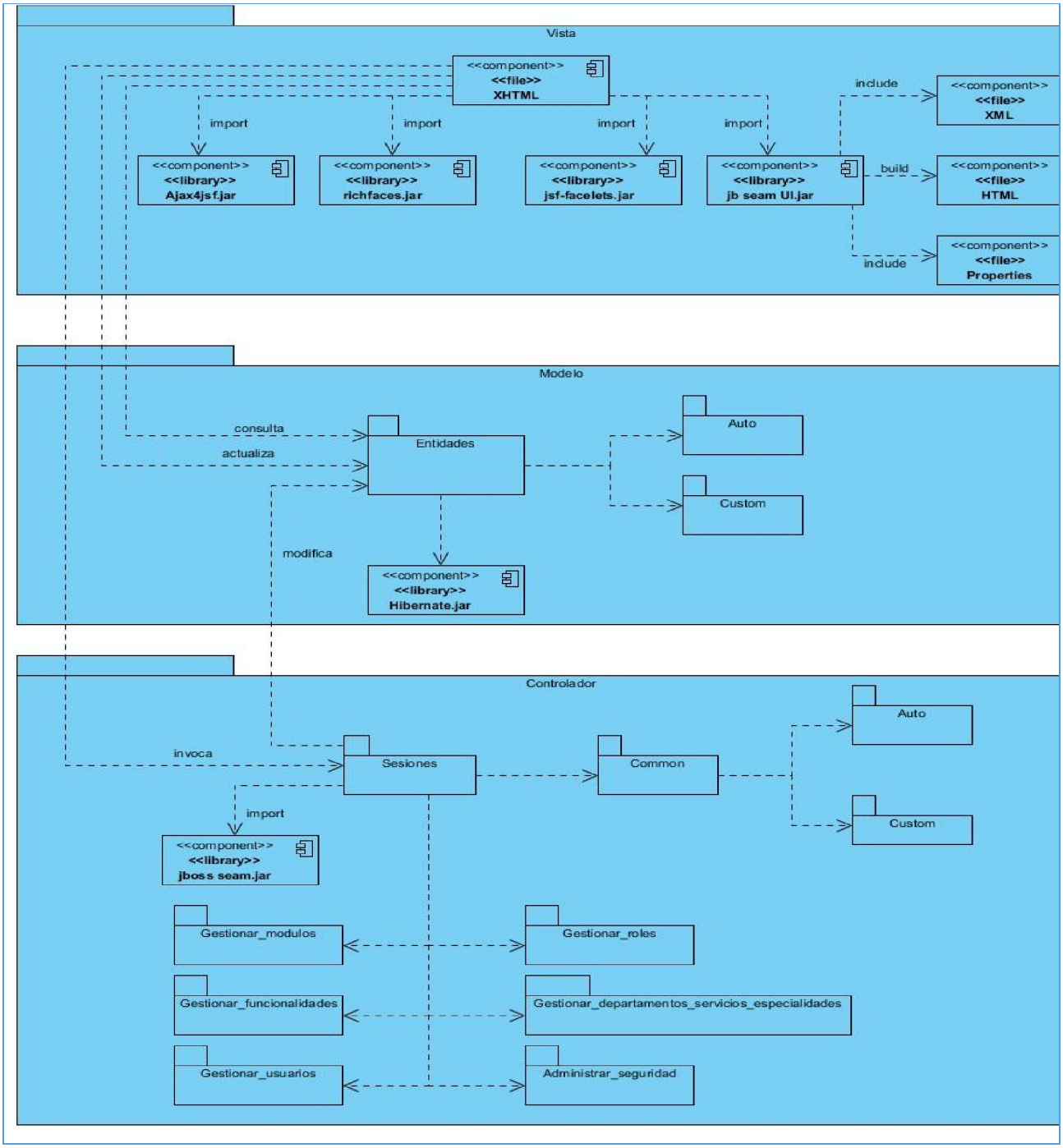


Figura 11. Diagrama de componentes.

4.4 Seguridad

La seguridad es de gran importancia para cualquier Sistema de Información y más si se trata de la salud de las personas. Para lograr un sistema seguro deben cumplir los aspectos de integridad, confidencialidad y disponibilidad de la información. La unión de estas características trae consigo que la información sea modificada y visualizada por el personal autorizado; de forma controlable y que esté disponible cuando se necesite.

Para que se garantice la seguridad, el módulo Configuración entidad solo es accedido por los administradores de la institución. El mismo contempla la gestión del personal asociado a la entidad, así como la administración de los permisos de roles y usuarios sobre directorios, páginas, y opciones del menú, asegurando así que solo tengan acceso a la información correspondiente según el papel que desempeñan en la entidad. Todas las actividades realizadas por los usuarios son almacenadas en la Bitácora de sucesos, posibilitando que los administradores puedan ver las trazas de acciones a la hora de realizar auditorías al sistema.

4.5 Tratamiento de errores

Las excepciones son el mecanismo recomendado para tratar los errores que se produzcan durante la ejecución de las aplicaciones. Cuando ocurre un error dentro de un método Java, automáticamente se crea un objeto *Exception* el cual es tratado en el sistema. Este objeto contiene información sobre la excepción, incluyendo su tipo y el estado del programa. (30)

En el HIS se propone el tratamiento de excepciones principalmente en las regiones críticas de código, es decir, donde los datos son insertados o modificados en la base de datos, así como en el proceso de validación. El control de la navegación, en caso de ocurrir una excepción que implique una redirección, se maneja mediante los *.pages.xml*, estos se encargan de capturar globalmente las excepciones y ejecutar las instrucciones determinadas. Para controlar el resto de las excepciones se utiliza el componente *FacesMessages* del *framework Seam.* Este se encarga de mostrar los mensajes que se manejan a través del objeto f*acesMessages* inyectado en las clases controladoras tratando los mensajes por tipo (error, alerta y notificación).

4.6 Conclusiones del capítulo

- Los artefactos correspondientes a modelo de datos y diagrama de componentes garantizan la descripción de los datos, los componentes y las dependencias entre estos en el módulo desarrollado.
- Con la implementación de las funcionalidades definidas se obtuvo el módulo Configuración entidad para el Sistema de Información Hospitalaria del CESIM.

Conclusiones

Con el desarrollo del módulo Configuración entidad del Sistema de Información Hospitalaria del CESIM se arriba a las siguientes conclusiones:

- Con la separación de las funcionalidades del módulo Configuración asociadas a la administración general del sistema y las específicas de una entidad se mejora el entendimiento, seguridad y usabilidad de la configuración del Sistema de Información Hospitalaria del CESIM.
- Con el uso de la tecnología definida se obtiene la documentación correspondiente al expediente de desarrollo del módulo Configuración entidad.
- La implementación del módulo Configuración entidad permite la correcta gestión y administración de las entidades del sistema.

Recomendaciones

Para lograr la continuidad de este trabajo, debido a su importancia, se recomienda a partir de la investigación realizada:

- Incorporar las funcionalidades asociadas a la Gestión de Ubicaciones y Gestión de Camas para fortalecer la administración de la entidad del sistema.

Referencias Bibliográficas

1. **Salud, Departamento de Informática en.** Hospital Italiano de Buenos Aires. [En línea] 2015. [Citado el: 20 de enero de 2015.] http://www.hospitalitaliano.org.ar/infomed/index.php?contenido=ver_curso.php&id_curso=18084.

2. **Martínez, Juan F. Guerrero.** *Tema 11 Informática Médica y Telemedicina.* Valencia : s.n., 2011.

3. **Centro Financiero Invercasa.** Digitech. [En línea] 2012. [Citado el: 5 de febrero de 2015.] http://www.digitech.net.ni/sistema-integral.

4. **Microsoft.** Microsoft Pinpoint. [En línea] 2015. [Citado el: 15 de marzo de 2015.] https://pinpoint.microsoft.com/es-mx/Applications/4295029536?id=searchResult..

5. **sivsa.** sivsa. [En línea] [Citado el: 14 de marzo de 2015.] http://www.sivsa.com/site/es/productos-y-servicios/soluciones/seguridad. ISO 9001:2008 e ISO 14.001:2004.

6. **Cabrera Hernández, Mirna, y otros.** *PLATAFORMA PARA LA ADMINISTRACIÓN, PROCESAMIENTO Y TRANSMISIÓN DE LA INFORMACIÓN EN EL SISTEMA DE SALUD: SISALUD.* La Habana : s.n.,2010.

7. *Phphclinica. Sistema de informacion hospitalaria.* **León Hernández, Lic. Pedro Luis y Santos, Lic. Adelys .** #147087, La Habana : s.n., 2012. ISSN 1886-8924.

8. **Latina, Osmosis.** Osmosis Latina. [En línea] 2011. [Citado el: 23 de febrero de 2015.] http://www.osmosislatina.com/java/basico.htm.

9. **Allen, Dan.** Seam in action. [En línea] Manning Publications Co, septiembre de 2008. [Citado el: 8 de marzo de 2015.] http://www.manning.com/dallen/. ISBN: 1933988401.

10. **Orshalick, Jacob.** DZone. [En línea] 2014. [Citado el: 26 de marzo de 2015.] http://refcardz.dzone.com/refcardz/seam-ui.

11. *Componente web para el análisis de información clínica usando la técnica de Minería de Datos por agrupamiento.* **Ochoa Reyes, Ing. Alexeis Joel , y otros.** no.1, La Habana : s.n., 2013, Vol. vol.6. ISSN 1684-1859.

12. **JBoss.** Jboss Aplication Server 7. [En línea] [Citado el: 20 de marzo de 2015.] http://jbossas.jboss.org/.

13. **Jhonson, Javid Jamae Peter.** *Jboss in action.* Greenwich : s.n., 2009. ISBN 978-1-933988-02-3.

14. **Faces, Java Server.** Java Server Faces.org. [En línea] 2010. [Citado el: 10 de marzo de 2015.] http://www.javaserverfaces.org/.

15. **Alvarez Lorenzo, Ing.Amaya, y otros.** *SLD237 DESARROLLO DE LA ESPECIALIDAD PSICOLOGÍA DEL MÓDULO CONSULTA EXTERNA DEL SISTEMA ALAS-HIS.* La Habana : s.n., 2013.

16. **Andalucía, Junta de.** Marco de Desarrollo de la Junta de Andalucía. [En línea] 2013. [Citado el: 16 de marzo de 2015.] http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/96.

17. **Java.** Java.net The Source for Java Technology Collaboration. [En línea] [Citado el: 15 de febrero de 2015.] https://www.java.net/.

18. **W3C.** W3C. [En línea] 1 de agosto de 2002. [Citado el: 4 de marzo de 2015.] http://www.w3.org/TR/xhtml1.

19. **Jacobson, Ivar, Booch, Grady y Rumbaugh, Jame.** *El proceso de desarrollo de software.* 2013.

20. **OMG.** Unified Modeling Language™ (UML®) Resource Page. [En línea] 22 de mayo de 2015. [Citado el: 30 de mayo de 2015.] http://www.uml.org.

21. **Middleware, Red Hat JBoss.** Red Hat JBoss Developer Studio. [En línea] 2012. [Citado el: 20 de abril de 2015.] http://www.redhat.com/es/technologies/jboss-middleware/developer-studio.

22. **Martínez, Rafael.** Postgres-es. [En línea] 02 de octubre de 2010. [Citado el: 25 de abril de 2015.] http://www.postgresql.org.es/sobre_postgresql.

23. **PgAdmin.** Guía de Ubuntu. [En línea] 2010. [Citado el: 28 de febrero de 2015.] http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.

24. **Larman, Craig.** *UML y patrones : introducción al análisis y diseño orientado a objetos.* Madrid : Prentice-Hall, 2006.

25. **Somerville, Ian.** *Ingeniería del Software Parte II.* Madrid : s.n., 2005.

26. **Pressman, Roger S.** *Ingeniería del software: un enfoque práctico.* 2002.

27. **Reynoso, Carlos Billy.** *Introducción a la Arquitectura de Software.* Buenos Aires : s.n., 2004.

28. *Patrón Modelo Vista Controlador.* **Fernández Romero, Yenisleidy y Díaz González, Yanette.** 1, La Habana : s.n., 2012, Vol. 11. ISSN 1729.

29. **Despliegue, Modelo.** Sparx systems. [En línea] 2010. [Citado el: 25 de abril de 2015.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

30. **Junta de, Andalucía.** Marco de dearollo de la Junta de Andalucía. [En línea] 2010. [Citado el: 12 de mayo de 2015.] http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/214.

Bibliografía

- **Allen Dan** Seam in action [En línea]. - Manning Publications Co, septiembre de 2008. - 8 de marzo de 2015. - http://www.manning.com/dallen/. - ISBN: 1933988401.
- **Alvarez Lorenzo Ing.Amaya [y otros]** SLD237 DESARROLLO DE LA ESPECIALIDAD PSICOLOGÍA DEL MÓDULO CONSULTA EXTERNA DEL SISTEMA ALAS-HIS [Informe]. - La Habana : [s.n.], 2013.
- **Andalucía Junta de** Marco de Desarrollo de la Junta de Andalucía [En línea]. - 2013. - 16 de marzo de 2015. - http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/96.
- **Borroto Carmona Dr. Gerardo** Reflexiones sobre la elaboración de una tesis. - La Habana : CUJAE, 2012.
- **Cabrera Hernández Mirna [y otros]** PLATAFORMA PARA LA ADMINISTRACIÓN, PROCESAMIENTO Y TRANSMISIÓN DE LA INFORMACIÓN EN EL SISTEMA DE SALUD: SISALUD [Informe]. - La Habana : [s.n.], 2010.
- **Centro Financiero Invercasa** Digitech [En línea]. - 2012. - 5 de febrero de 2015. - http://www.digitech.net.ni/sistema-integral.
- **Clara Hernández Anisbel y Ise Morales Delís** Integración de herramientas de pruebas de seguridad para aplicaciones web en XILEMA-PlatSI. [Informe]. - La Habana : Universidad de las Ciencias Informáticas, 2014.
- **Despliegue Modelo** Sparx systems [En línea]. - 2010. - 25 de abril de 2015. - http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
- **Faces Java Server** Java Server Faces.org [En línea]. - 2010. - 10 de marzo de 2015. - http://www.javaserverfaces.org/.
- **Fernández Romero Yenisleidy y Díaz González Yanette** Patrón Modelo Vista Controlador [Publicación periódica]. - La Habana : [s.n.], 2012. - 1 : Vol. 11. - ISSN 1729.
- **Grosso Andrés** Pácticas de software [En línea]. - 21 de marzo de 2011. - 14 de abril de 2015. - http:// Patrones GRASP _ Prácticas de Software.htm..
- **Hernández León Rolando Alfredo y Coello González Sayda** EL PROCESO DE INVESTIGACIÓN CIENTÍFICA [Libro]. - Ciudad de la Habana : Universitaria, 2011. - ISBN 978-959-16-1307-3.
- **Jacobson Ivar, Booch Grady y Rumbaugh Jame** El proceso de desarrollo de software [Libro]. - 2013.
- **Java** Java.net The Source for Java Technology Collaboration [En línea]. - 15 de febrero de 2015. - https://www.java.net/.
- **JBoss** Jboss Aplication Server 7 [En línea]. - 20 de marzo de 2015. - http://jbossas.jboss.org/.

Bibliografía

- **Jhonson Javid Jamae Peter** Jboss in action [Informe]. - Greenwich : [s.n.], 2009. - ISBN 978-1-933988-02-3.
- **Junta de Andalucía** Marco de dearollo de la Junta de Andalucía [En línea]. - 2010. - 12 de mayo de 2015. - http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/214.
- **Larman Craig** UML y patrones : introducción al análisis y diseño orientado a objetos [Libro]. - Madrid : Prentice-Hall, 2006.
- **Latina Osmosis** Osmosis Latina [En línea]. - 2011. - 23 de febrero de 2015. - http://www.osmosislatina.com/java/basico.htm.
- **León Hernández Lic. Pedro Luis y Santos Lic. Adelys** Phphclinica. Sistema de informacion hospitalaria [Publicación periódica]. - La Habana : [s.n.], 2012. - #147087. - ISSN 1886-8924.
- **Martínez Juan F. Guerrero** Tema 11 Informática Médica y Telemedicina [Informe]. - Valencia : [s.n.], 2011.
- **Martínez Rafael** Postgres-es [En línea]. - 02 de octubre de 2010. - 25 de abril de 2015. - http://www.postgresql.org.es/sobre_postgresql.
- **Microsoft** Microsoft Pinpoint [En línea]. - 2015. - 15 de marzo de 2015. - https://pinpoint.microsoft.com/es-mx/Applications/4295029536?id=searchResult..
- **Middleware Red Hat JBoss** Red Hat JBoss Developer Studio [En línea]. - 2012. - 20 de abril de 2015. - http://www.redhat.com/es/technologies/jboss-middleware/developer-studio.
- **Ochoa Reyes Ing. Alexeis Joel [y otros]** Componente web para el análisis de información clínica usando la técnica de Minería de Datos por agrupamiento [Publicación periódica]. - La Habana : [s.n.], 2013. - no.1 : Vol. vol.6. - ISSN 1684-1859.
- **OMG** Unified Modeling Language™ (UML®) Resource Page [En línea]. - 22 de mayo de 2015. - 30 de mayo de 2015. - http://www.uml.org.
- **Orshalick Jacob** DZone [En línea]. - 2014. - 26 de marzo de 2015. - http://refcardz.dzone.com/refcardz/seam-ui.
- **PgAdmin** Guía de Ubuntu [En línea]. - 2010. - 28 de febrero de 2015. - http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.
- **Pressman Roger S.** Ingeniería del software: un enfoque práctico [Libro]. - 2002.
- **Reynoso Carlos Billy** Introducción a la Arquitectura de Software [Libro]. - Buenos Aires : [s.n.], 2004.
- **Salud Departamento de Informática en** Hospital Italiano de Buenos Aires [En línea]. - 2015. - 20 de enero de 2015. - http://www.hospitalitaliano.org.ar/infomed/index.php?contenido=ver_curso.php&id_curso=18084.

Bibliografía

- **sivsa** sivsa [En línea]. - 14 de marzo de 2015. - http://www.sivsa.com/site/es/productos-y-servicios/soluciones/seguridad. - ISO 9001:2008 e ISO 14.001:2004.
- **Somerville Ian** Ingeniería del Software Parte II [Libro]. - Madrid : [s.n.], 2005.
- **Trellini Lic. Ariel** Desarrollo de Aplicaciones Empresariales. [Informe]. - 2014.
- **W3C** W3C [En línea]. - 1 de agosto de 2002. - 4 de marzo de 2015. - http://www.w3.org/TR/xhtml1.

Anexos

Anexo 1. Diagramas de Casos de Uso

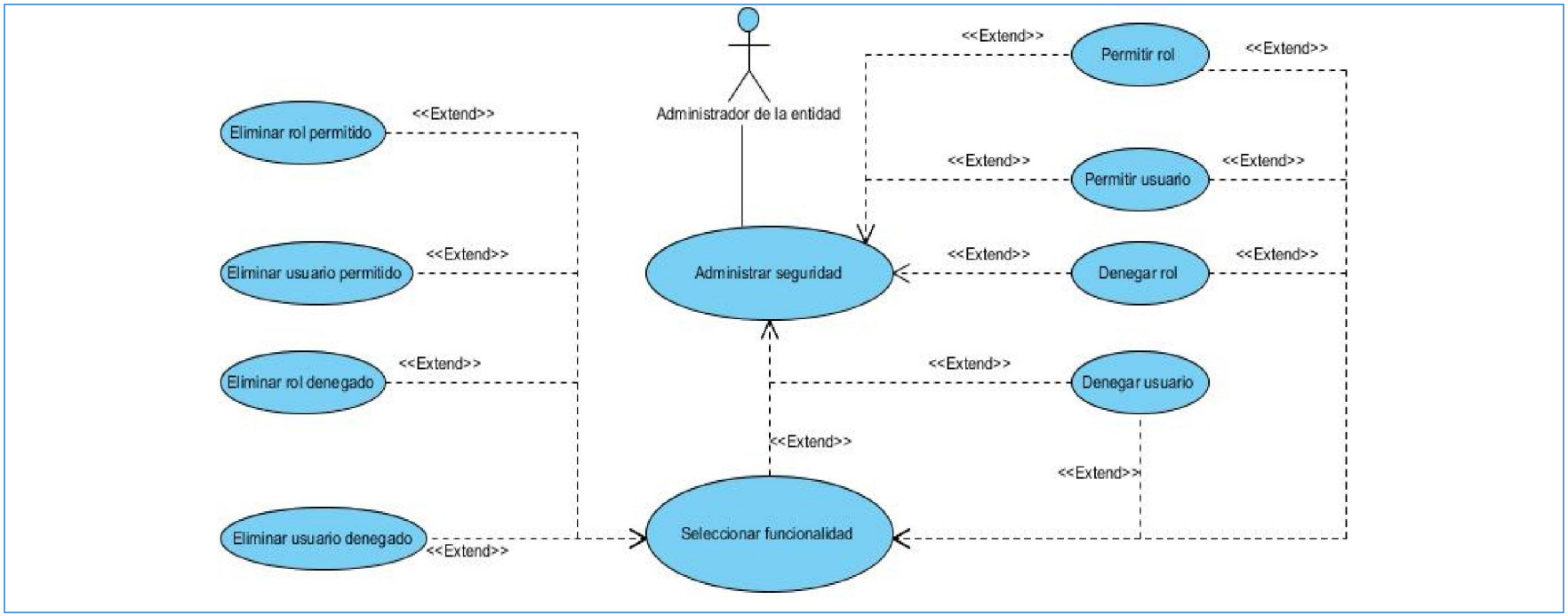


Figura 12. Diagrama de caso de uso del sistema Administrar seguridad.

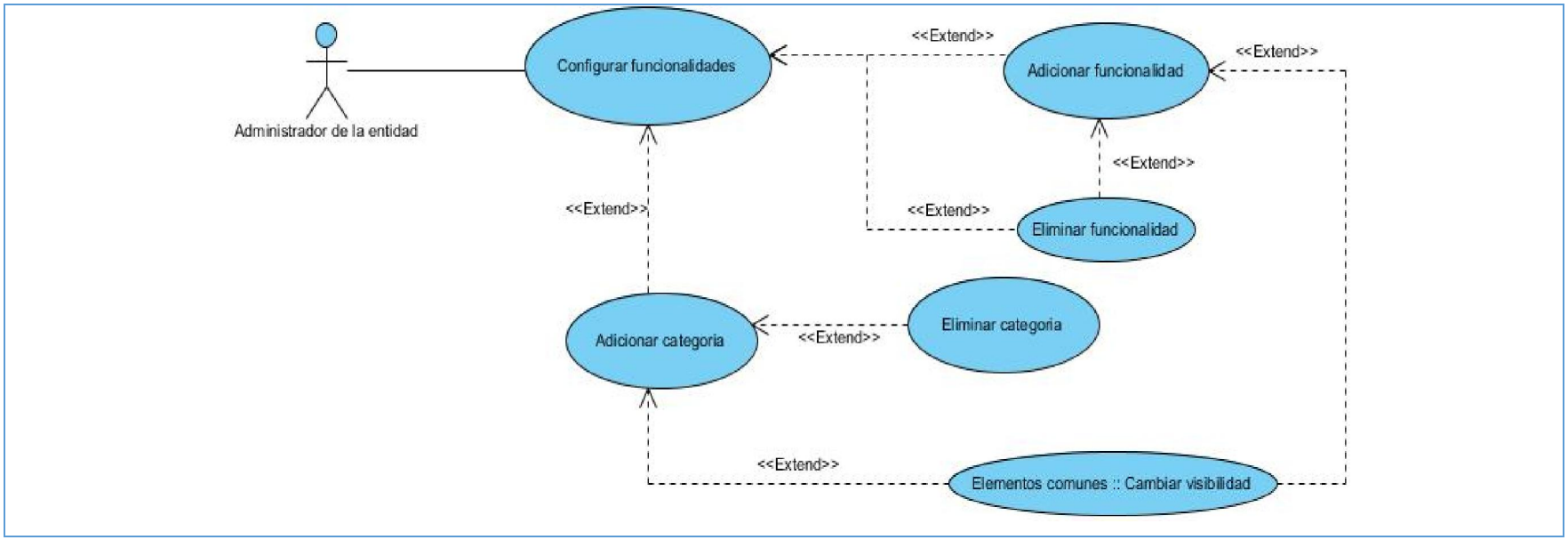


Figura 13. Diagrama de caso de uso del sistema Configurar funcionalidades.

Anexo 2. Especificaciones de casos de uso

Tabla 10. Descripción del caso de uso: Configurar módulo

Objetivo	Permite configurar los módulos del sistema
Actores	Administrador de la entidad
Resumen	El caso de uso inicia cuando el actor accede a la opción Configurar módulos en el menú. El sistema muestra un listado de módulos a los que el actor tiene acceso y permite adicionar, eliminar y cambiar la visibilidad de los mismos; termina el caso de uso.
Complejidad	Alta
Prioridad	Crítico

Precondiciones	No existen
Postcondiciones	Se configuró un módulo.

Flujo de eventos		
Flujo básico Configurar módulo		
	Actor	**Sistema**
1.	El caso de uso inicia cuando se accede a la opción Configurar módulos en el menú.	
2.		Muestra un listado de módulos y permite: • Salir • Seleccionar el módulo: Ver **Evento 1:** Seleccionar módulo • Adicionar módulo. Ver **Evento 2:** Adicionar módulo • Eliminar módulo: Ver **Evento 3:** Eliminar módulo • Cambiar visibilidad. Ver **Evento 4:** Cambiar visibilidad.
3.	Selecciona la opción Salir.	
4.		Regresa a la vista anterior.
5.		Termina el caso de uso
Flujos alternos		
Evento 1 Seleccionar módulo		
	Actor	**Sistema**
1.	Selecciona la opción Seleccionar módulo.	
2.		Muestra los datos del módulo seleccionado: Ver CU

		Seleccionar módulo.
Evento 2 Adicionar módulo		
	Actor	**Sistema**
1.	Selecciona la opción adicionar módulo.	
2.		Permite adicionar módulo: ver caso de uso **Adicionar módulo.**
Evento 3 Eliminar módulo		
	Actor	**Sistema**
1.	Selecciona la opción eliminar módulo.	
2.		Permite eliminar módulo: ver caso de uso **Eliminar módulo.**
Evento 4 Cambiar Visibilidad		
	Actor	**Sistema**
1.	Selecciona la opción cambiar visibilidad.	
2.		Permite cambiar la visibilidad: ver caso de uso **Elementos comunes:: Cambiar visibilidad.**

Relaciones	**CU incluidos**	No existen
	CU extendidos	Elementos comunes: Ver caso de uso Cambiar visibilidad. Elementos comunes:: Cambiar visibilidad. Seleccionar módulo: Ver caso de uso Seleccionar módulo. Adicionar módulo: Ver caso de uso Adicionar módulo. Eliminar módulo: Ver caso de uso Eliminar módulo.
Requisitos no funcionales		
Asuntos pendientes	No existen	

Tabla 11. Descripción del caso de uso: Adicionar usuario

Objetivo	Permite crear usuario.
Actores	Administrador de la entidad.
Resumen	El caso de uso inicia cuando el actor accede a la opción Administrar usuarios en el menú. El sistema permite Adicionar usuario y brinda la posibilidad de introducir los datos para adicionar el nuevo usuario, el actor introduce los datos del usuario, el sistema adiciona el usuario, el caso de uso termina.
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	No existen.
Postcondiciones	Se creó un usuario.
Flujo de eventos	
Flujo básico Adicionar usuario	

	Actor	**Sistema**
1.	El caso de uso inicia cuando se accede a la opción Adicionar usuario.	
2.		Brinda la posibilidad de introducir los datos: Pestaña Generales: • Usuario. • Contraseña. • Repetir contraseña. • Nombre. • Primer apellido. • Segundo apellido.

		• Cédula. • Pasaporte. • Teléfono. • Dirección particular. Seleccionar: • Idioma. • Cuenta habilitada. • Foto. • Fecha de nacimiento. • Tipo de usuario • Pestaña Roles (Ver **Sección 1**: Roles). • Pestaña Funcionarios (Ver **Sección 2**: Funcionarios). • Pestaña Servicios (Ver **Sección 3**: Servicios). • Pestaña Firma. • Seleccionar: • Firma. y permite: • Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** “Cancelar operación.”
3.	Introduce los datos de usuario: • Usuario. • Contraseña.	

	• Repetir contraseña. • Nombre. • Primer apellido. • Cédula. • Opcionalmente introduce los datos: • Segundo apellido. • Pasaporte. • Teléfono. • Dirección particular. Selecciona: • Idioma. • Cuenta habilitada. • Foto. • Fecha de nacimiento. • Pestaña Roles (Ver **Sección 1**: Roles). • Pestaña Funcionarios (Ver **Sección 2**: Funcionarios). • Pestaña Servicios (Ver **Sección 3**: Servicios). • Tipo de usuario	
4.		Si tipo de usuario: Médico (Ver **Sección 4**: Datos como médico), y Especialidades (Ver **Sección 5**: Especialidades de graduación). Enfermero (Ver **Sección 6**: Datos de enfermero) Bioanalista (Ver **Sección 7**: Datos de bioanalista)

5.	Selecciona la opción de Aceptar adicionar usuario.	
6.		Valida los datos. Si hay datos incompletos, ver **Evento 2:** "Existen datos incompletos". Si hay datos incorrectos, ver **Evento 3:** "Existen datos incorrectos."
7.		Adiciona usuario.
8.		Muestra los datos del usuario. Ver caso de uso: Ver detalles de usuario.
9.		Termina el caso de uso.
Secciones		
Sección 1: Roles		
	Actor	**Sistema**
1.	Accede a la pestaña Roles.	
2.		Muestra un listado con los roles existentes en el sistema y brinda la posibilidad de seleccionar: • Roles Y permite: • Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** "Cancelar operación."
3.	Selecciona el o los datos de: • Roles.	
4.		Regresa al paso 5 del **Flujo Normal de Eventos.**
Sección 2: Funcionarios		

	Actor	Sistema
1.	Accede a la pestaña Funcionarios.	
2.		Muestra un listado con los funcionarios y un listado con los cargos existentes en el sistema y brinda la posibilidad de seleccionar: • Tipos de funcionarios. • Cargos. Y permite: • Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** "Cancelar operación."
3.	Selecciona el o los datos de: • Tipos de funcionarios. • Cargos.	
4.		Regresa al paso 5 del **Flujo Normal de Eventos.**
Sección 3: Servicios		
	Actor	**Sistema**
1.	Accede a la pestaña Servicios.	
2.		Muestra un listado con los departamentos existentes en la entidad y brinda la posibilidad de: Seleccionar: • Departamentos.

		Y permite: • Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** "Cancelar operación."
3.	Selecciona el dato de: • Departamentos.	
4.		Muestra un listado con los servicios existentes en el sistema para el(los) departamento(s) seleccionado(s) y brinda la posibilidad de: Seleccionar: • Servicios. Y permite: • Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** "Cancelar operación."
5.	Selecciona el dato de: • Servicios.	
6.		Regresa al paso 5 del **Flujo Normal de Eventos**.
Sección 4: Datos como médico		
	Actor	**Sistema**
1.	Accede a la pestaña Datos como médico.	
2.		Brinda la posibilidad de introducir los datos: • Mat. MPPS. • Mat. CM.

		Seleccionar: • Fecha de graduación • Tipo de médico y permite: • Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** “Cancelar operación”.
3.	Introduce el dato de: • Mat. MPPS. • Mat. CM. Selecciona: • Fecha de graduación • Tipo de médico	
4.		Regresa al paso 5 del **Flujo Normal de Eventos**.
Sección 5: Especialidades de graduación		
	Actor	**Sistema**
1.	Accede a la pestaña Especialidades de graduación.	
2.		Muestra un listado con las Especialidades existentes en el sistema y brinda la posibilidad de seleccionar: • Especialidades Y permite: • Aceptar Adicionar usuario.

		• Cancelar operación. Ver **Evento 1:** “Cancelar operación.”
3.	Selecciona el o los datos de: • Especialidades	
4.		Regresa al paso 5 del **Flujo Normal de Eventos.**
Sección 6: Datos de enfermero		
	Actor	**Sistema**
1.	Accede a la pestaña Datos de enfermero.	
2.		Brinda la posibilidad de introducir los datos: • Mat. MPPS. y permite: • Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** “Cancelar operación”.
3.	Introduce el dato de: • Mat. MPPS.	
4.		Regresa al paso 5 del **Flujo Normal de Eventos.**
Sección 7: Datos de bioanalista		
	Actor	**Sistema**
1.	Accede a la pestaña Datos de enfermero	
2.		Brinda la posibilidad de introducir los datos: • Mat. CM. y permite:

		• Aceptar Adicionar usuario. • Cancelar operación. Ver **Evento 1:** “Cancelar operación”.
3.	Introduce el dato de: • Mat. CM.	
4.		Regresa al paso 5 del **Flujo Normal de Eventos.**
Flujos alternos		
Evento 1 “Cancelar la operación”		
	Actor	**Sistema**
1.	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
Evento 2 “Existen datos incompletos”		
	Actor	**Sistema**
1.		Muestra un indicador sobre los campos incompletos.
2.		Regresa al paso 3 del **Flujo Normal de Eventos.**
Evento 3 “Existen datos incorrectos”		
	Actor	**Sistema**
1.		Muestra un indicador sobre los campos incorrectos.
2.		Regresa al paso 3 del **Flujo Normal de Eventos.**

Relaciones	**CU incluidos**	Ver detalles de usuario: Ver caso de uso: Ver detalles de usuario.
	CU extendidos	No existen
Requisitos no funcionales		

Asuntos pendientes	No existen.

Anexo 3. Descripción de las clases.

Tabla 12. Descripción de la clase: UsuarioCrearControlador_configuracionentidad

Nombre	UsuarioCrearControlador_configuracionentidad
Propósito	Esta clase controladora es la encargada de crear los usuarios del sistema y asociarles sus roles, departamentos, servicios y especialidades así como el tipo de usuario y de funcionario que será en la entidad.
Descripción	

Tabla 13. Descripción de la clase: DepartamentosYServiciosManager

Nombre	DepartamentosYServiciosManager
Propósito	Esta clase controladora es la encargada de crear, modificar y eliminar departamentos servicios y especialidades

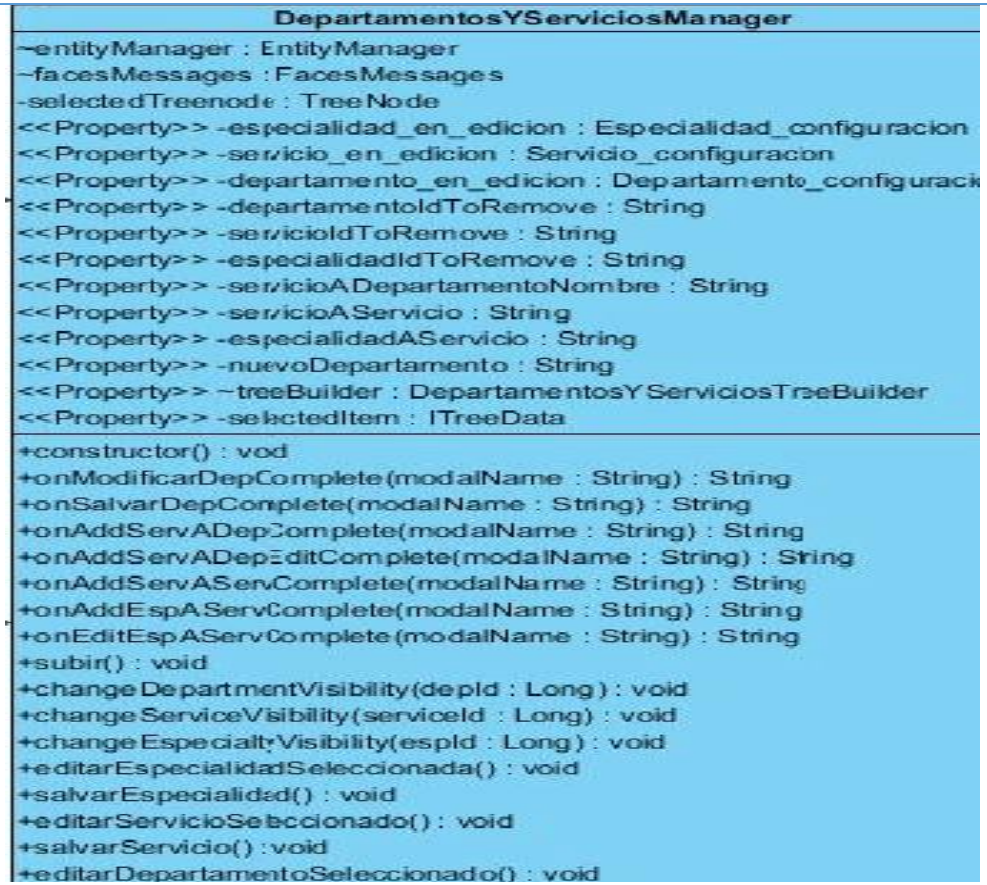
	del sistema.
Descripción	

Tabla 14. Descripción de la clase: FuncionalidadesManager_configuracionentidad

Nombre	FuncionalidadesManager_configuracionentidad
Propósito	Esta clase es la encargada de crear y eliminar categorías y funcionalidades del sistema.

Descripción

Anexo 4. Modelo de datos

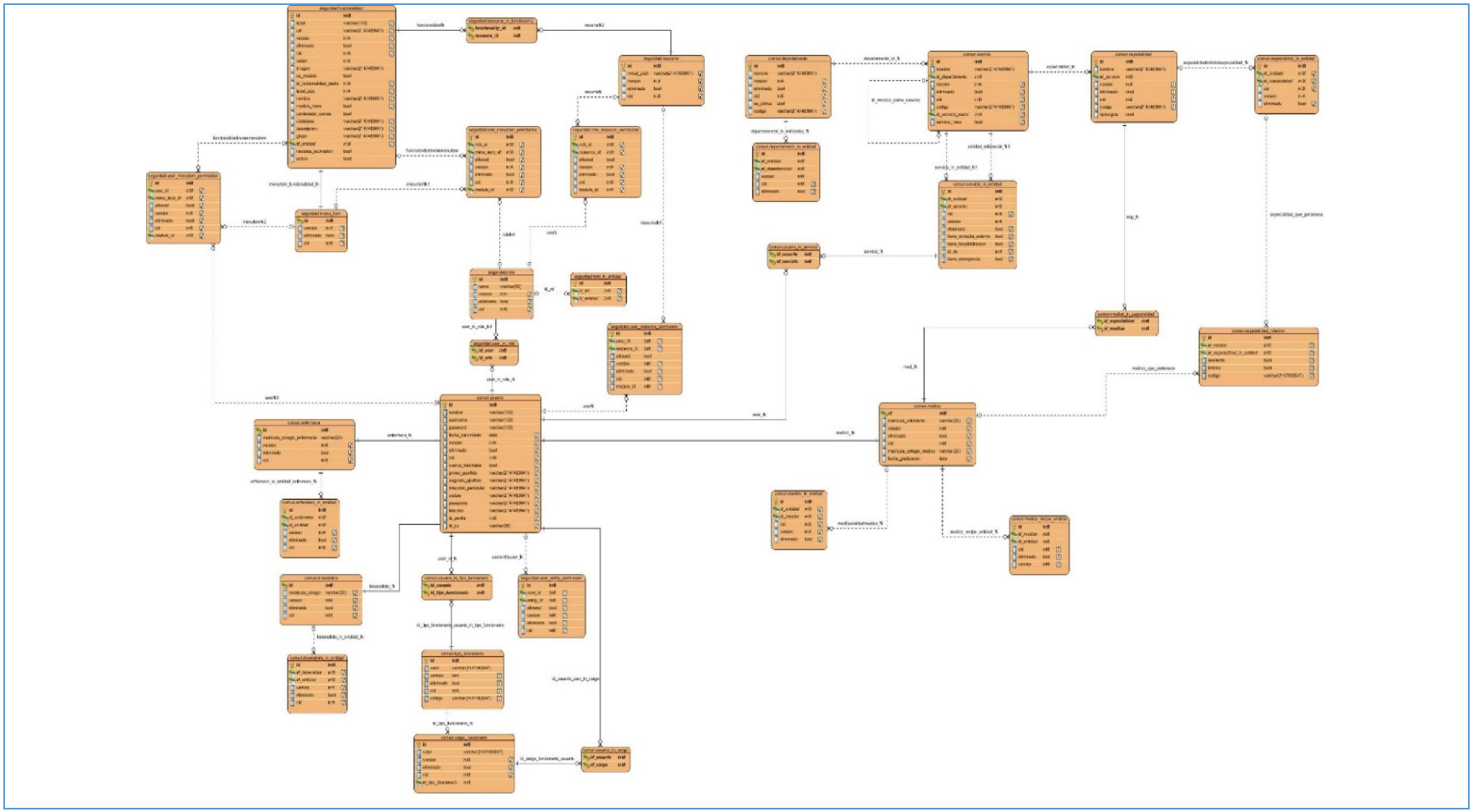


Figura 14. Modelo de datos Configuración_entidad.

Glosario de Términos

Aplicación: En informática, una aplicación es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.

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

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

DDL: Lenguaje de definición de datos que se puede utilizar para crear objetos de base de datos.

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

Herramientas: Son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo.

IDE: Entorno de Desarrollo Integrado, es un programa compuesto por un conjunto de herramientas para un programador, además está compuesto por un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

Mapeo: Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una.

Motor de plantillas: Su función principal es separar el código PHP, como lógica de negocios, del código HTML, como lógica de presentación, y genera contenidos web mediante la colocación de etiquetas.

MVC: Del inglés Model-View-Controller, en español Modelo-Vista-Controlador, patrón utilizado en el diseño y desarrollo web.

Patrones: Los patrones son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Definen una estructura común debido al aprendizaje pasado.

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

Scripts: Un script es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los script es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

UML: Del inglés Unified Modeling Language, en español Lenguaje Unificado de Modelado. Utilizado para modelar procesos y artefactos dentro del desarrollo de un software.