

**Universidad de las Ciencias Informáticas**

**Facultad 2**



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

**Funcionalidades asociadas al componente para el  
proceso de emisión de diagnósticos del Sistema de  
Información Hospitalaria del Centro de Informática  
Médica.**

**Autores: Arianna Carbonell Tamayo**

**Gustavo Lázaro Pérez Bornot**

**Tutores: Ing. Anny Campos Cosme**

**Ing. Omar García Bonelly**

**Co-Tutor: Ing. Dunia Santos Curbelo**

**Ciudad de La Habana, Junio de 2014**

**“Año 56 de la Revolución”**

# Declaración de Autoría

---

## 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 16 días del mes de junio del año 2014.

\_\_\_\_\_  
Arianna Carbonell Tamayo

\_\_\_\_\_  
Gustavo Lázaro Pérez Bornot

\_\_\_\_\_  
Tutor: Ing. Omar García Bonelly

\_\_\_\_\_  
Tutor: Ing. Anny Campos Cosme

\_\_\_\_\_  
Co- Tutor: Ing. Dunia Santos Curbelo

### 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 Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM), desempeñándose como Analista del mismo.

Correo Electrónico: [acampos@uci.cu](mailto:acampos@uci.cu)

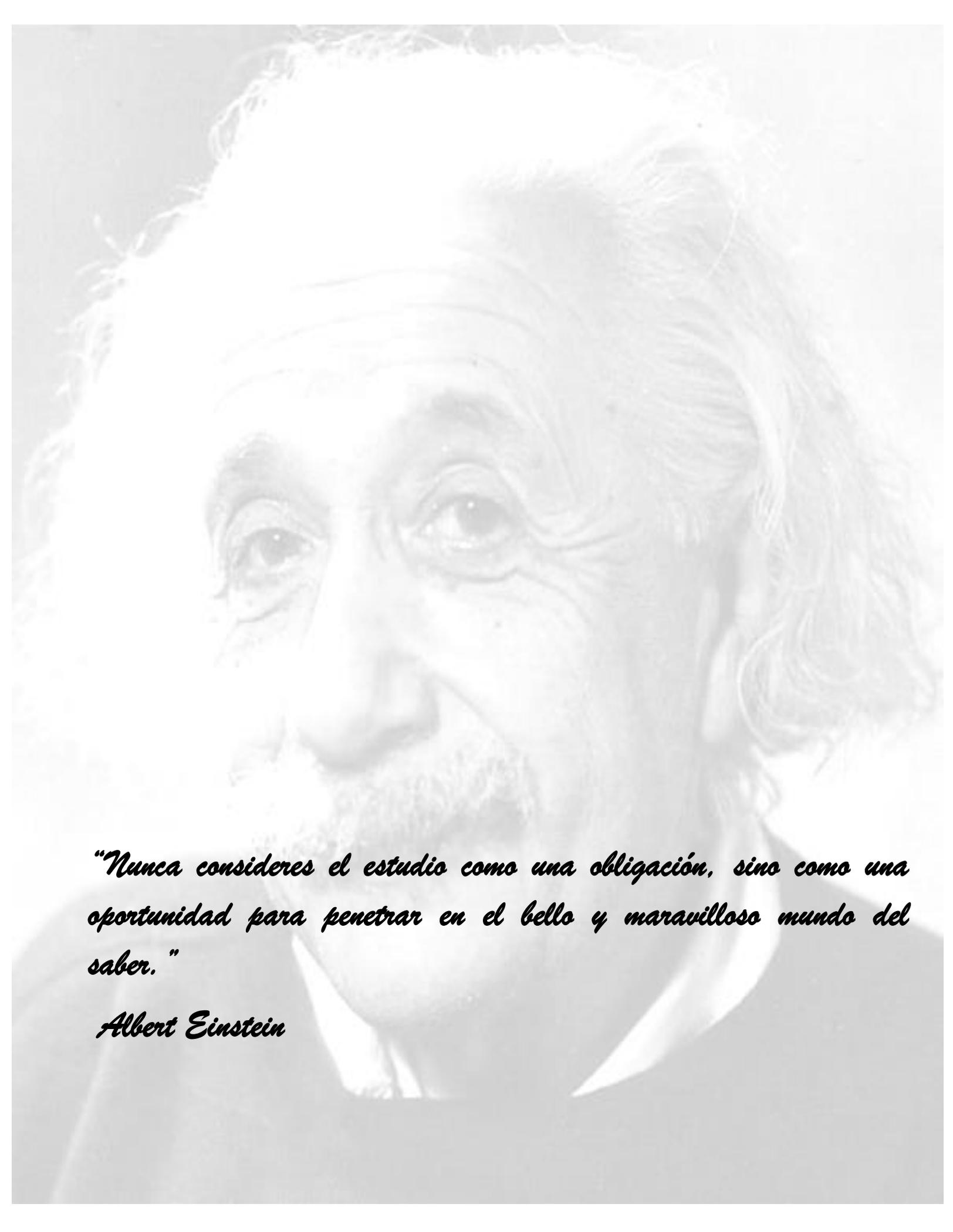
**Ing. Omar García Bonelly:** Graduado de Ingeniería en Ciencias Informáticas en el año 2009 en la Universidad de las Ciencias Informáticas de la Habana. Actualmente se desempeña como profesor universitario en la Universidad de las Ciencias Informáticas (UCI). Posee la categoría docente de Instructor. Desde que se graduó se vinculó a las actividades productivas en el Departamento de Gestión Hospitalaria del Centro de Informática Médica (CESIM), desempeñándose como implementador y jefe del Módulo de Epidemiología del Sistema de Información Hospitalaria (HIS).

Correo Electrónico: [obonelly@uci.cu](mailto:obonelly@uci.cu)

#### Co-Tutor:

**Ing. Dunia Santos Curbelo:** Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2012. Ha participado en varios proyectos de desarrollo vinculados al perfil de salud. Actualmente labora en el Departamento de Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM), desempeñándose como Analista del mismo.

Correo Electrónico: [dsantos@uci.cu](mailto:dsantos@uci.cu)



*"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."*

*Albert Einstein*

# Agradecimientos

---

## **Agradecimientos**

### **Arianna:**

Quiero agradecer a todas aquellas personas que desde un principio confiaron en mí y me alentaron a seguir siempre adelante con la frente en alto.

A mi mamá, a mi papá por apoyarme tanto y confiar en mí.

A mi hermana por sacrificarse tanto por nuestra familia y ser la mejor hermana del mundo.

A mi sobrinito por ser lo más grande que tengo en esta vida.

A mi abuela Ita por protegerme en cualquier lugar que me encontraba.

A mi novio Jordan por apoyarme tanto, darme mucho amor y ayudarme en todo lo que necesitaba.

A mis mejores amigos Lenny y Gustavo por ser como hermanos y apoyarme, soportarme, comprenderme en cada paso de la carrera.

A mis amigas Yazmin, Claudia, Mayelin y Yisel por compartir conmigo estos años de universidad que fueron muy buenos.

A todas las personas que contribuyeron en mi formación profesional y personal.

### **Gustavo:**

Quiero agradecer a mi mamá por su amor, su dedicación y por darlo todo por mí, siempre serás la mejor madre del universo.

A mi papá por ser un ejemplo de sacrificio y voluntad inquebrantable, nunca hubiera llegado hasta aquí si no fuera por ti.

A mi hermana que siempre en los momentos difíciles ha estado para darme una mano.

A mi novia Sady por demostrarme todo su amor sobre todo cuando más estresado estaba, por su confianza y su ayuda, sin ti no hubiera podido alcanzar esta meta.

A mi compañera de tesis que es como mi otra hermana gracias por soportarme y por el impulso que siempre me diste, te voy a extrañar mucho después que nos graduemos.

A mi familia en general, a mis tíos y a mis suegros, gracias a todos, también a todos los profesores que me dieron la mano para ayudarme, muchas gracias a todos.

# Dedicatoria

---

## **Dedicatoria**

Arianna:

A mis padres por ser mi luz y mi ejemplo a seguir.

Gustavo:

A mis padres.

### Resumen

El Centro de Informática Médica desarrolla el Sistema de Información Hospitalaria. El mismo está compuesto por 17 módulos vinculados a cada una de las áreas de cualquier institución hospitalaria, dentro de los cuales Consulta Externa, Hospitalización, Salud Ocupacional y Emergencia están orientados a la atención al paciente. El foco de atención de estos módulos se centra en la Historia Clínica electrónica, la cual facilita el proceso de consulta mediante componentes que permiten, entre otras cosas, la emisión de diagnósticos. Sin embargo, el funcionamiento que posee el componente de diagnóstico provoca que en ocasiones se dificulten la búsqueda y ordenamiento de los padecimientos que conforman el diagnóstico médico.

El objetivo de la presente investigación es desarrollar las funcionalidades asociadas al componente para el proceso de emisión de diagnósticos. Para efectuar estas funcionalidades se utilizó la técnica estadística Series Temporales, la que permitió la obtención del pronóstico de apariciones de cada enfermedad en cada especialidad registrada en el sistema, y luego, ordenarlas partir de este criterio. El desarrollo de estas funcionalidades estuvo guiado por el Proceso Unificado de Desarrollo de Software. Se utilizó Java como lenguaje de programación, Visual Paradigm para el modelado, JBoss Developer Studio como herramienta de desarrollo, JBoss Server como servidor de aplicaciones, PostgreSQL como Sistema Gestor de Base de Datos. Junto a estas herramientas se incorporaron frameworks y librerías. Luego de concluir el desarrollo se obtuvo como resultado un componente que facilita a los especialistas las búsquedas, ordenamiento y selección de las enfermedades.

**Palabras claves:** diagnóstico, pronóstico, series temporales.

# Tabla de Contenidos

---

## Tabla de Contenidos

Introducción .....	1
Capítulo 1. Fundamentación Teórica de los procesos de emisión de diagnósticos en las instituciones hospitalarias.....	6
1.1 Conceptos básicos relacionados con el dominio del problema.....	6
1.2 Principales métodos utilizados para el desarrollo de Sistemas Inteligentes .....	7
1.3 Sistemas automatizados existentes vinculados con el campo de acción.....	13
1.4 Tendencias y tecnologías actuales a considerar .....	18
1.5 Tecnologías utilizadas en el proceso de desarrollo .....	20
1.6 Lenguaje de programación.....	24
1.7 Metodologías de desarrollo .....	24
1.8 Servidor de Base de Datos .....	26
1.9 Herramientas .....	26
Capítulo 2. Características del componente de emisión de diagnósticos .....	30
2.1 Método estadístico utilizado en la solución del problema .....	30
2.2 Propuesta del componente de emisión de diagnósticos.....	36
2.3 Modelo de Dominio .....	37
2.4 Especificación de los requerimientos de software. ....	39
2.5 Modelo de casos de uso del sistema .....	42
Capítulo 3. Análisis y Diseño del componente de emisión de diagnósticos.....	55
3.1 Descripción de la arquitectura. Fundamentación.....	55
3.2 Estrategia de integración y reutilización del sistema .....	56
3.3 Modelo de diseño.....	57
Capítulo 4. Implementación de las funcionalidades asociadas al componente para el proceso de emisión de diagnóstico.....	65
4.1 Modelo de Datos.....	65

# Tabla de Contenidos

---

4.2 Modelo de implementación .....	67
4.3 Seguridad del sistema.....	70
4.4 Estrategias de codificación. Estándares y estilos a utilizar en desarrollo del sistema.....	70
Conclusiones .....	74
Recomendaciones .....	75
Referencias Bibliográficas.....	76
Bibliografía.....	80
Anexos.....	84
Glosario de Términos.....	92

# Tabla de Contenidos

---

## Figuras

Figura 2.1 Representación de la Serie de Tiempo .....	31
Figura 2.2 Representación de la Tendencia.....	31
Figura 2.3 Representación de la Estacionalidad .....	32
Figura 2.4 Representación la Ciclicidad .....	32
Figura 2.5 Representación de la Incertidumbre.....	32
Figura 2.6 Diagrama de Modelo de dominio.....	39
Figura 2.7 Diagrama de actores del componente.....	45
Figura 2.8 Diagrama de Casos de Uso .....	46
Figura 3.1 Diagrama de Clases de Diseño: “Seleccionar enfermedad” .....	58
Figura 3.2 Diagrama de Clases de Diseño: “Listar enfermedades” .....	59
Figura 3.3 Diagrama de Clases de Diseño: “Calcular pronóstico de apariciones” .....	60
Figura 3.4 Clase RegistroAparicion.....	61
Figura 3.5 Clase ReporteWrapper. ....	62
Figura 3.6 Clase CalculoPronosticoScheduled. ....	62
Figura 3.7 Clase EnfermedadCalculada_diagnostico.....	63
Figura 3.8 Clase EnfermedadCalculadaConsList_custom.....	63
Figura 4.1 Modelo de Datos.....	66
Figura 4.2 Diagrama de componentes. ....	68
Figura 4.3 Diagrama de despliegue .....	69

# Tabla de Contenidos

---

## Tablas

Tabla 2.1 Descripción de los actores del sistema.....	44
Tabla 2.2 Descripción del caso de uso: “Calcular pronóstico de apariciones.” .....	47
Tabla 2.3 Descripción del caso de uso: “Listar enfermedades.” .....	49
Tabla 2.4 Descripción del caso de uso: “Seleccionar enfermedad.” .....	54
Tabla 4.1 Descripción de los atributos comunes entre todas las tablas.....	67
Tabla 4.2 Descripción de la tabla enfermedad_calculada. ....	67
Tabla A.1 Observaciones correspondientes a 10 períodos. ....	84
Tabla A.2 Valores del Promedio Móvil. ....	87
Tabla A 3. Valores de calculados de TC y EI. ....	88
Tabla A 4. Ubicación de EI en los períodos correspondientes .....	89
Tabla A 5. Promedio aritmético de cada período.....	89
Tabla A 6. Ubicación de los valores E, EI, PM3. ....	91

## Introducción

La necesidad de gestionar la información se aprecia desde los albores del desarrollo de la humanidad, cuando el hombre en su afán de comunicarse utilizaba diferentes técnicas para intercambiar, transmitir y compartir sucesos o apreciaciones de su quehacer y modo de vida. El vertiginoso crecimiento de los volúmenes de datos que resultaban de los procesos realizados por el hombre, dificultó la manipulación, almacenamiento y centralización de la información derivada de su actividad.

El desarrollo acelerado de las Tecnologías de la Información y las Comunicaciones (TIC) constituye una respuesta a los problemas que se venían presentando en la recolección, almacenamiento y procesamiento de la información, constituyendo una poderosa herramienta para el perfeccionamiento y evolución del conocimiento humano, brindando a su vez, un conjunto de técnicas y métodos para agilizar y optimizar numerosos procesos que responden a necesidades específicas de la sociedad. El sector de la salud es sin dudas dentro de la sociedad actual, uno de los que apuesta por el uso de métodos novedosos que ayuden desde el punto de vista socioeconómico a llevar a cabo uno de sus principales procesos: la atención al paciente.

Las instituciones hospitalarias, como entidades principales del sistema sanitario, generan un gran volumen de información sensible, que en ocasiones puede encontrarse dispersa o no disponible en el momento y forma necesarios. Como consecuencia de esto se ha visto un incremento paulatino del uso de la informática en este sector, siendo las primeras aplicaciones enfocadas al mundo médico, de tipo administrativo-financiero basadas en los hábitos de solución de problemas numéricos y en la facilidad con que estos se resolvían. Luego, atendiendo a las necesidades de crear sistemas que fueran capaces de gestionar todas las áreas de los hospitales y llevar un control de todos los servicios prestados a los pacientes, surgen los denominados Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés).

Los HIS son útiles para agilizar los procesos administrativos y de atención médica en las instituciones hospitalarias, lo que influye de forma directa en la calidad de los servicios brindados al paciente. Estos sistemas responden al funcionamiento de las diferentes áreas que conforman la entidad médica que los utilice, las cuales mantienen una estrecha relación entre sí. Además, permiten la recolección, almacenamiento, procesamiento, recuperación y comunicación de la información relacionada con la atención al paciente y la labor administrativa de todas las actividades de la institución hospitalaria. (1)

# Introducción

---

Cuba es un país que aboga por la incorporación de nuevas soluciones informáticas enfocadas al sector de la salud. En los últimos años se ha ampliado la producción de software y la creación de centros de desarrollo para la salud; como es el caso del Centro de Informática Médica (CESIM), perteneciente a la Universidad de las Ciencias Informáticas (UCI). El mismo protagoniza el desarrollo de productos, sistemas, servicio y soluciones de alta calidad y competitividad.

Unos de los productos insignias de este centro es el Sistema de Información Hospitalaria el cual gestiona la información de los procesos llevados a cabo en las instituciones hospitalarias, compuesto por 17 módulos vinculados a cada una de las diferentes áreas de un hospital. A través del Sistema de Información Hospitalaria, los especialistas pueden realizar un conjunto de acciones que responden a la gestión de la información emitida durante una consulta en cualquier centro de salud, utilizadas en los módulos Consulta Externa, Emergencias, Hospitalización y Salud Ocupacional. Para ello se cuenta con componentes que facilitan la selección y utilización de elementos con información de vital importancia para los especialistas durante la consulta.

Ejemplo de estos componentes son:

- componente con los procesos quirúrgicos a utilizar en una intervención quirúrgica;
- componente que contiene el conjunto de presentaciones de medicamentos existentes según su clasificación ATC (en español, Sistema de Clasificación Anatómica, Terapéutica, Química) y otros elementos distintivos para su recomendación durante la consulta;
- componente que permite la emisión de diagnósticos, el cual contiene una lista de posibles enfermedades a diagnosticar, las cuales están codificadas y estandarizadas por la Clasificación Internacional de Enfermedades en su décima versión (CIE10).

Durante cualquiera de las consultas realizadas por los especialistas, es de selección obligatoria una o varias enfermedades que conforman el diagnóstico del paciente, por lo que se considera el componente de emisión de diagnósticos uno de los elementos más usados en el sistema. Entre las principales dificultades que se han identificado a partir de la experiencia del trabajo con los especialistas del área de la salud, para la emisión del diagnóstico mediante el uso del componente, se encuentra el frecuente uso del paginado del mismo, debido a que el CIE10 contiene alrededor de 26 000 enfermedades posibles a diagnosticar y por pautas de interfaz de usuario que posee el sistema, las páginas deben contener 10 elementos.

A pesar de que el componente permite realizar una búsqueda avanzada a través del Capítulo, Grupo, Categoría, Código y Subcategoría del clasificador, algunos médicos no están familiarizados con la terminología utilizada en la redacción de las enfermedades, por lo que en ocasiones se les dificulta las búsquedas a realizar de los padecimientos que desea seleccionar.

Por otra parte el sistema permite identificar la especialidad por la que el médico está atendiendo en el momento de la consulta, sin embargo las enfermedades mostradas en el componente no poseen criterios de ordenamiento ya sea por especialidad médica u otro específico. Se conoce además, que en una especialidad se diagnostican generalmente el mismo grupo de enfermedades, y en la actualidad el componente visualiza las enfermedades recogidas en el clasificador sin tener en cuenta ningún criterio de agrupamiento, tal es el caso de la especialidad del médico autenticado, por lo que las acciones realizadas para la búsqueda y selección de los padecimientos provoca un aumento en el número de consultas realizadas a la base de datos del sistema.

Tomando en cuenta la situación descrita anteriormente, surge el siguiente **problema científico**: el funcionamiento actual del componente para el diagnóstico del Sistema de Información Hospitalaria del Centro de Informática Médica dificulta la búsqueda, ordenamiento y selección de las enfermedades en el proceso de emisión de diagnóstico en las instituciones hospitalarias. En correspondencia con el problema el **objeto de estudio** lo constituye: el proceso de emisión de diagnósticos en las instituciones hospitalarias.

El **campo de acción** está centrado en: el proceso de emisión de diagnósticos en instituciones hospitalarias basados en técnicas de sistemas inteligentes. Para dar solución a dicho problema se definió como **objetivo general**: desarrollar las funcionalidades asociadas al componente para el proceso de emisión de diagnóstico del Sistema de Información Hospitalaria del Centro de Informática Médica que faciliten a los especialistas la búsqueda, ordenamiento y selección de las enfermedades.

En aras de cumplir el objetivo anteriormente planteado se definieron las siguientes **tareas de la investigación**:

- Analizar los sistemas y tendencias actuales que existen para la implementación de herramientas de apoyo a la emisión de diagnósticos médicos.

- Analizar los diferentes métodos y técnicas utilizados en sistemas inteligentes para el apoyo a la toma de decisiones que permitan dar solución al problema planteado.
- Asimilar las tecnologías y herramientas del departamento de Gestión Hospitalaria del Centro de Informática Médica.
- Obtener mediante el Proceso Unificado de Desarrollo, los artefactos correspondientes a los flujos de trabajo: Modelado de negocio, Gestión de requerimientos, Diseño e Implementación relacionados con el proceso de emisión de diagnósticos.
- Desarrollar las funcionalidades asociadas al componente de diagnóstico del Sistema de Información Hospitalaria del CESIM.

Al desarrollar el componente de apoyo a la emisión de diagnósticos del Sistema de Información Hospitalaria del CESIM se esperan obtener los siguientes **beneficios**:

- Se facilitarán las búsquedas y obtención de las enfermedades recogidas en el componente para la emisión de diagnósticos.
- Se aumentará la calidad de los informes estadísticos generados en las instituciones hospitalarias.
- Se reducirá la cantidad de consultas innecesarias realizadas a la base de datos.
- Se obtendrá una mayor organización del componente, a través de la agrupación de las enfermedades por la especialidad, que permita reducir a los especialistas el tiempo de obtención de los diagnósticos deseados.

Este documento se encuentra estructurado en cuatro capítulos, el primero de ellos, **Fundamentación Teórica de los procesos de emisión de diagnósticos en las instituciones hospitalarias**, se estudian los conceptos y sistemas relacionados con el proceso de emisión de diagnóstico médico, así como se analizan los métodos que darán solución al problema identificado. Se plantean además las tendencias y tecnologías más utilizadas para la elaboración de las funcionalidades propuestas.

Seguidamente el capítulo **Características del componente de emisión de diagnósticos**, contiene un marco conceptual asociado a la información que será manipulada por el sistema, llegándose a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

El tercer capítulo **Análisis y Diseño del componente de emisión de diagnósticos** se centra en la modelación detallada y la construcción de la estructura de la aplicación. En el cuarto y último, **Implementación de las funcionalidades asociadas al componente para el proceso de emisión de**

## Introducción

---

**diagnóstico**, se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los procesos hospitalarios asociados al área.

## **Capítulo 1. Fundamentación Teórica de los procesos de emisión de diagnósticos en las instituciones hospitalarias.**

En el presente capítulo se exponen las bases conceptuales y teóricas para el desarrollo del componente para la emisión de diagnósticos para el Sistema de Información Hospitalaria del CESIM. Se abordan los principales conceptos que facilitan un mejor entendimiento de los procesos asociados al campo de acción. Se realiza un estudio de los sistemas inteligentes que pueden brindar una solución al problema planteado, del mismo modo, son analizados algunos sistemas informáticos que brinden apoyo a la emisión de diagnósticos médicos. Además se describen las tecnologías, metodología y herramientas con las cuales se llevará a cabo el proceso de desarrollo del software.

### **1.1 Conceptos básicos relacionados con el dominio del problema**

#### **Diagnóstico**

El diagnóstico o propedéutica clínica es el procedimiento por el cual se identifica una enfermedad, entidad nosológica, síndrome, o cualquier condición de salud-enfermedad. En términos de la práctica médica, el diagnóstico es un juicio clínico sobre el estado psicofísico de una persona, que generalmente es proporcionado por un médico. (2)

#### **Inteligencia Artificial**

La inteligencia artificial es considerada una rama de la computación que relaciona un fenómeno natural con una analogía artificial a través de programas de computador. La inteligencia artificial puede ser tomada como ciencia si se enfoca hacia la elaboración de programas basados en comparaciones con la eficiencia del hombre, contribuyendo a un mayor entendimiento del conocimiento humano. (3) La inteligencia artificial, por lo tanto, consiste en el diseño de procesos que, al ejecutarse sobre una arquitectura física, producen resultados que maximizan una cierta medida de rendimiento. Estos procesos se basan en secuencias de entradas que son percibidas y almacenadas. (4)

#### **Series Temporales**

Una serie temporal o cronológica es una secuencia de datos, observaciones o valores, medidos en determinados momentos y ordenados cronológicamente. Los datos pueden estar espaciados a intervalos

iguales (como la temperatura en un observatorio meteorológico en días sucesivos al mediodía) o desiguales (como el peso de una persona en sucesivas mediciones en el consultorio médico, la farmacia, etc.). Para el análisis de las series temporales se usan métodos que ayudan a interpretarlas y que permiten extraer información representativa sobre las relaciones subyacentes entre los datos de la serie o de diversas series y que permiten en diferente medida y con distinta confianza extrapolar o interpolar los datos y así predecir el comportamiento de la serie en momentos no observados, sean en el futuro (extrapolación pronóstica), en el pasado (extrapolación retrógrada) o en momentos intermedios (interpolación). (5)

### **CIE10**

Clasificación Internacional de Enfermedades (CIE) creada por la Organización Mundial de la Salud (OMS) en 1948, con el objetivo de estandarizar los nombres de las enfermedades, y además, asignarles a cada una de ellas un código único que la representa. Este provee los códigos para clasificar las enfermedades, hallazgos anormales, denuncias, circunstancias sociales y causas externas de daños y/o enfermedad. Dicho estándar tiene como propósito permitir el registro sistemático, análisis, interpretación, y comparación de los datos de mortalidad y morbilidad recolectados en diferentes países o áreas, atendiendo a diferentes épocas del año. Una de sus funciones es convertir los términos diagnósticos y otros problemas de salud, de palabras a códigos alfanuméricos permitiendo un fácil almacenamiento y posterior recuperación para el análisis de la información. Además permite realizar estudios de la situación de la salud de grupos de población y hace posible conocer la incidencia y prevalencia de las enfermedades.

El CIE en su décima versión consta de tres volúmenes, estos están estructurados por capítulos que se refieren a familias de enfermedades que agrupan las categorías por códigos alfanuméricos que constan de tres dígitos siendo el primero una letra y los dos siguientes son números, así tenemos en teoría códigos desde A00 a Z99. Está exenta la letra U ya que los códigos U00-U49 son para ser utilizados en la asignación provisional de nuevas enfermedades de que posean un origen desconocido. (6)

### **1.2 Principales métodos utilizados para el desarrollo de Sistemas Inteligentes**

Desde hace algunas décadas se ha visto un gran desarrollo en la aplicación de la informática en el campo de la salud, donde los sistemas inteligentes han jugado un papel fundamental en el apoyo a la

toma de decisiones, además de alcanzar un puesto como gestores de información, reduciendo la sobrecarga que pesa sobre la rutina clínica, almacenando, indexando y recuperando todo tipo de datos médicos. En los últimos años, se han desarrollado para el mercado médico una gran variedad de sistemas informáticos basados en este enfoque y diseñados para ayudar en la dosificación de la medicación, mantenimiento de la salud, diagnóstico y otras decisiones clínicas relevantes. En los primeros años de la década de los 70 se produjo un gran auge para los investigadores en el campo de la Inteligencia Artificial en la medicina, desarrollándose un sinnúmero de sistemas de apoyo a la toma de decisiones en el campo del diagnóstico médico. El uso de sistemas informáticos como CASNET, DENDRAL y MYCIN constituyen ejemplos de herramientas basadas en métodos inteligentes orientados al proceso de emisión de diagnósticos médico. Actualmente con el uso de técnicas inteligentes tales como: los Sistemas Basados en Casos, Sistemas basados en Reglas, Redes Neuronales, Regresión Lineal, Regresión Múltiple y Series Temporales, en el área del diagnóstico médico se han afrontado problemas complejos y se ha logrado una mayor fiabilidad en los resultados obtenidos. La identificación de un tipo específico de enfermedad es mucho más precisa y sencilla en la actualidad, puesto que muchos de los sistemas de diagnóstico que se aplican hoy en día, poseen una gran cantidad de conocimiento plasmado generalmente como bases de datos.

### **1.2.1 Sistemas Basados en Casos**

Los sistemas basados en casos usan el conocimiento de experiencias anteriores cuando se enfrenta a nuevos casos. Un caso en términos generales, hace referencia a una situación difícil, aunque los casos pueden ser descritos por un vector de atributo-valor, los sistemas basados en casos generalmente utilizan una estructura de datos jerárquica. Depende de una base de datos en la cual se encuentran los casos previamente analizados, los que han sido diseñados de forma tal que faciliten el reconocimiento de casos similares

El razonamiento basado en casos ofrece una variedad de herramientas para el análisis de los datos. Los casos no solamente son almacenados, también son analizados para obtener las características más representativas y así compararlas de éste modo con los casos que se estén analizando. Con el fin de que la organización jerárquica de los datos pueda incorporar mecanismos de explicación adicionales.

Cuando es aplicado el razonamiento basado en casos al análisis de datos médicos se tienen que identificar una gran cantidad de preguntas no triviales, incluyendo el nivel de similaridad usado en las

medidas, la vigencia de los casos antiguos (dado que el conocimiento médico cambia con marcada rapidez), como son ejecutadas las diferentes soluciones (tratamientos, acciones) por los diferentes médicos. Muchos sistemas de Sistemas Basado en Casos han sido usados, adaptados o implementados para dar soporte a los diagnósticos y al análisis de datos en medicina. (7)

### **1.2.2 Sistemas Basados en Reglas**

La base de conocimiento y los datos son los principales elementos que conforman un sistema basado en reglas. Los datos están formados por la evidencia o los hechos conocidos en una situación particular. Este elemento es dinámico, es decir, puede cambiar de una aplicación a otra. El conocimiento se almacena en la base de conocimiento y consiste en un conjunto de objetos y un conjunto de reglas que gobiernan las relaciones entre esos objetos. La información almacenada en la base de conocimiento es de naturaleza permanente y estática, es decir, no cambia de una aplicación a otra, a menos que se incorporen al sistema experto elementos de aprendizaje.

El motor de inferencia combina los hechos y las preguntas particulares, utilizando la base de conocimiento, seleccionando los datos y pasos apropiados para representar los resultados. Se encarga de obtener conclusiones comenzando desde el conocimiento abstracto hasta el conocimiento concreto. El motor de inferencia usa los datos y el conocimiento para obtener nuevas conclusiones. Por ejemplo, si la premisa de una regla es cierta, entonces la conclusión de la regla debe ser también cierta. Los datos iniciales se incrementan incorporando las nuevas conclusiones. Por ello, tanto los hechos iniciales o datos de partida como las conclusiones derivadas de ellos, forman parte de los hechos o datos de que se dispone en un instante dado. Las conclusiones pueden clasificarse en dos tipos: simples (son las que resultan de una regla simple) y compuestas (son las que resultan de más de una regla).

Para obtener conclusiones, los expertos utilizan diferentes tipos de reglas y estrategias de inferencia y control. Entre las estrategias de inferencia se encuentran, encadenamiento de reglas y encadenamiento de reglas orientado a un objetivo (más conocidas por encadenamiento hacia adelante y encadenamiento hacia atrás respectivamente), que son utilizadas por el motor de inferencia para obtener conclusiones simples y compuestas. (8)

### 1.2.3 Redes Neuronales

Una red neuronal artificial (RNA) es un sistema de procesadores paralelos conectados entre sí en forma de grafo dirigido. Esquemáticamente cada elemento de procesamiento de la red (neuronas) se representa como un nodo. Estas conexiones establecen una estructura jerárquica que, tratando de emular la fisiología del cerebro, busca nuevos modelos de procesamiento para solucionar problemas concretos del mundo real. La utilidad de las RNA radica en la capacidad de aprender, reconocer y aplicar relaciones entre objetos y tramas de objetos del mundo real. Su comportamiento depende de los pesos de las funciones de entrada-salida (funciones de transferencia). Su novedad radica en el aprendizaje secuencial, el hecho de utilizar transformaciones de las variables originales para la predicción y la no linealidad del modelo. Permiten aprender en contextos difíciles, sin precisar la formulación de un modelo concreto. (9)

### 1.2.4 Técnica estadística

La estadística es una ciencia formal y una herramienta que estudia el uso y los análisis provenientes de una muestra representativa de datos. Busca explicar las correlaciones y dependencias de un fenómeno, de ocurrencia en forma aleatoria o condicional. Su objetivo principal es investigar la causalidad, y en particular extraer una conclusión como consecuencia del comportamiento de la información. Tradicionalmente la solución a problemas de predicción se ha llevado a cabo desde la óptica de modelos estadísticos, que tienen como base el uso de procedimientos y algoritmos matemáticos. Esta manera de extraer, analizar, procesar y deducir cierta cantidad de información, es considerada, a grandes rasgos, una técnica estadística. (10)

Las técnicas estadísticas son un tipo particular de técnicas matemáticas de procesamiento de datos. Estas le dan un sentido aritmético y real al proceso que se estudia, determinando relaciones y analogías en ciertas variables de decisión para deducir otras. Muchos de los Sistemas Inteligentes actuales han encontrado como solución a sus problemas de predicción, la utilización de técnicas estadísticas.

#### **Regresión lineal simple:**

La variable dependiente se puede obtener como combinación lineal de los parámetros y existe una sola variable independiente. Se parte del supuesto de que la más simple relación entre dos variables es una línea recta. Se presentan pares de observaciones  $(x, y)$ , tal que el valor de “y” está condicionado

únicamente por el valor numérico de “x”. Al igual que la ecuación de una recta en el plano, su modelo está representado de la siguiente manera:

$$Y_i = a_0 + a_1 x_i + \varepsilon_i \quad i=1, \dots, n$$

Donde  $A_0$  y  $A_1$  son coeficientes del modelo que hay que determinar y  $E_i$  es un parámetro calculable (término de error) que depende del tipo de solución que se pretenda encontrar en el problema y del efecto adicional de un conjunto de variables no relevantes pero existentes.

Regresión múltiple:

La variable dependiente depende de un grupo definido de variables independientes. Frecuentemente los pares de observaciones se representan en un modelo multidimensional de la siguiente manera:  $(X_{1p}, X_{2p}, \dots, X_{ip}, Y_i)$ . Siguen el siguiente modelo matemático:

$$Y_i = a_0 + a_1 x_i + \dots + a_p x_{ip} + \varepsilon_i \quad i=1, \dots, n$$

Donde  $E_i$  es el error del modelo y los  $A_{ip}$  son los coeficientes a calcular.

De manera general, el proceso de técnicas de regresión estadística se centra básicamente en dos aspectos fundamentales:

- Estimar los parámetros del modelo de regresión. Este proceso es llamado ajuste del modelo a los datos.
- Chequear que tan bueno es el modelo ajustado. El resultado de este chequeo puede indicar si el modelo es razonable o si el ajuste original debe ser modificado. (11)

### **Series Temporales**

Una serie temporal o cronológica es una secuencia de datos, observaciones o valores, medidos en determinados momentos y ordenados cronológicamente. Los datos pueden estar espaciados a intervalos iguales o desiguales. Desde el punto de vista probabilístico una serie temporal es una sucesión de variables aleatorias indexadas según un parámetro creciente con el tiempo. (12) Son un conjunto de

observaciones producidas en determinados momentos durante un período, ya sea diario, semanal, trimestral o anual, generalmente a intervalos iguales y se usan para estudiar la relación entre diversas variables que cambian con el tiempo y su influencia entre sí. (13)

Para el análisis de las series temporales se usan métodos que ayudan a interpretarlas y que permiten extraer información representativa sobre las relaciones subyacentes entre los datos de la serie o de diversas series y que permiten en diferente medida y con distinta confianza extrapolar o interpolar los datos y así predecir el comportamiento de la serie en momentos no observados, sean en el futuro, en el pasado o en momentos intermedios. (14)

Después de realizado el estudio de los principales métodos para sistemas inteligentes, se definió que la técnica que mejor se ajusta para cumplir el objetivo de la presente investigación es el método de Series Temporales, ya que para realizar una predicción de los valores futuros de una o varias variable se utiliza como única información, valores pasados o historiales de la serie, midiendo la evolución de la o las variables en cuestión. Este método trata de identificar patrones históricos procesados con anterioridad y luego suponiendo que se mantendrán en un futuro se extrapolan. Un modelo de serie temporal puede predecir tendencias basadas únicamente en el conjunto de datos original utilizado para crear el modelo. Además es posible agregar nuevos datos al modelo, al realizar una predicción e incorporar automáticamente los nuevos datos en el análisis de tendencias.

Entre las principales razones para la utilización de las series temporales se encontraron las siguientes:

- La información histórica se encuentra almacenada de forma cronológica.
- Puede ser que se desee sólo predecir qué es lo que va a pasar y no el por qué pasa.
- Puede ser muy útil a la hora de realizar una predicción de la cantidad de elementos que puede poseer una variable.

Además, se tuvo en cuenta que: actualmente el Sistema de Información Hospitalaria no cuenta con una base de casos lo suficientemente sólida para establecer comparaciones y determinar un caso semejante, además de a la hora de dar solución a un nuevo problema, los datos iniciales con los que cuenta el sistema para resolver la situación son exiguos para establecer semejanzas válidas para la resolución de un nuevo caso. En los sistemas basados en reglas se utilizan un conjunto de premisas para llegar a una conclusión lo más acertada posible y a pesar de utilizar datos históricos, solo se recomienda su uso

cuando el número de acciones es pequeño además de que actualmente no se cuenta con un experto que defina todas las reglas necesarias para el desarrollo de las funcionalidades en el sistema. El uso de redes neuronales puede no ser práctico, pues recuperar un caso de cientos de posibles alternativas, haría demasiado grande la red y por consiguiente el proceso de recuperación se torna lento. Además podría ser complicado el entrenamiento de una red que contenga datos clínicos y aprender acerca de los casos en cuestión, dentro de una estructura de conexiones. Por otra parte aún, cuando los modelos de regresión tratan de predecir los valores de una variable dependiente en términos de una o más variables independientes, no se establece una relación formal de causa-efecto; solo se trata de una relación matemática empírica.

### **1.3 Sistemas automatizados existentes vinculados con el campo de acción**

En la actualidad se ha generalizado el uso de sistemas orientados a la gestión hospitalaria, los cuales están enfocados a elevar el nivel de vida de la sociedad. Durante la investigación se realizó un estudio de algunos sistemas dirigidos al proceso de emisión de diagnóstico médico, destacando sus principales funcionalidades, ventajas y herramientas con las cuales fueron desarrollados.

#### **Sistemas asociados al campo de acción**

**Symcat** (Aplicación asistida por ordenadores basada en síntomas)

Symcat es una solución tecnológica versátil y muy poderosa, que combina los datos agregados de los registros de salud de los pacientes, con los síntomas del usuario y los datos demográficos que informan los diagnósticos. Esta aplicación web que además es compatible con la tecnología móvil, fue creada por estudiantes de medicina, en la universidad Johns Hopkins de los Estados Unidos. Una de las principales funcionalidades que brinda este software es que a través de una interfaz permite a los usuarios introducir información acerca de la duración de sus síntomas, su género, e historia médica para luego sugerir uno o varios diagnósticos con una probabilidad de ocurrencia. Por otra parte, calcula la frecuencia de los síntomas y de las enfermedades a partir de los datos clínicos y usa esa información para predecir cuál podría ser el problema, además da una sugerencia de tratamiento a seguir. Este sistema no se especializa en un área de la medicina, sino que abarca todas las ramas de esta ciencia logrando una mayor usabilidad del sistema. (15)

### **SDTEC** (Sistema de Detección y Tratamiento de Enfermedades Cutáneas)

SDTEC es un sistema experto creado en la universidad Pontificia Comillas, Madrid, España, encargado de la detección y tratamiento de enfermedades cutáneas, dirigido a la especialidad médica dermatología. Fue desarrollado en el lenguaje de programación Java. SDTEC está pensado para estudiantes de medicina, residentes, otros estudiantes de ciencias de la salud y médicos generales, aunque especialmente dirigido para los dermatólogos o estudiantes de dicha especialidad. La aplicación no sólo se encargará de ofrecer la(s) enfermedad(es) más apropiada(s) para las entradas de síntomas y observaciones clínicas, sino que ofrece un tratamiento de acuerdo a las especificaciones descritas en el sistema, estudiando los antecedentes del paciente y posibles incompatibilidades con dicho tratamiento, es decir, es sistema, además de ofrecer la solución más adecuada a los síntomas introducidos permite gestionar los tratamientos y los pacientes que han sido diagnosticados a través de este sistema. Este software ofrece al usuario cuatro opciones principales:

- **Diagnóstico** (permite al usuario diagnosticar una enfermedad a partir de los síntomas): A través de una serie de ventanas, se le pregunta al usuario datos sobre el paciente a tratar. Con los datos introducidos y los síntomas percibidos, se hace un primer acercamiento a la solución, llamado hipótesis, donde se muestra el conjunto de posibles soluciones a los datos introducidos.
- **Consulta** (permite la consulta de enfermedades): Esta opción no es interactiva para el usuario ya que se trata únicamente de búsqueda de información, ya que se introduce la enfermedad que se desea consultar y se muestra la información correspondiente a dicha enfermedad.
- **Tratamiento** (consiste en la consulta de tratamientos específicos para una enfermedad): Muy similar a la opción de “Consulta”, consiste en la búsqueda del tratamiento correspondiente con la enfermedad introducida.
- **Estadísticas**: Ofrecen al usuario información estadística en base a la edad, enfermedad, fecha, entre otras informaciones.

Este programa no diagnosticará todas las lesiones dermatológicas, sino que han sido limitadas a un cierto número de enfermedades ya que éstas no se limitan únicamente al diagnóstico visual. (16)

### **Sistema asistente de diagnóstico de enfermedades cardiovasculares a través de síntomas**

El Sistema asistente de diagnóstico de enfermedades cardiovasculares a través de síntomas es un ayudante virtual para el médico especialista, médico clínico y estudiantes de cardiología, creado en la universidad de Buenos Aires, Argentina. Este sistema mediante la inserción de los síntomas asiste al usuario, ya sea a un especialista o un médico clínico sobre los posibles síntomas de las enfermedades cardiovasculares y de las posibles anomalías cardíacas, también puede mostrar actividad cardíaca mediante la información importada y generada desde un medio externo, mostrando la gráfica de un electrocardiograma. El sistema tiene a su vez la posibilidad de generar y mostrar un reporte básico de los resultados obtenidos en un archivo de texto.

Dicho software además de brindar un diagnóstico preciso, también tiene la capacidad de percatarse si hay síntomas no específicos de una determinada enfermedad, alertando al usuario sobre los posibles síntomas que no pertenecen a un cuadro determinado. Posee pantallas de dialogo interactivas sensitivas al contexto, en las que es posible guiar al usuario a través de las distintas situaciones que se presentan para llegar al diagnóstico correcto de las enfermedades cardiovasculares. (17)

**SIGHO** (Sistema de Información para la Gerencia Hospitalaria).

SIGHO es un Sistema de Información para la Gerencia Hospitalaria, basado en la Norma Oficial Mexicana NOM-168-SSA1-1998 referente al resguardo y uso del expediente clínico electrónico para facilitar las actividades de gerencia dentro del hospital. Está integrado por 14 módulos entre los que incluye el de consulta externa que permite al médico tener acceso de manera sencilla al historial del paciente; incluye los distintos diagnósticos que le han emitido, el tratamiento recibido, así como evoluciones y estudios auxiliares realizados.

Para el diagnóstico clínico de las enfermedades el SIGHO utiliza la Clasificación Internacional de Enfermedades en su décima revisión, CIE10. A la hora de diagnosticar, el sistema muestra una ventana donde se puede realizar una búsqueda de enfermedades ya sea por la clave o por el nombre. Según la búsqueda que se realice el sistema mostrará un listado de las enfermedades organizadas de forma descendente según la frecuencia con que fueron diagnosticadas. (18)

### **SEADE** (Sistema experto para el análisis y detección de enfermedades)

SEADE es un sistema experto creado en el Instituto Superior Minero Metalúrgico, Cuba, cuyo objetivo es diagnosticar enfermedades a partir de un conjunto de síntomas y sugerir un tratamiento para cada paciente. Está diseñado para ser utilizado por profesionales y estudiantes de medicina como un medio de apoyo para realizar su trabajo. SEADE fue implementado utilizando dos lenguajes de programación: Java y Prolog. En Java se creó la interfaz de usuario y de forma embebida en Prolog se diseñó todo el mecanismo de razonamiento y consultas. SEADE cuenta con una interfaz gráfica simple que permite recordar fácilmente objetos u operaciones, además posee eficientes mecanismos internos capaces de ayudar en la toma de decisiones.

Su principal funcionalidad es que una vez identificada una enfermedad, muestra al usuario una serie de recomendaciones para dar tratamiento a la misma. Además cuando un paciente ha sido atendido y se ha logrado identificar cuál es la enfermedad que presenta y el tratamiento que se le debe dar, existe una gran posibilidad de que otro paciente, cuyos síntomas y signos sean similares, sufra de la misma enfermedad, a pesar de que necesite otro tratamiento. Por estas razones, SEADE posee la capacidad de inferir conocimiento durante cada consulta que se realice lo cual agiliza el proceso de identificación de una enfermedad. Este sistema no pretende servir como un consultor médico, y por ende, no debe ser usado por nadie de forma independiente, por lo que es tomado como un apoyo a los médicos que lo utilicen. (19)

### **DITRITS** (Sistema experto para el diagnóstico y tratamiento de enfermedades de transmisión sexual)

DITRITS es un sistema experto para el diagnóstico y tratamiento de Infecciones de Transmisión Sexual, que fue desarrollado en Cuba por la Universidad de Cienfuegos en conjunto con especialistas de segundo grado del Hospital Provincial Dr. Gustavo Aldereguía Lima. Esta aplicación de escritorio cuenta con un conjunto de elementos como: Base de conocimientos, Base de hechos, Motor de inferencia, Subsistema de explicación y la Interfaz de usuario. La principal funcionalidad de esta aplicación informática es generar a partir del ingreso de diferentes síntomas asociados a las enfermedades de transmisión sexual, un posible diagnóstico del padecimiento así como su tratamiento.

Este sistema informático posee un conocimiento sobre las diferentes enfermedades de transmisión sexual. Dicho conocimiento se obtuvo de un conjunto de expertos, tras varias investigaciones y entrevistas,

durante las cuales se determinaron las principales infecciones de transmisión sexual en el territorio, sus características distintivas, sus rasgos comunes y las diferentes formas en que la enfermedad suele presentarse. Luego de un fructífero y largo trabajo de mesa se obtuvieron las siguientes patologías a diagnosticar en el sistema, Gonorrea, Vaginitis, Cervicitis, Sífilis, Herpes genital, Condiloma, Granuloma inguinal, Hepatitis B, VIH, Vaginosis bacteriana, Trichomoniasis y Candidiasis.

DITRITS presenta una interfaz de usuario amigable facilitando la navegación en el sistema. La aplicación está compuesta por diferentes secciones entre las que se destacan:

- **Datos Generales:** se registran los datos personales del paciente así como la fecha de la consulta y su número de historia clínica.
- **Motivos de Consulta:** se registran las causas por las que el paciente solicita atención médica especializada.
- **Cuadro Clínico:** Se registran los síntomas y signos que presenta el paciente.
- **Diagnóstico:** A partir de la información recogida, el sistema realiza un análisis y propone un posible diagnóstico.
- **Tratamiento:** Una vez emitido el diagnóstico, el sistema propone un tratamiento a seguir para erradicar o mitigar la enfermedad.

Dicho software puede ser usado no sólo en el diagnóstico, sino como herramienta de apoyo a la docencia tanto en centros educacionales como en centros de asistencia primaria a la población. (20)

**SEGEDIS** (Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias)

SEGEDIS es un Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias. Fue creada en la Universidad de las Ciencias Informáticas, Cuba. Es una aplicación informática de tipo web desarrollada sobre el framework Symfony, ExtJS, Drools y los lenguajes de programación Java y PHP. Este software posee una interfaz amigable y segura, acorde con el personal que trabajará con la misma. La aplicación permite la autenticación del usuario como administrador y como genetista permitiendo los genetistas del Centro Nacional de Genética Médica (CNGM) que poseen los permisos de administración agregar nuevas enfermedades, actualizando sistemáticamente la aplicación. (21)

SEGEDIS posee como principales funcionalidades la emisión de diagnósticos, además de proporcionar tanto al médico como al paciente información de una o varias enfermedades que contengan los signos seleccionados con anterioridad. Posee una sección que está relacionada con los detalles de la enfermedad, en la cual se muestran imágenes del padecimiento clínico. Además brinda otras funcionalidades como mostrar un listado de las enfermedades genéticas con dismorfias que contenga la base de datos, mostrar un listado de signos clínicos con sus respectivas características, adicionar enfermedades y signos clínicos, eliminar estos, agregar signos a criterios y hacer una búsqueda de las enfermedades que presentan los signos seleccionados.

En el mundo existe una gran variedad de sistemas que tienen como objetivo automatizar los procesos de emisión de diagnósticos que son brindados en clínicas y hospitales. La utilización de estos sistemas no resolvería el problema identificado, debido a que estas soluciones no se pueden integrar con el Sistema de Información Hospitalaria del CESIM, pues no cumplen con la arquitectura definida para dicha solución. Por otra parte estos sistemas de apoyo a la emisión de diagnóstico, predicen los padecimientos a partir de la inserción de datos, síntomas o enfermedades. Además la mayoría están enfocados solamente a un área específica de la medicina o a un conjunto de enfermedades delimitadas. Por otra parte estos sistemas toman la responsabilidad de asignar un diagnóstico a los pacientes, siendo esto riesgoso pues comprometen su salud a un sistema informático.

Todas estas razones ratifican la elaboración de funcionalidades para el componente de diagnósticos, teniendo en cuenta los sistemas analizados y tomándolos como punto de partida para el desarrollo de la presente investigación. A continuación se mencionan las tecnologías y herramientas seleccionadas para su desarrollo.

### **1.4 Tendencias y tecnologías actuales a considerar**

#### **Patrones de arquitectura y diseño**

Los patrones arquitectónicos, son patrones de software que ofrecen soluciones a problemas de arquitectura de software. Especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. Expresan un esquema organizativo estructural fundamental para sistemas de software. Resuelven problemas arquitectónicos, adaptabilidad a requerimientos cambiantes, modularidad, acoplamiento, entre otros. La solución que se

plantea es la creación de patrones de llamadas entre objetos (similar a los patrones de diseño), decisiones y criterios arquitectónicos, empaquetado de funcionalidad. (22)

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. Constituye 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. (23)

### **Patrón Modelo-Vista-Controlador (MVC)**

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML (Hyper Text Markup Language) y el código es el que provee los datos dinámicos a la página. El modelo lo constituyen los datos con los que trabaja la aplicación y la lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista. (24)

**Modelo:** Es el componente encargado del acceso a datos, representando las estructuras de datos del sistema. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

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

**Controlador:** Responde a eventos y actúa como intermediario entre el modelo, la vista y cualquier otro recurso necesario para generar una página.

La utilización del patrón MVC facilita la realización de cambios en la aplicación puesto que cuando se realiza un cambio de bases de datos, programación o interfaz de usuario solo se maneja uno de los componentes. Además se puede modificar uno de los componentes sin conocer cómo funcionan los otros.

### **Arquitectura Cliente Servidor**

La arquitectura Cliente – Servidor (C/S) es una forma de dividir y especializar programas y equipos de cómputo de forma que la tarea que cada uno de ellos realice se efectúe con la mayor eficiencia posible y permita simplificar las actualizaciones y mantenimiento del sistema. En este modelo, el papel del cliente es iniciar el diálogo, enviando peticiones al servidor conforme a algún protocolo asimétrico, pidiéndole que actúe, que le informe, o ambas cosas. El servidor es quien responde las solicitudes del cliente. (25)

Entre sus principales características se encuentran:

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

### **1.5 Tecnologías utilizadas en el proceso de desarrollo**

En este epígrafe se exponen las tecnologías utilizadas en el proceso de desarrollo, agrupadas según las capas del patrón Modelo-Vista-Controlador.

#### **Java Enterprise Edition 5 (JEE5)**

Es la plataforma que provee SunMicrosystem para dar soporte y desarrollar software para las empresas. El gran éxito de java como plataforma para el desarrollo de aplicaciones se encuentra en esta especificación, que no es más que un conjunto de librerías que establecen un estándar para lograr un producto altamente calificado. Con el lanzamiento de esta edición, la especificación dio un gran paso hacia la excelencia, pues incorporó nuevas tecnologías de punta que no estaban contempladas anteriormente y que indiscutiblemente le incorporan a la plataforma gran robustez y simplicidad a la hora de trabajar, suficientes razones para no dudar en establecerla como base para el desarrollo. (26)

#### **Java virtual machine**

La Máquina Virtual Java es el núcleo del lenguaje de programación Java. En ella se encuentra el motor que en realidad ejecuta el programa Java y es la clave de muchas de las características principales de

Java, como la portabilidad, la eficiencia y la seguridad. Se encarga de interpretar todo el código java y convertirlo al lenguaje nativo del sistema operativo en uso. (27) La utilización de la máquina virtual java permite la portabilidad de la aplicación.

### 1.5.1 Capa de presentación

#### XHTML

Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es solamente la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas de XML. La utilización de XHTML facilita en la aplicación la edición directa del código y de mantenimiento. (28)

#### JavaScript

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web. JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, entre otros.

JavaScript se usa principalmente 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 existe una forma de JavaScript del lado del servidor. Dado que el código JavaScript puede ejecutarse localmente en el navegador del usuario (en lugar de en un servidor remoto), el navegador puede responder a las acciones del usuario con rapidez, haciendo una aplicación más sensible. Por otra parte, el código JavaScript puede detectar acciones de los usuarios que HTML por sí sola no puede (como pulsaciones de teclado). (29)

#### Java Server Face (JSF)

Java Server Faces (JSF) es un framework Java que permite crear interfaces de usuario (UI) para una aplicación web, mediante el uso de componentes reutilizables. Permite el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web. (30)

### **Facelets**

Facelets es un framework simplificado de presentación, en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. (31)

### **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. Mediante 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. (32)

### **Richfaces**

Es una biblioteca de componentes JSF y un avanzado framework para la integración de Ajax con facilidades en la capacidad de desarrollo de aplicaciones de negocio. Se integra completamente dentro del ciclo de vida 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. Richfaces, además, es un proyecto que fue desarrollado con una arquitectura abierta para que fuera compatible con la mayor cantidad de entornos. (33)

### **Seam UI**

Serie de controles Seam altamente integrables con JBossSeam. Es utilizado para complementar los controles JSF incorporados y los controles de otras bibliotecas externas. (34)

## **1.5.2 Capa del negocio**

### **Seam**

Es un poderoso y moderno framework creado para unificar todas las tecnologías estándares JSF, EJB3, JPA. Es un software de código abierto que asocia diferentes tecnologías y estándares Java, adicionando algunas funcionalidades no contempladas en frameworks anteriores. Automatiza muchas de las tareas

comunes y hace un uso extensivo de las anotaciones para poder reducir la cantidad de código XML que es necesario escribir, reduce de manera importante la cantidad total de codificación necesaria. Permite la creación de complejas aplicaciones Web.

Integra además el concepto de workspaces (entornos de trabajo) permitiendo que el usuario tenga en varios tabs o ventanas del navegador actividades del negocio con contextos completamente aislados. Seam integra transparentemente la administración de procesos del negocio vía jBoss-jBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones e interacciones con el usuario. (35)

### 1.5.3 Capa acceso a datos

#### **Hibernate**

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 un software libre, distribuido bajo los términos de la licencia GNU LGPL. (36)

#### **Enterprise Java Bean 3 (EJB 3)**

Enterprise JavaBeans (EJB) es una arquitectura de componentes de servidor que simplifica el proceso de construcción de aplicaciones de componentes empresariales distribuidos en Java. Con su utilización es posible escribir aplicaciones escalables, fiables y seguras sin escribir código de infraestructura. La existencia de infraestructura permite un desarrollo más rápido de la parte servidora. Un EJB es un componente software que se ejecuta del lado del servidor en una aplicación multicapa. Los clientes del EJB acceden a él por medio de una interfaz que esconde los detalles de implementación del componente. Esta interfaz debe cumplir la especificación EJB. La especificación fuerza la presencia de unos determinados métodos que permitirán al contenedor de EJBs manejar los componentes y su ciclo de vida. (37)

### 1.6 Lenguaje de programación

#### Java

Java es una plataforma de software desarrollada por un grupo de ingenieros de SunMicrosystem en 1991. 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.

Java 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. Proporciona numerosas comprobaciones en compilación y tiempo de ejecución y elimina la necesidad de liberación explícita de memoria, convirtiéndolo en un lenguaje robusto.

Este lenguaje es utilizado de manera horizontal en el desarrollo del sistema, pues puede estar presente en las diferentes capas de la aplicación. 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. (38)

### 1.7 Metodologías de desarrollo

#### Proceso Unificado de Desarrollo (RUP)

RUP es una de las metodologías de desarrollo de software más utilizada en el mundo para el análisis, implementación y documentación de sistemas orientados a objetos. Durante todo el ciclo de desarrollo, se administran los requisitos, con el objetivo de mantener la trazabilidad de los requerimientos funcionales, siendo así el proceso de desarrollo guiado por casos de uso. Permite la reutilización de estos requerimientos para otros sistemas o funcionalidades del propio software. RUP es centrado en la

arquitectura, por lo que es fundamental identificar los casos de uso de mayor prioridad e importancia basándose en este criterio.

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 requerimientos. 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.

Dentro de sus disciplinas gestiona el control de cambios, que permite mantener al equipo trabajando en los mismos artefactos, en cualquier momento del desarrollo del producto. En su modelación RUP define como sus principales elementos a los trabajadores, sus responsabilidades, los artefactos y los flujos de actividades. Los trabajadores son los propietarios de elementos o artefactos y se encargan de realizar las actividades, las cuales describen cómo una tarea es realizada por un trabajador y a su vez, manipulan elementos. RUP realiza varias iteraciones en número variable en las que se hacen mayor o menor hincapié en las distintas actividades por lo que se han definido en nueve flujos o disciplinas. (39)

### Lenguaje Unificado de Desarrollo (UML)

UML es un lenguaje usado para especificar, visualizar, construir y documentar los componentes de un sistema orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está apoyado en gran manera por el OMG (Object Management Group).

Además ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Permite modelar sistemas utilizando técnicas orientadas a objetos, es independiente aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (40)

### 1.8 Servidor de Base de Datos

#### PostgreSQL 9.1

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostgreSQL 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:** El 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. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

**Multiplataforma:** PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas. Diseñado para ambientes de alto volumen. (41)

### 1.9 Herramientas

#### JBoss Developer Studio 5.0.0

La plataforma Eclipse consiste en un Entorno de Desarrollo Integrado (IDE, Integrated Development Environment) abierto y extensible. Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un IDE cuenta con un editor de código, un compilador/intérprete y un depurador. Eclipse sirve como IDE Java y cuenta con numerosas

herramientas de desarrollo de software. Es de gran utilidad porque da soporte a otros lenguajes de programación, como son C/C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plugins) para extender la funcionalidad.

Dentro de las características que presenta se destacan: poseer un editor visual con sintaxis coloreada, permitir la compilación incremental de código, modificar e inspeccionar valores de variables, avisar de los errores cometidos mediante una ventana secundaria y depurar el código que reside en una máquina remota. (42) Es soportado por los sistemas operativos:

- Linux
- Windows
- Solaris 8 (SPARC/GTK 2)
- Mac OSX –Mac/Carbon

### **JBoss Tools**

JBoss Tools 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. (43)

Los módulos de JBoss Tools son:

- RichFaces VE: El editor visual aportado por Exadel proporciona el apoyo para la edición visual de páginas HTML, JSF, JSP y Facelets. También incluye soporte visual para las librerías de componentes JSF incluyendo JBossRichFaces.
- 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, ejecución interactiva de HQL/JPA-QL/Criteria.
- JBoss AS Tools: Fácil de iniciar, detener y debuguear al estar integrado con Eclipse. También incluye funciones para el despliegue eficaz de cualquier tipo de proyecto en el IDE.
- Drools IDE: Editor de ficheros de reglas, debugueo e inspección de reglas.
- jBPM Tools: Edición del flujo de trabajo del jBPM, motor de procesos BPM.
- JBossWS Tools: Desarrollo, invocación, inspección y pruebas de webservices sobre http con la adición y soporte de características JBossWS.

### **PgAdmin III**

PgAdmin III es una aplicación para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como PervasivePostgres, EnterpriseDB, MammothReplicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. (44)

### **Visual Paradigm para UML 8.0**

Se utilizará el Visual Paradigm 6.0 como herramienta CASE (Computer Aided Software Engineering), esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Ayuda a una construcción de aplicaciones de calidad más rápida y a menor costo. Permite construir todo tipo de diagramas de clases, generar código desde diagramas y generar documentación. Apoya los estándares más altos de las notaciones de Java y de UML. Soporta aplicaciones web, es fácil de instalar y actualizar. Está diseñado para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientado a la creación de diseños y se usa el paradigma de programación orientada a objetos.

Visual Paradigm es una poderosa herramienta para visualizar y diseñar elementos de software, para ello utiliza UML (UML 2.1) y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos. Provee soporte para la generación de código, tiene integración con diversos IDE's como NetBeans (de SunMicrosystems), Developer (de Oracle), Eclipse (de IBM), JBuilder (de Borland), así como la posibilidad de realizarse la ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML e Hibernate. (45)

### **JBoss Server 4.2.2 GA**

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4 e incluye servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, lo que hace el desarrollo de las aplicaciones mucho más simple. Además, al ser desarrollado con tecnología Java, es multiplataforma. (46)

Sus principales características son:

- Producto de licencia de código abierto sin coste adicional.
- Confiable a nivel de empresa.
- Orientado a arquitectura de servicios.
- Flexibilidad consistente.
- Servicios del middleware para cualquier objeto de Java.
- Soporte completo para Java Management Extensions (JMX).

### **Conclusiones**

En este capítulo se realizó un estudio de sistemas que emitan diagnósticos médicos, llegando a la conclusión de que se tomarán como guía para el desarrollo de las funcionalidades asociadas al componente de emisión de diagnósticos, pues las soluciones que plantean no dan respuesta al problema identificado. Además se definió que el método de Series Temporales dará una mejor solución a la situación planteada. Por otra parte se expusieron las herramientas y tecnologías a utilizar en el desarrollo de las funcionalidades para cumplir con la arquitectura definida en el desarrollo de las mismas.

## Capítulo 2. Características del componente de emisión de diagnósticos

En este capítulo, con motivo de comprender 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 los vínculos que se establecen entre ellos. Se explica de manera detallada el funcionamiento del método de Series Temporales utilizado en la solución del problema. Además se definen los requerimientos funcionales y no funcionales a partir de los cuales se representan los casos de uso, con el objetivo de estructurar el Diagrama de Casos de Uso del componente.

### 2.1 Método estadístico utilizado en la solución del problema

En este epígrafe se definirá de manera concreta la forma en que se implementará el algoritmo de Series Temporales para el desarrollo de las funcionalidades asociadas al componente para el proceso de emisión de diagnósticos del Sistema de Información Hospitalaria del CESIM.

#### Serie Temporal

Una serie temporal es una secuencia de  $N$  observaciones recogidas en intervalos regulares de tiempo. Se utilizan en numerosas áreas para llevar el control de actividades como: las temperaturas horarias, los precios diarios de acciones al cierre de la bolsa, la tasa de desempleo mensual, la tasa de mortalidad infantil por año, la cantidad de ventas de un producto en el mercado, etc.

#### Clasificación de las series temporales

- **Forma de observación de la serie:**

**Discreta:** las observaciones se recogen solo en momentos determinados de tiempo, generalmente a intervalos iguales.

**Continua:** la serie se genera y se observa de forma continua, como, por ejemplo, la temperatura, que se observa de forma continua en el tiempo por medio de aparatos físicos. (47)

- **Tipo de predicción de la serie:**

**Determinista:** los futuros valores de la serie pueden predecirse con exactitud.

**Estocástica:** el futuro sólo se puede determinar de modo parcial por las observaciones pasadas, se considera que la distribución de probabilidad de los futuros valores está condicionada a los valores pasados. (48)

En la presente investigación se van a considerar únicamente las series discretas y estocásticas. Para el trabajo con una serie de tiempo, los datos u observaciones se suelen recoger a intervalos iguales de tiempo (días, meses, años); en el presente caso se trabajará con series mensuales. Para llevar a cabo el análisis de series temporales el primer paso es graficar la serie, ya que permite detectar las componentes esenciales de la misma y definir un polinomio conveniente para el ajuste.

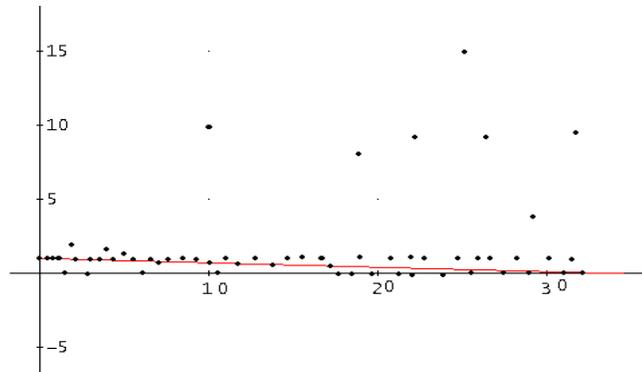


Figura 2.1 Representación de la Serie de Tiempo

Estas componentes son:

**Tendencia (T):** Son cambios continuos, tanto positivos como negativos, que ocurren en un período de tiempo determinado.

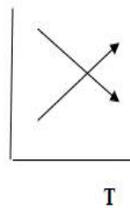


Figura 2.2 Representación de la Tendencia

**Estacionalidad (E):** Son cambios fuertes, positivos o negativos, que se presentan en ciertas épocas del año y se repiten en los años siguientes.

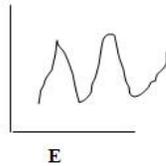


Figura 2.3 Representación de la Estacionalidad

**Ciclicidad (C):** Son cambios continuos que ocurren en períodos largos de tiempo, cuya principal característica es que se conoce el inicio pero no se conoce el final. Se repiten con el transcurrir de los años.

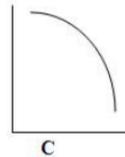


Figura 2.4 Representación la Ciclicidad

**Incertidumbre (I):** Son alteraciones súbitas o fuertes que se presentan por situaciones ajenas al negocio y que no se conoce si ocurrirá nuevamente.



Figura 2.5 Representación de la Incertidumbre

La serie obtenida es la unión de todos o algunos componentes presentes en la misma. Después de graficada la serie, se determinó que una recta suavizada pudiera ajustarse al modelo. Además por ser una

serie que recoge datos mensuales estará conformada por las componentes Tendencia, Estacionalidad e Irregularidad. La combinación de estos elementos deflacta (suaviza) el modelo de la serie.

Dos son los esquemas o modelos que generalmente son admitidos sobre la forma en que una serie temporal se descompone: (49)

Aditivo:  $X(t) = T(t) + E(t) + I(t)$

Donde:

Multiplicativo :  $X(t) = T(t) \times E(t) \times I(t)$

$X(t)$ : serie observada en instante t.

$T(t)$ : componente de tendencia.

$E(t)$ : componente estacional.

$I(t)$ : componente irregular (aleatoria).

Estos modelos son usados en dependencia del comportamiento que posea los componentes de la serie temporal. Por las características que posee el negocio (emisión de diagnósticos), el modelo a utilizar es el multiplicativo.

**Para la aplicación de esta técnica se identificaron dos pasos o momentos esenciales:**

- Extracción de la información de la base de datos.
- Aplicación de la técnica de series temporales para dar un pronóstico.

**Extracción de la información de la base de datos:**

El objetivo principal de este momento es garantizar que toda la información necesaria para el cálculo de la serie temporal sea recogida en una estructura conveniente. Debido a la complejidad del procedimiento y a la variedad de perspectivas para observar los datos se identificó que una matriz M era la solución. Esta matriz constituirá la salida fundamental de este momento.

Características de M:

- Sus filas representarán un tipo de enfermedad en el CIE10.

- Sus columnas representarán al tiempo  $t$ , donde  $t$  expresa la secuencia ascendente de meses a analizar.
- Presenta dimensión  $X \times Y$ , donde el valor de  $X$  está dado por la cantidad de enfermedades en el CIE10, y el valor de  $Y$ , por la cantidad de meses a analizar.
- En cada intersección  $(X_i, Y_i)$  de  $M$  se almacenará un listado, que recogerá, por cada especialidad, la cantidad de apariciones de la enfermedad  $X_i$  en el mes  $Y_i$ .

De esta manera se tendrá recogida detalladamente toda la información asociada al historial de diagnósticos del Sistema de Información Hospitalaria del CESIM. Además, se podrá hallar la cantidad de apariciones de una enfermedad en una especialidad, en un mes determinado.

#### **Aplicación de la técnica de series temporales para dar un pronóstico:**

El método inicia con una modificación a la llamada ecuación de la recta:  $y(t) = a + bt$ , donde  $y$  y  $t$  son las variables, siendo  $t$  la variable independiente (tiempo) y  $y$  la dependiente (cantidad de apariciones de una determinada enfermedad en una especialidad). Las incógnitas a buscar serían  $a$  y  $b$ , siendo  $b$  la pendiente de la recta, si es positiva será una recta creciente y si es negativa será una recta decreciente. Esta ecuación se puede escribir de la forma:  $T(t) = a + bt$  donde  $T$  es el valor de Tendencia. Para determinar el pronóstico de apariciones de una enfermedad en una determinada especialidad, es necesario multiplicar sus componentes: **T x E x I** o sea Tendencia x Estacionalidad x Incertidumbre.

- **Método para estimar Tendencia  $T(t)$**

Partiendo de los períodos de tiempo identificados (para la presente investigación se trabajarán con períodos mensuales), se comienza a calcular la ecuación necesaria mediante el sistema de **mínimos cuadrados (MCO)**. Esta es una técnica de análisis numérico encuadrada dentro de la optimización matemática, en la que, dados un conjunto de pares ordenados: variable independiente, variable dependiente, y una familia de funciones, se intenta encontrar la función, dentro de dicha familia, que se aproxime a los datos (en ese caso una recta). Este método presenta muchas ventajas en cuanto a la facilidad de su uso y por lo adecuado del planteamiento estadístico matemático, que permite adecuarse a los supuestos para los modelos econométricos.

Para hallar las componentes de la recta a través del método de mínimos cuadrados se aplican las siguientes fórmulas.

$$a = \frac{\sum e}{n} - (b) \frac{\sum t}{n} \quad b = \frac{\frac{\sum et}{n} - \frac{\sum e}{n} * \frac{\sum t}{n}}{\frac{\sum t^2}{n} - \left(\frac{\sum t}{n}\right)^2}$$

donde:

$e$ : cantidad de apariciones de una enfermedad

$t$ : número de los períodos (1, 2, 3, ...,  $t$ )

$n$ : cantidad de períodos

Luego de calcular las componentes de  $a$  y  $b$ , se sustituyen en la ecuación de la recta ( $T(t) = a + bt$ ) para así poder hallar el valor de tendencia de una enfermedad en cada período.

- **Método para estimar la Estacionalidad  $E(t)$**

Para el cálculo de la estacionalidad se utiliza un método de Promedio Móvil base 3 (PM3) y consiste en sacar un promedio de los primeros tres valores de apariciones en sus correspondientes períodos (1, 2, 3), luego se hace lo mismo con el segundo, tercer y cuarto período, y así sucesivamente. El resultado de la primera suma de los valores de los 3 períodos iniciales quedará ubicado en el segundo período, el cual corresponde a la mitad visual de 3 cifras. De esta manera se va determinando el PM3 de cada período.

Este valor del promedio móvil PM3, se llamará **TC** (se tiene en cuenta la Ciclicidad (C)), pues de los cuatro conceptos: T, E, C e I, hemos quitado la E y la I que eran los que tenían irregularidades. A continuación se procede a dividir las apariciones por período (que tienen los 4 conceptos incluidos o sea apariciones=TECI) entre **TC**, este resultado **TECI/TC** (se tiene en cuenta la Incertidumbre (I)) se llamará **EI**. Ya calculado cada EI, se procede a ubicarlo en relación con el período que le corresponde.

En este punto es necesario aclarar que la serie temporal se aplicará para definir el pronóstico de ocurrencia de una enfermedad en una especialidad para el mes  $Y_n$ , tomando en cuenta valores pasados

en períodos de un mes. Teniendo en cuenta este aspecto y una vez asociado cada valor de **EI** con cada uno de los períodos, se procede a sumar los valores de cada **EI** por período y se divide entre la cantidad de sumandos. Este valor de suma hallado es el promedio aritmético. La suma de todos los promedios aritméticos debe tender a un valor numérico igual a la cantidad de períodos (en este caso 12). Si no lo es, se aplica regla de tres para determinar el valor correcto (**X1**) en cada período, teniendo en cuenta que: “12 es a la suma aritmética en cada período como **X1** es a cada valor del promedio aritmético en cada período”. De esta manera se tendrá un valor **X1** para cada período. Este valor **X1**, es precisamente la Estacionalidad (E).

Para hallar **TxE** solo se tendrá que multiplicar el valor de Tendencia (de cada día) y el de Estacionalidad del período correspondiente. Este valor **TxE** se considera el valor del pronóstico de la cantidad de apariciones de la enfermedad en estudio. El cálculo anteriormente descrito es para determinar el pronóstico de ocurrencia de una enfermedad en una determinada especialidad. Análogamente, se aplica el mismo procedimiento para determinar el pronóstico de ocurrencia de cada enfermedad en todas las especialidades.

Para lograr un mejor entendimiento del procedimiento a efectuar para predecir la cantidad de apariciones de una enfermedad, se realiza un ejemplo ilustrativo, (Ver **Anexo 1**).

## 2.2 Propuesta del componente de emisión de diagnósticos

Como resultado de la investigación desarrollada y con el objetivo de agregarle nuevas prestaciones al componente de emisión de diagnósticos del Sistema de Información Hospitalaria del CESIM, se propone la creación de funcionalidades que faciliten la emisión de diagnósticos médicos en cualquiera de los módulos de atención: Consulta Externa, Salud Ocupacional, Emergencia y Hospitalización.

Estas funcionalidades permitirán:

- Calcular mediante el método Series Temporales el pronóstico de apariciones de una enfermedad en el tiempo.
- Agrupar las enfermedades según la especialidad médica por la cual un doctor se encuentra atendiendo a un paciente.

- Ordenar el listado de las enfermedades de forma descendente por el pronóstico de apariciones que posean.
- Mostrar el listado de las enfermedades.

La solución desarrollada posibilitará obtener una mayor organización del componente, a través de la agrupación de enfermedades por especialidad, que permitirá reducir a los especialistas el tiempo de obtención de los diagnósticos deseados. Además de facilitar la búsqueda y ordenamiento de las enfermedades mostradas en el componente.

### 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 del sistema, quien ejecuta esta tarea con ayuda de la visión del sistema y del glosario de requisitos. (50)

#### Conceptos fundamentales del dominio

Con el objetivo de lograr una mejor comprensión del Diagrama de Modelo de Dominio se dará una breve descripción de los conceptos asociados al problema a tratar.

**Hoja general de consulta externa:** Hoja general de consulta del Sistema de Información Hospitalaria del CESIM que recoge los datos comunes de cualquier consulta no especializada.

**Hoja traumatología y ortopedia:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta de Traumatología y Ortopedia.

**Hoja ginecológica:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta de Ginecología.

**Hoja oftalmológica:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta de oftalmología.

**Hoja otorrinolaringología:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta de Otorrinolaringología.

**Hoja uro-genital:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta Uro-Genital.

**Hoja obstétrica:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta de obstetricia.

**Hoja gastroenterológica:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta de gastroenterología.

**Hoja cirugía colo-rectal:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta de cirugía colo-rectal.

**Hoja preanestésica:** Hoja de consulta especializada del Sistema de Información Hospitalaria del CESIM que permite recoger los datos correspondientes a la consulta Preanestésica.

**Diagnóstico:** El diagnóstico es el resultado del análisis que se realiza en una primera instancia y que tiene como fin permitir conocer las características específicas de la situación determinada para así poder actuar en consecuencia, sugiriendo tratamiento o no. El diagnóstico puede aplicarse para ratificar o rectificar la presencia de una enfermedad, como también para conocer su evolución en el caso de confirmarse la misma.

**CIE10:** La Clasificación Internacional de Enfermedades (CIE 10), elaborado por la Organización Mundial de la Salud, tiene como propósito permitir el registro sistemático, análisis, interpretación, y comparación de los datos de mortalidad y morbilidad recolectados en diferentes países o áreas, y en diferentes épocas. Se utiliza para convertir los términos diagnósticos y otros problemas de salud, de palabras a códigos alfanuméricos que permiten su fácil almacenamiento y posterior recuperación para el análisis de la información.

**Médico especialista:** Es aquel cuya especialidad permite la creación de hojas especializadas en el Sistema de Información Hospitalaria del CESIM.

**Médico general:** Es aquel cuya especialidad permite la creación de la hoja general de consulta en el Sistema de Información Hospitalaria del CESIM.

**Componente de emisión de diagnóstico:** Componente del sistema de Información Hospitalaria del CESIM que permite la selección de diagnósticos asociados a pacientes en cualquiera de los módulos de atención.

### 2.3.1 Diagrama de modelo de dominio

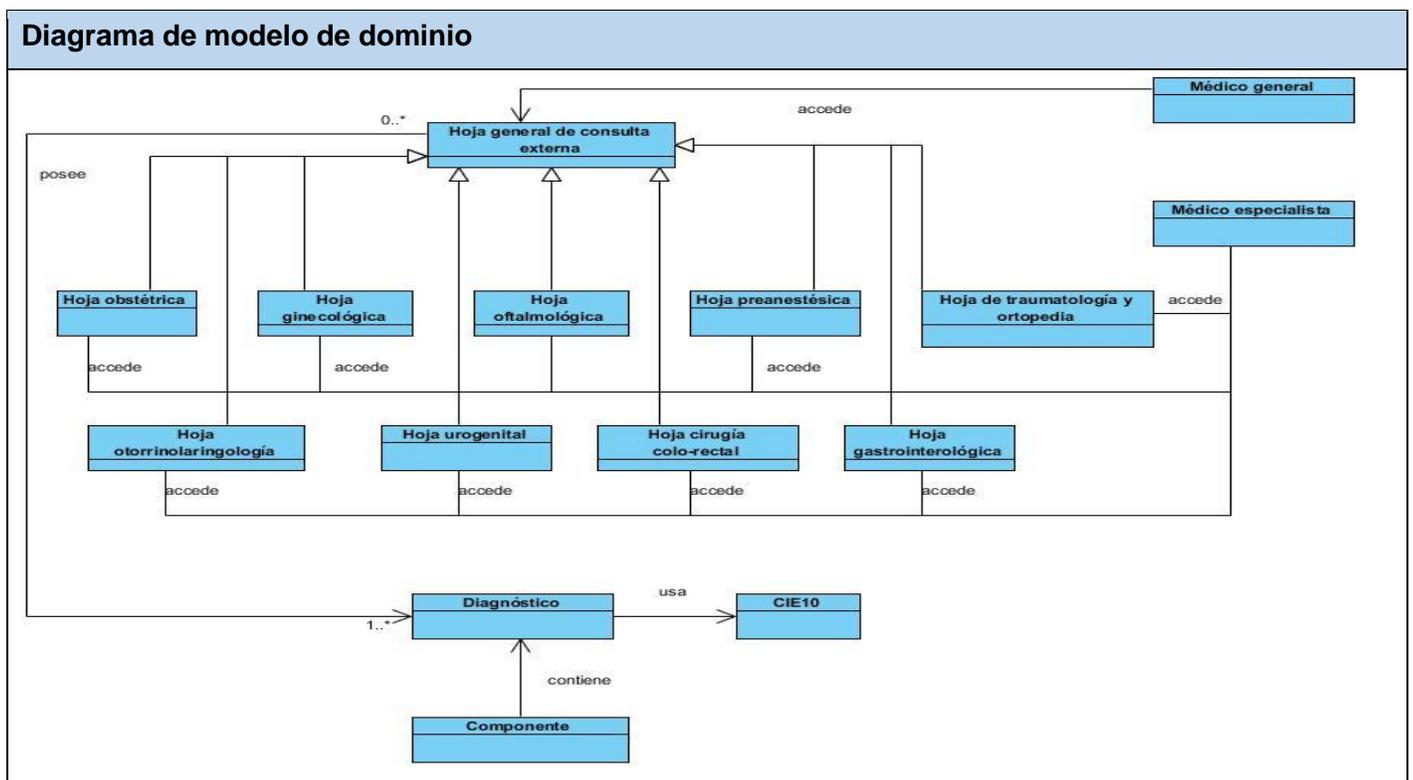


Figura 2.6 Diagrama de Modelo de dominio

## 2.4 Especificación de los requerimientos de software.

### 2.4.1 Requerimientos funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de

entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas, se pueden definir los siguientes requisitos funcionales. (51)

RF1: Calcular pronóstico de apariciones.

RF2: Ordenar diagnósticos según el pronóstico para cada especialidad.

RF3: Listar enfermedades.

RF4: Seleccionar enfermedad.

### **2.4.2 Requerimientos no funcionales**

Una vez identificado lo que el componente debe hacer, se determina cómo debe comportarse, qué características debe tener o cuán ágil debe ser. Los requisitos no funcionales se centran en responder a las necesidades que deben satisfacerse para que el componente pueda dar respuesta a las especificaciones del cliente.

En la práctica, los requerimientos no funcionales son primordiales para el éxito de cualquier sistema, por lo que es importante identificarlos, al menos en términos generales, para detallar las características del producto en cuanto a: usabilidad, rendimiento y portabilidad, que junto a las funcionalidades esperadas del software ayudarán a marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

#### **Usabilidad**

El componente deberá estar diseñado de manera que brinde un rápido y fácil acceso a todos los usuarios que cuenten con permisos suficientes para acceder a él. Podrá ser utilizado por usuarios con pocos conocimientos informáticos, por lo que debe presentar una interfaz intuitiva y de fácil asimilación. Además deberá facilitar la búsqueda de enfermedades.

#### **Seguridad**

Para acceder al componente se implementarán mecanismos de seguridad y control a nivel de usuario, que garanticen el acceso a la información a los usuarios permitidos. Se registrarán todas las acciones que se realizan por cada usuario en todo momento.

### **Rendimiento**

El trabajo con el componente minimizará el volumen de datos en las peticiones y optimizar el uso de la memoria RAM, persistiendo en la base de datos los pronósticos de aparición de las enfermedades que se hayan modificado. Se respetarán las buenas prácticas de programación para incrementar el rendimiento en las operaciones complejas.

### **Soporte**

Los requerimientos de soporte abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software, con el objetivo de asistir a los clientes de dicho software. Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, así como asignación de perfiles. Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.

Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema. Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados. Se permitirá administración remota, monitoreo del funcionamiento del componente en los centros hospitalarios y detección de fallas de comunicación.

### **Hardware**

#### *Estaciones de trabajo*

El componente debe poder utilizarse en clientes que posean como mínimo memoria RAM de 1GB y microprocesador Intel® Core-2 Duo o Intel® Dual-Core.

#### *Servidores*

Los servidores, por su parte, deberán 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 base de datos: PowerEdge R910, Procesador Intel® Xeon® CPU E7- 8837 @ 2.67 GHz, 16 GB de memoria RAM, 1199 GB de disco duro.

- Servidores de aplicaciones: PowerEdge R910, Procesador Intel® Xeon® CPU E7- 8837 @ 2.67 GHz, 16 GB de memoria RAM, 1199 GB de disco duro.

### **Software**

El componente formará parte del Sistema de Información Hospitalaria del CESIM, por lo que debe poder desplegarse en los sistemas operativos Windows 7 o superior y Linux, utilizando Java Runtime Environment, JBoss Server y PostgreSQL. El usuario deberá disponer de un navegador web, el cual puede ser Google Chrome, Internet Explorer 7.0, Firefox 4.0 o versiones superiores de estos.

### **Portabilidad**

La aplicación podrá ser desplegada sobre los Sistemas Operativos Windows 7 o Linux.

### **Interfaz**

La interfaz dispondrá de teclas de función y menús desplegables que faciliten y aceleren su utilización. Las ventanas del componente contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en el idioma previamente configurado.

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

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.

### **Elementos básicos:**

- Actores: Los actores representan un tipo de usuario del sistema. Se entiende como usuario cualquier cosa externa que interactúa con el sistema. No tiene por qué ser un ser humano, puede ser otro sistema informático o unidades organizativas o empresas.
- Caso de uso: Es una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando. Se representan mediante un óvulo. Cada caso de uso debe detallarse, habitualmente mediante una descripción textual

- Asociaciones: Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso. (52)

## 2.5.1

## Definición de los actores del sistema

Actor	Objetivos
Médico de consulta externa	Se encarga de agrupar el comportamiento común de los actores: <ul style="list-style-type: none"> <li>• Gestor de la Hoja general</li> <li>• Gestor de la Hoja de ginecología</li> <li>• Gestor de la Hoja obstétrica</li> <li>• Gestor de la Hoja de traumatología y ortopedia</li> <li>• Gestor de la Hoja neurológica</li> <li>• Gestor de la Hoja gastroenterológica</li> <li>• Gestor de la Hoja oftalmológica</li> <li>• Gestor de la Hoja de ORL</li> <li>• Gestor de la Hoja uro-genital</li> <li>• Gestor de la Hoja preanestésica</li> <li>• Gestor de la Hoja de cirugía colo-rectal</li> </ul>
Gestor de la Hoja general	Es el encargado de crear la Hoja general de consulta externa.
Gestor de la Hoja de ginecología	Es el encargado de crear la Hoja ginecológica.
Gestor de la Hoja obstétrica	Es el encargado de crear la Hoja obstétrica.
Gestor de la Hoja de traumatología y ortopedia	Es el encargado de crear la Hoja de traumatología y ortopedia.
Gestor de la Hoja neurológica	Es el encargado de crear la Hoja neurológica.
Gestor de la Hoja	Es el encargado de crear la Hoja gastroenterológica.

gastroenterológica	
Gestor de la Hoja oftalmológica	Es el encargado de crear la Hoja oftalmológica.
Gestor de la Hoja de ORL	Es el encargado de crear la Hoja de ORL.
Gestor de la Hoja uro-genital	Es el encargado de crear la Hoja uro-genital.
Gestor de la Hoja preanestésica	Es el encargado de crear la Hoja preanestésica.
Gestor de la Hoja de cirugía colo-rectal	Es el encargado de crear la Hoja de cirugía colo-rectal.
Técnico (a) de registros y estadísticas de salud de consulta externa	Es el encargado de planificar los horarios de los médicos y generar los reportes estadísticos de consulta externa.
Reloj del sistema	Es el encargado de a ejecutar la tarea programada a una hora definida.

Tabla 2.1 Descripción de los actores del sistema

2.5.2 Vista global de actores del componente

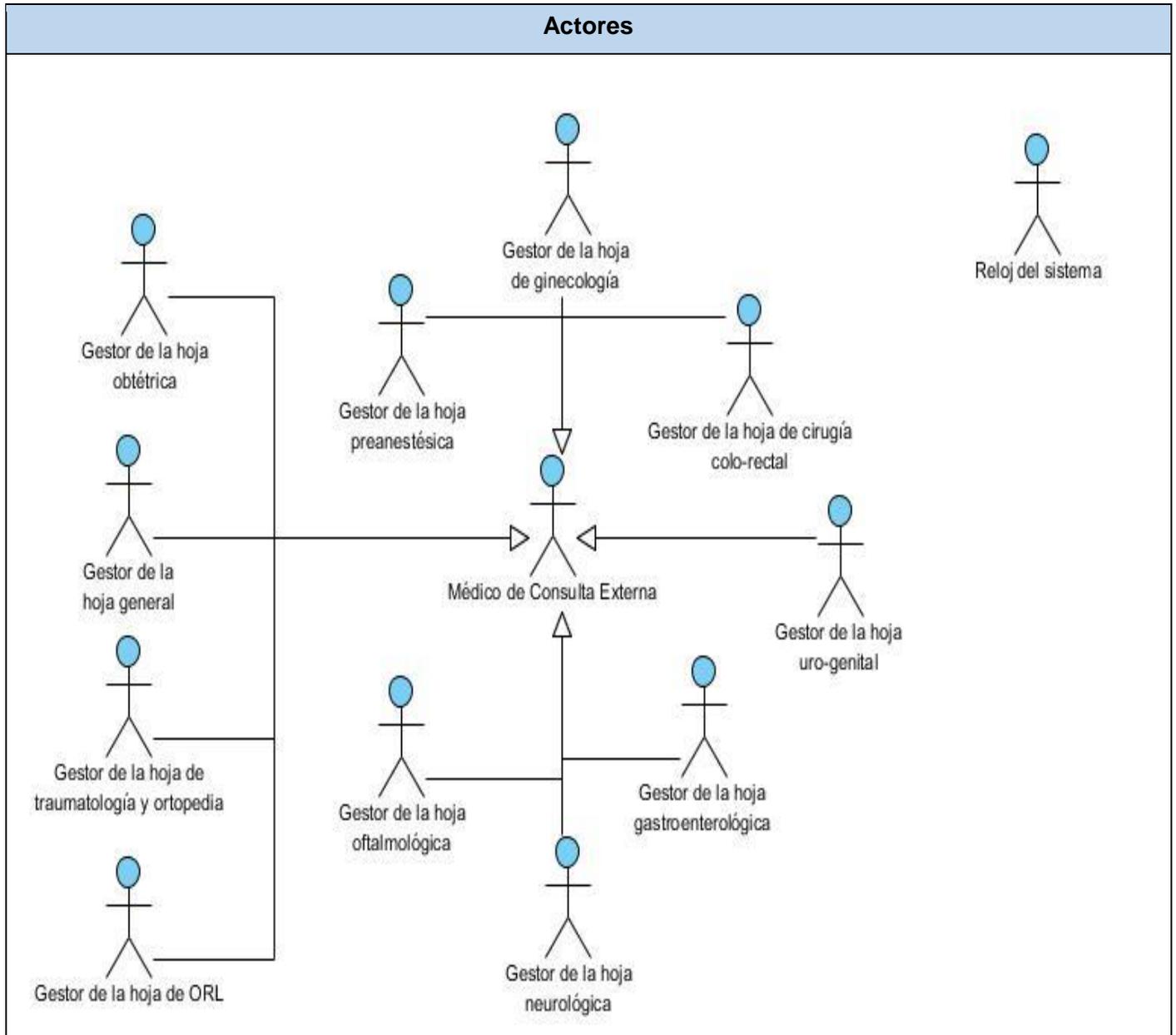


Figura 2.7 Diagrama de actores del componente

2.5.3 Diagrama de Casos de Uso

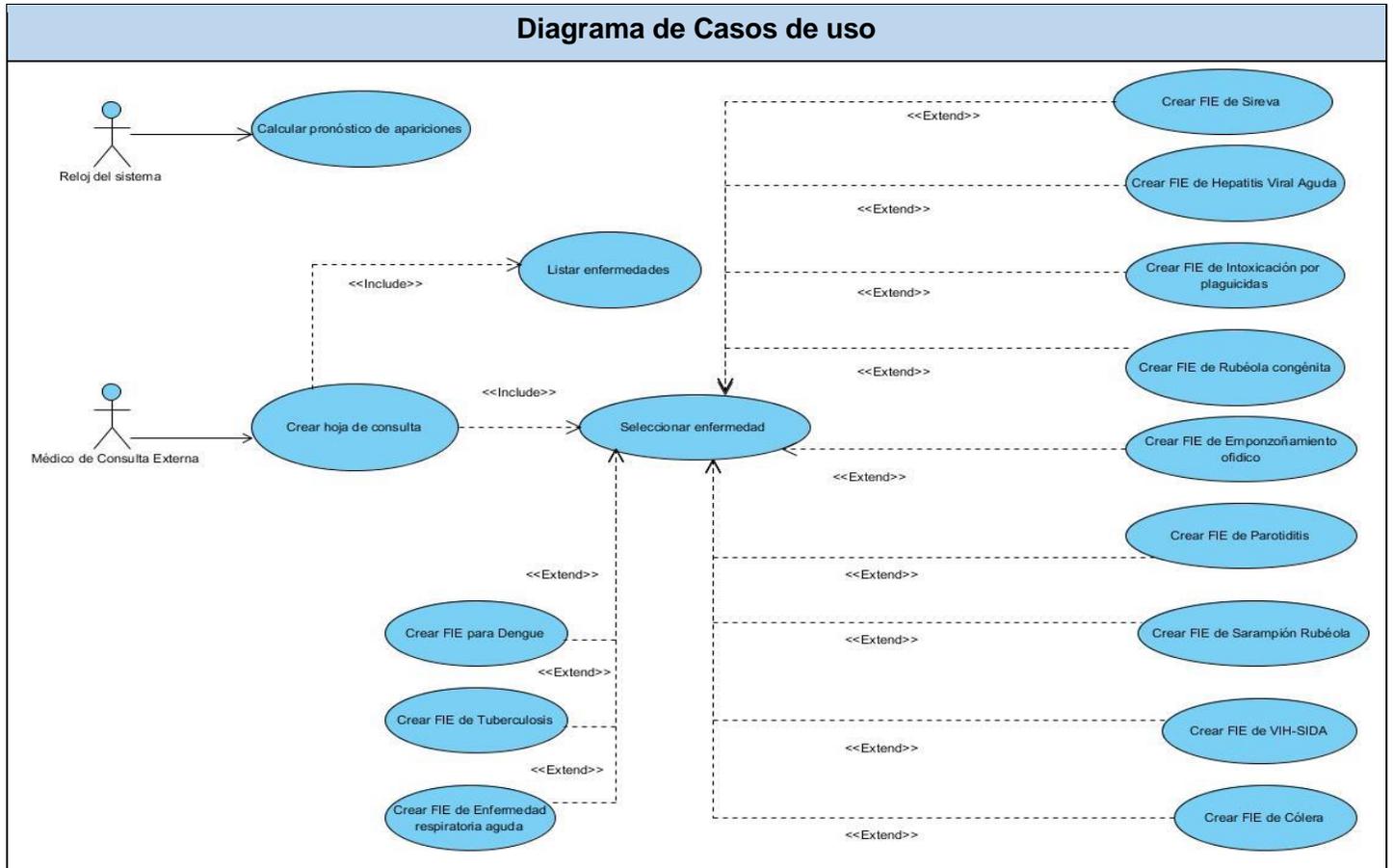


Figura 2.8 Diagrama de Casos de Uso

2.5.4 Especificación de los casos de uso

CU1: Calcular pronóstico de apariciones

Objetivo	Determinar un valor numérico que exprese el pronóstico de la cantidad de apariciones de una enfermedad.
Actores	Médico de consulta externa
Resumen	El sistema ejecuta la tarea programada a una hora especificada, esta ejecución permite calcular el pronóstico de aparición para cada enfermedad en cada especialidad registrada en el sistema. Luego se guarda dicho pronóstico en la base de

	datos y el caso de uso termina.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Se haya ejecutado la tarea programada.	
Poscondiciones	Se calculó el pronóstico de apariciones de cada diagnóstico.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Se ejecuta la tarea programada a la hora especificada.	
2.		Calcula el pronóstico de aparición para cada enfermedad en cada especialidad.
3.		Guarda el pronóstico de apariciones de cada enfermedad en la base de datos.
4.		Termina el caso de uso.
Relaciones	CU Incluidos	No existen
	CU Extendidos	No existen

Tabla 2.2 Descripción del caso de uso: "Calcular pronóstico de apariciones."

**CU2: Listar enfermedades**

Objetivo	Mostrar el listado de las enfermedades ordenas descendentemente.
Actores	Médico de consulta externa
Resumen	El caso de uso inicia cuando el médico de consulta externa crea la hoja de consulta, luego se carga en el componente de diagnóstico el listado de las enfermedades almacena en la base de datos, ordenándolas descendentemente por el pronóstico de aparición que posean, se muestra el listado de las enfermedades ordenadas y permite

	la selección de las mismas, el caso de uso termina.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Se halla calculado el pronóstico de apariciones de las enfermedades.	
Poscondiciones	Se listaron las enfermedades de forma ordenada.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Crea la hoja de consulta. <b>Ver CU:</b> "Crear hoja de consulta."	
2.		Carga el listado de las enfermedades de la base de datos.
3.		Muestra el listado de enfermedades, ordenado de forma descendente según el pronóstico de ocurrencia y con los siguientes atributos: <ul style="list-style-type: none"> <li>• Código</li> <li>• Descripción</li> <li>• Estructura</li> </ul>
4.		Permite: <ul style="list-style-type: none"> <li>• Seleccionar una o varias enfermedades. <b>Ver CU:</b> "Seleccionar enfermedad."</li> <li>• Mostrar el resultado ordenado de forma descendente, por pronóstico de ocurrencia de cada enfermedad, teniendo en cuenta la especialidad del médico.</li> </ul>
5.		Termina el caso de uso.
Relaciones	CU Incluidos	No existen

	CU Extendidos	No existen
--	---------------	------------

Tabla 2.3 Descripción del caso de uso: “Listar enfermedades.”

**CU3: Seleccionar enfermedad.**

Objetivo	Seleccionar enfermedad	
Actores	Médico de consulta externa	
Resumen	El caso de uso inicia cuando el actor necesita seleccionar o modificar una enfermedad, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar enfermedades, el actor introduce los datos que considera como criterios para realizar la búsqueda, el sistema busca y muestra las enfermedades que cumplen con los criterios de búsqueda, el actor selecciona o descarta las enfermedades, el caso de uso termina.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Se debe haber iniciado una consulta.	
Poscondiciones	Se seleccionaron enfermedades.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	El caso de uso inicia cuando el actor necesita seleccionar o modificar una enfermedad.	
2.		Verifica la existencia de enfermedades registradas anteriormente. Si existen enfermedades registradas anteriormente. <b>Ver Alternativa 1:</b> “Mostrar enfermedades registradas anteriormente.”
3.		Brinda la posibilidad de buscar enfermedades por los criterios elementales de búsqueda: <ul style="list-style-type: none"> <li>• Código</li> </ul>

		<ul style="list-style-type: none"> <li>• Descripción</li> <li>• Estructura</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Buscar enfermedades dado criterios.</li> <li>• Realizar una búsqueda avanzada con más criterios. <b>Ver Alternativa 2:</b> “Realizar una búsqueda avanzada.”</li> </ul>
4.	Introduce los datos que considera como criterios para realizar la búsqueda y selecciona la opción de Buscar enfermedades dado criterios.	
5.		Busca las enfermedades que cumplen con los criterios de búsqueda.
6.		Si no se encuentra ninguna enfermedad que cumpla con los criterios de búsqueda. <b>Ver Alternativa 3:</b> “No se encuentra información que cumpla con los criterios de búsqueda.”
7.		<p>Muestra el listado de enfermedades, ordenado de forma descendente según el pronóstico de ocurrencia, que cumplan con los criterios de búsqueda seleccionados y muestra los siguientes atributos:</p> <ul style="list-style-type: none"> <li>• Código</li> <li>• Descripción</li> <li>• Estructura</li> </ul>
8.		<p>Permite:</p> <ul style="list-style-type: none"> <li>• Seleccionar una o varias enfermedades.</li> <li>• Mostrar el resultado ordenado por pronóstico</li> </ul>

		de ocurrencia teniendo en cuenta la especialidad del médico. <b>Ver Alternativa 4:</b> “Seleccionar especialidad”
9.	Selecciona una o varias enfermedades.	
		Muestra de las enfermedades seleccionadas los datos: <ul style="list-style-type: none"> <li>• Código</li> <li>• Descripción</li> </ul> Y permite: Eliminar enfermedades seleccionadas. <b>Ver Alternativa 6:</b> “Eliminar enfermedades seleccionadas.”
10.		Si una enfermedad seleccionada es de notificación obligatoria. <b>Ver Alternativa 5:</b> “Crear ficha epidemiológica”
11.		El caso de uso termina.
Flujos alternos		
Alternativa 1: “Mostrar enfermedades registradas anteriormente.”		
	Actor	Sistema
1.		Muestra de las enfermedades registradas anteriormente los datos: <ul style="list-style-type: none"> <li>• Código</li> <li>• Descripción</li> </ul> Y permite: <ul style="list-style-type: none"> <li>• Eliminar enfermedades registradas anteriormente. <b>Alternativa 6:</b> “Eliminar enfermedad registrada anteriormente”.</li> <li>• Seleccionar una nueva enfermedad. Regresa</li> </ul>

		al paso 3 del <b>Flujo Normal de Eventos</b> .
Flujos alternos		
Alternativa 2: “Realizar una búsqueda avanzada.”		
	Actor	Sistema
1.	Selecciona la opción de realizar una búsqueda avanzada.	
2.		Brinda la posibilidad de introducir los criterios de búsqueda avanzada: <ul style="list-style-type: none"> <li>• Capítulo</li> <li>• Grupo</li> <li>• Categoría</li> <li>• Subcategoría</li> <li>• Código</li> <li>• Descripción</li> </ul>
3.		Regresa al paso 3 del <b>Flujo Normal de Eventos</b> .
Flujos alternos		
Alternativa 3: “No se encuentra información que cumpla con los criterios de búsqueda”.		
	Actor	Sistema
1.		Muestra el mensaje de información “No se encontró información que cumpla con los criterios de búsqueda”.
2.		Regresa al paso 3 del <b>Flujo Normal de Eventos</b> .
Flujos alternos		
Alternativa 4: “Seleccionar especialidad.”		
	Actor	Sistema
1.	Selecciona la especialidad.	
2.		Muestra el listado de enfermedades ordenado por el pronóstico de aparición según la especialidad seleccionada.

3.		Regresa al paso 8 del <b>Flujo Normal de Eventos</b> .
Flujos alternos		
Alternativa 5: "Crear ficha epidemiológica".		
	Actor	Sistema
1.		<p>En dependencia de la enfermedad seleccionada permite Crear Ficha Epidemiológica. Ver casos de uso:</p> <ul style="list-style-type: none"> <li>• Epidemiología:: Crear FIE para Dengue</li> <li>• Epidemiología:: Crear FIE de Hepatitis Viral Aguda</li> <li>• Epidemiología:: Crear FIE de Intoxicación por plaguicidas</li> <li>• Epidemiología:: Crear FIE de Rubéola congénita</li> <li>• Epidemiología:: Crear FIE de Emponzoñamiento ofídico</li> <li>• Epidemiología:: Crear FIE de Parotiditis</li> <li>• Epidemiología:: Crear FIE de enfermedad respiratoria aguda</li> <li>• Epidemiología:: Crear FIE de Sireva</li> <li>• Epidemiología:: Crear FIE de Sarampión Rubéola</li> <li>• Epidemiología:: Crear FIE de VIH-SIDA</li> <li>• Epidemiología:: Crear FIE de Cólera</li> <li>• Epidemiología:: Crear FIE de Tuberculosis</li> </ul>
3.		Regresa al paso 8 del <b>Flujo Normal de Eventos</b> .
Flujos alternos		
Alternativa 6: "Eliminar enfermedades seleccionadas".		

Actor	Sistema	
1. Selecciona la opción de eliminar enfermedad.		
2.	Oculto la enfermedad selecciona para eliminar.	
3.	Regresa al paso 8 del <b>Flujo Normal de Eventos</b> .	
Relaciones	CU Incluidos	No existen
	CU Extendidos	Epidemiología:: Crear FIE para Dengue Epidemiología:: Crear FIE de Hepatitis Viral Aguda Epidemiología:: Crear FIE de Intoxicación por plaguicidas Epidemiología:: Crear FIE de Rubéola congénita Epidemiología:: Crear FIE de Emponzoñamiento ofídico Epidemiología:: Crear FIE de Parotiditis Epidemiología:: Crear FIE de enfermedad respiratoria aguda Epidemiología:: Crear FIE de Sireva Epidemiología:: Crear FIE de Sarampión Rubéola Epidemiología:: Crear FIE de VIH-SIDA Epidemiología:: Crear FIE de Cólera Epidemiología:: Crear FIE de Tuberculosis

Tabla 2.4 Descripción del caso de uso: “Seleccionar enfermedad.”

**CU4: Crear hoja de consulta. (Ver Artefacto: Especificación de caso de uso)**

**Conclusiones**

Como resultado de del análisis realizado en este capítulo se define el modelo de dominio, donde se representa la interacción entre los conceptos encontrados y las relaciones definidas entre ellos, lo que permite un mejor entendimiento del problema y una rápida identificación de las funcionalidades a desarrollar. Se definió el Modelo de Caso de Uso del sistema con su diagrama y especificaciones correspondientes, con el fin de crear las funcionalidades que respondan de forma satisfactoria a la problemática planteada.

## Capítulo 3. Análisis y Diseño del componente de emisión de diagnósticos

En este capítulo se describen la concepción arquitectónica y los patrones de diseño definidos para el desarrollo de funcionalidades asociada al componente de emisión de diagnósticos. Se realiza el diagrama de clases de diseño y se brinda una descripción de las clases identificadas para su posterior implementación. Además se exponen las estrategias de integración y reutilización empleadas en el desarrollo del componente de emisión de diagnósticos.

### 3.1 Descripción de la arquitectura. Fundamentación

La arquitectura de software es un conjunto de patrones y abstracciones que proporcionan el marco de referencia necesario para guiar la construcción del sistema. Permite a los programadores, diseñadores y analistas trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo deseado. La arquitectura de un sistema, ofrece de forma conceptual, la descripción del software en subsistemas y componentes, dejando bien claro la forma en que estos interactúan.

Para la creación de las funcionalidades asociadas al componente para el proceso de emisión de diagnósticos se hará uso del patrón arquitectónico Modelo Vista Controlador (MVC), muy usado en aplicaciones web (expuesto en el capítulo 1). Este patrón permite la separación de los datos de una aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos: el modelo, para la administración de los datos; 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 desacople la vista del modelo y permita la reusabilidad de los componentes. Brinda 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 JBossSeam. Se utilizan también componentes Seam de interfaz de usuario y Facelets como motor de plantillas. El uso de estos componentes enriquece el diseño de la interfaz de usuario.

La capa de controlador o capa de negocio es la encargada de manipular y procesar la información. Estará compuesto por clases Java controladoras, que definen la lógica del negocio, a las cuales, mediante anotaciones que provee el framework Seam, se le especifica el contexto en el que se encuentran (evento, página, conversacional, etc.), los cuales definen el estado de las entidades y datos 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 también 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, esto permite llevar las consultas a un lenguaje de objetos. (53)

La comunicación que se establece entre los elementos de estas capas se encuentra regida por el framework Seam, el cual permite mediante anotaciones, que las páginas de la interfaz de usuario referencien las funcionalidades definidas en las clases controladoras, y que estas puedan usar los componentes de acceso a datos y otros de la capa del negocio.

### 3.2 Estrategia de integración y reutilización del sistema

El componente de emisión de diagnósticos no responde a un área específica de las instituciones hospitalarias, sino que puede estar ubicado en cualquier lugar donde se emita un diagnóstico, tal es el caso de los módulos Hospitalización, Consulta Externa, Salud Ocupacional o Emergencia. Para lograr una mejor descripción, entendimiento y evaluación del funcionamiento del mismo se determinó en la presente investigación, integrar las funcionalidades asociadas al componente de emisión de diagnósticos en el módulo Consulta Externa, debido a que este módulo permite identificar con facilidad la especialidad del médico que está realizando la consulta. Además de los módulos de atención, es el más usado en clínicas y hospitales para registrar la información relacionada con la atención al paciente, por lo que sería el más eficiente para comprobar el funcionamiento del componente luego de agregarle las funcionalidades realizadas. La aplicación de estas funcionalidades se extenderá posteriormente a los restantes módulos de atención.

La integración de las clases utilizadas en la elaboración de las funcionalidades asociadas al componente de diagnósticos se realizó de la siguiente manera: la clase controladora `CalculoPronosticoScheduled` es la clase encargada de realizar el cálculo del pronóstico de apariciones de una enfermedad, utiliza las clases `RegistroAparicion`, `EnfermedadCalculada_diagnostico` y `ReporteWrapper` las cuales se encuentran almacenadas en el paquete **entity**. Por otra parte en el paquete **entity** se almacenan las clases mapeadas

de la base de datos, las que son utilizadas en la recolección de la información empleada en la realización de la predicción. La clase controladora `EnfermedadCalculadaConsList_custom` es inyectada en la clase `CrearHojaConsultaControlador` con el objetivo de retornar las listas de enfermedades teniendo en cuenta los parámetros recibidos de la interfaz. Además de implementar los métodos de búsqueda de la lista a mostrar.

### 3.3 Modelo de diseño

Un Modelo de Diseño es un artefacto que describe la realización de los casos de uso, centrándose gráficamente en la lógica de los requerimientos no funcionales y funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. (54) En el diseño se modela el sistema y encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los funcionales y las restricciones que se le suponen. Es un modelo de objetos que describe la realización física de los casos de uso centrándose en los requisitos funcionales y no funcionales.

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. Son una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Facilitan la reutilización del conocimiento experto como componentes de diseño y mejoran así la documentación, comprensión, comunicación del diseño final y el aprendizaje al programador inexperto, permitiendo que el software construido sea más fácil de mantener y extender. Además identifican clases, instancias, roles, colaboraciones y la distribución de responsabilidades. De manera general, los patrones constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, basados en la experiencia y que se ha demostrado que funcionan. 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), los cuales se utilizan con el objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño. Cumpliendo el prototipo del diseño a cada clase se le fueron asignadas las tareas que podrían realizar según la información que poseía, además se conserva el encapsulamiento ya que los objetos se valen de su propia información para hacer lo que se les pide, estas características responden al patrón Experto. Por otra parte el patrón Creador guía la asignación de responsabilidades relacionadas con

la creación de objetos. El diseño obtenido cumple también con el Bajo acoplamiento y Alta cohesión lo que facilita la centralización de actividades de cada elemento, y logra que estos realicen una única tarea dentro del sistema, además de permitir la colaboración entre clases o elementos del diseño sin que se afecte su reutilización y entendimiento cuando se encuentren aislados. (55)

### 3.3.1 Diagramas de clases de 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. Estos son de gran importancia, ya que permiten visualizar, especificar y documentar modelos estructurales. Los diagramas de clases de diseño forman parte de las realizaciones de casos de usos.

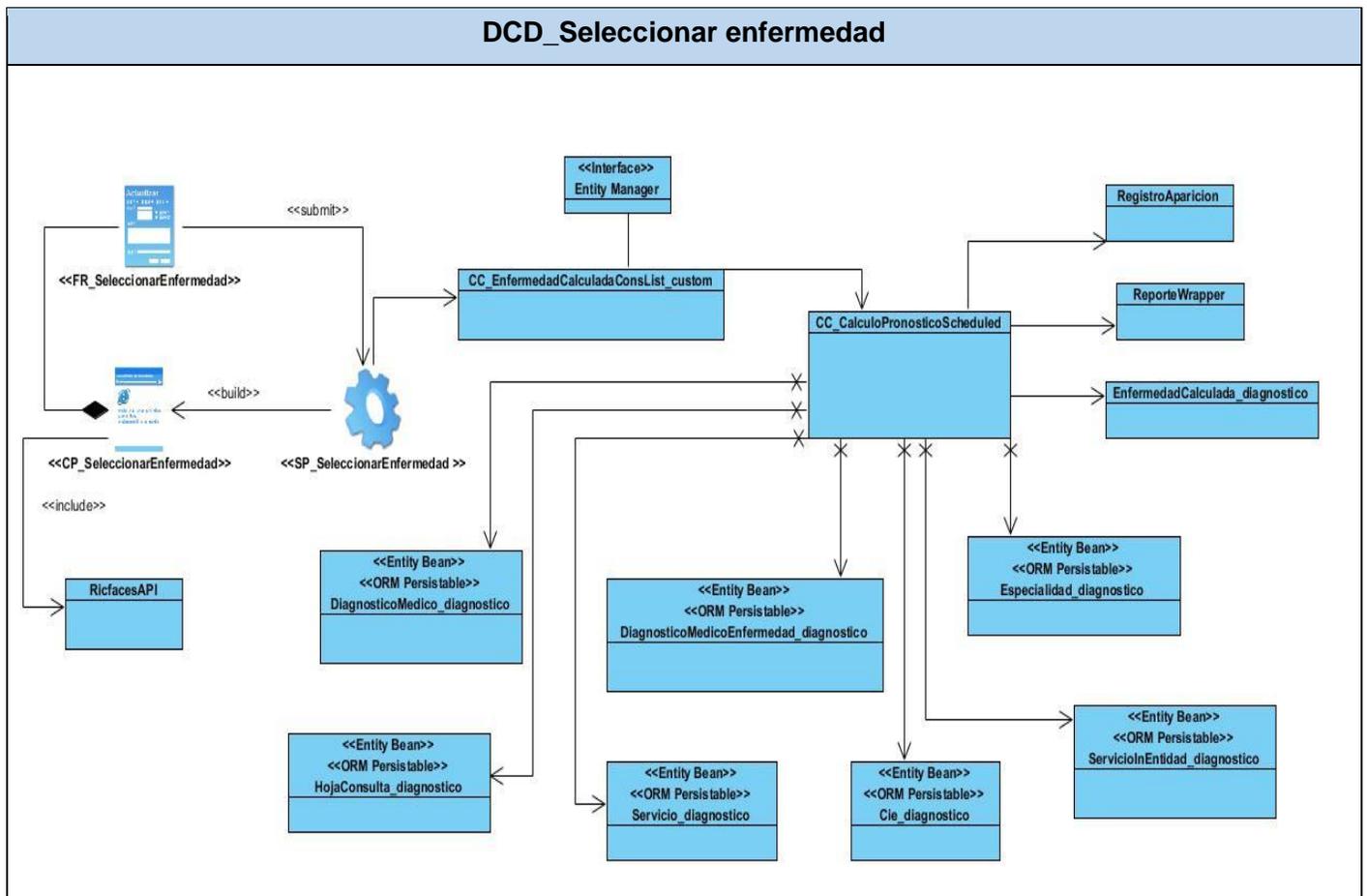


Figura 3.1 Diagrama de Clases de Diseño: “Seleccionar enfermedad”

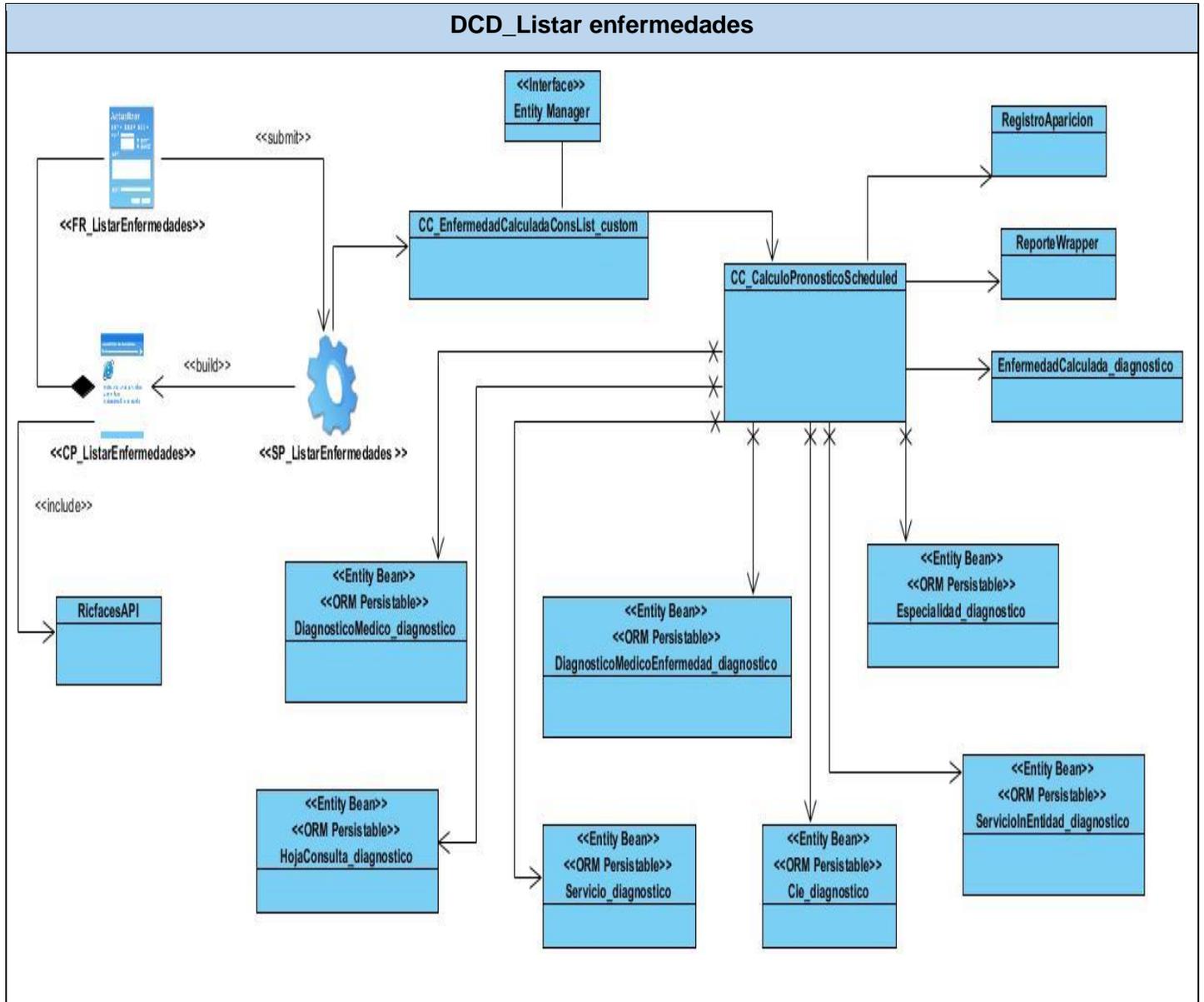


Figura 3.2 Diagrama de Clases de Diseño: "Listar enfermedades"

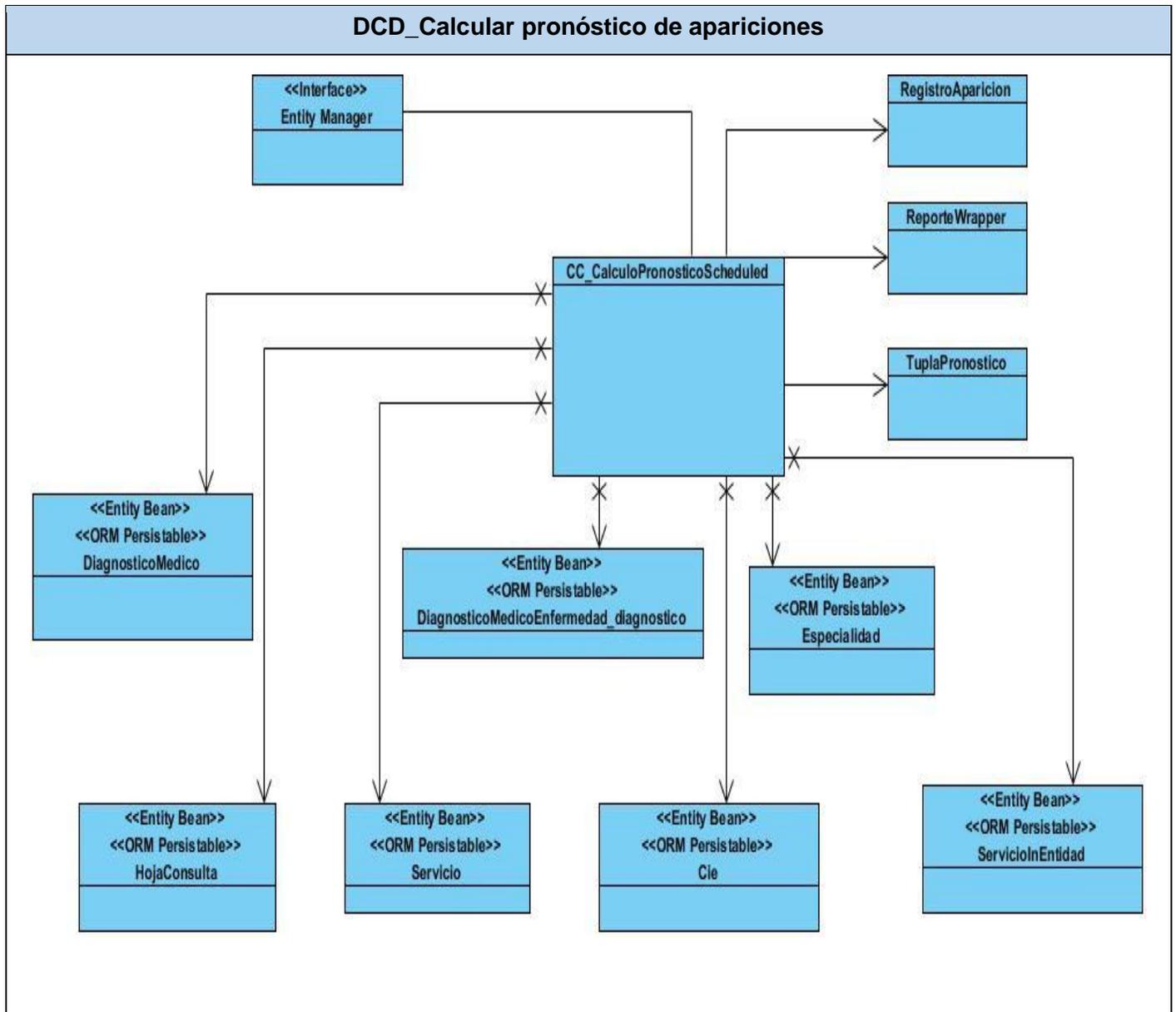


Figura 3.3 Diagrama de Clases de Diseño: “Calcular pronóstico de apariciones”

**Nota:** Para un mayor entendimiento de los diagramas de clases **Ver Artefacto:** “Modelo de diseño”.

### 3.3.2 Descripción de las clases de diseño

#### RegistroAparicion

**Propósito:** La clase entidad RegistroAparicion se utiliza como plantilla para construir objetos que contengan el código de una enfermedad y la cantidad de apariciones que posee una enfermedad en cada una de las especialidades. Esta clase es utilizada por la clase controladora CalculoPronosticoScheduled para construir la matriz de RegistroApariciones con los que va a trabajar el método de Series Temporales.



Figura 3.4 Clase RegistroAparicion

#### ReporteWrapper

**Propósito:** La clase ReporteWrapper se utiliza como plantilla para construir objetos encargados de almacenar las consultas realizadas a la base de datos y los datos resultantes de dichas consultas, para su posterior utilización en el cálculo del pronóstico de aparición una enfermedad.

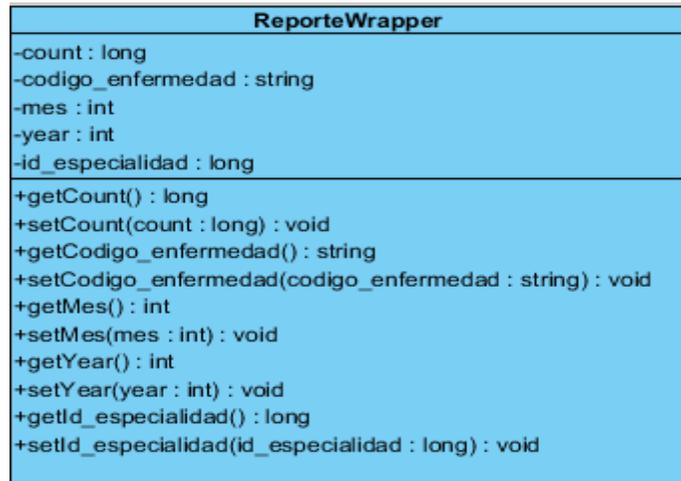


Figura 3.5 Clase ReporteWrapper.

### CalculoPronosticoScheduled

**Propósito:** La clase controladora CalculoPronosticoScheduled, es la encargada de realizar el cálculo del pronóstico de apariciones de una enfermedad a través del método de series de tiempo. Para la realización de este método es necesario la creación e inicialización de una matriz de tipo RegistroApariciones [][] , la cual contendrá los datos necesarios para la realización de la predicción.

El cálculo del pronóstico se realizara a través de una tarea programada que se ejecutará en un tiempo previamente definido.

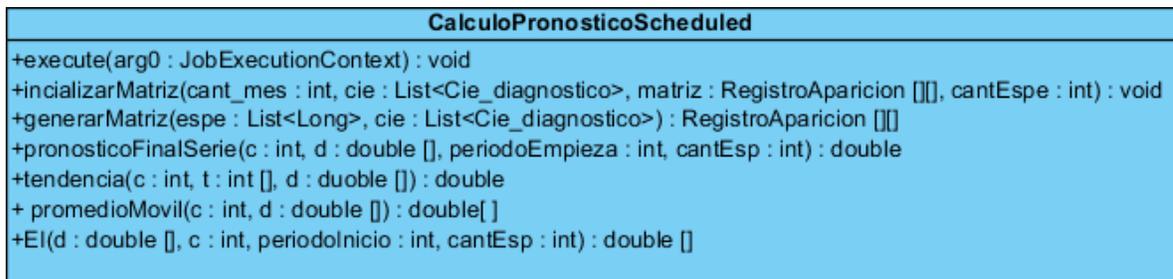


Figura 3.6 Clase CalculoPronosticoScheduled.

### EnfermedadCalculada\_diagnostico

**Propósito:** La clase entidad EnfermedadCalculada\_diagnostico se utiliza como plantilla para construir objetos que luego serán almacenados en la tabla enfermedad\_calculada de la base de datos.

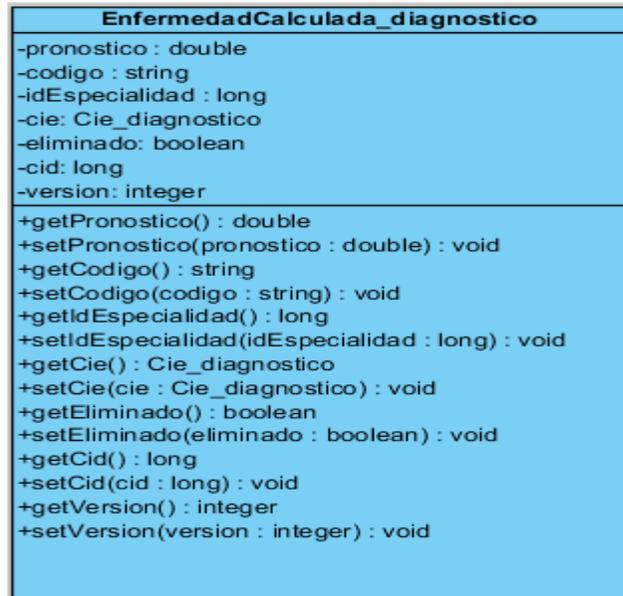


Figura 3.7 Clase EnfermedadCalculada\_diagnostico.

### EnfermedadCalculadaConsList\_custom

**Propósito:** La clase controladora EnfermedadCalculadaConsList\_custom es la encargada de retornar las listas de la tabla enfermedad\_calculada teniendo en cuenta los parámetros recibidos de la interfaz. Además de implementar los métodos de búsqueda de la lista a mostrar.

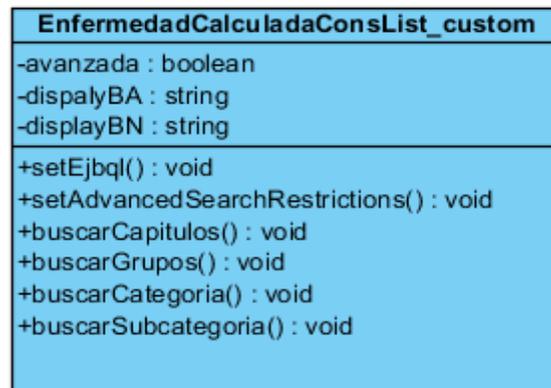


Figura 3.8 Clase EnfermedadCalculadaConsList\_custom.

### **Conclusiones**

Como resultado del estudio realizado en este capítulo, correspondiente al flujo de diseño, se identificaron las clases fundamentales que deben ser definidas para que el sistema funcione satisfactoriamente. Se definieron los patrones y la arquitectura a utilizar para brindar al desarrollador una idea más clara de lo que se debe implementar.

## **Capítulo 4. Implementación de las funcionalidades asociadas al componente para el proceso de emisión de diagnóstico.**

En este capítulo se describe el flujo de trabajo de implementación. Se detalla el Modelo de Datos, en el que se muestra la estructura donde se almacena toda la información requerida en el sistema. Luego se presenta el Modelo de Implementación, que está compuesto por el Diagrama de Despliegue y el Diagrama de Componentes. Estos diagramas, describen los componentes a construir, su organización y dependencias entre los nodos físicos en los que funcionará la aplicación. También se exponen aspectos referentes a la seguridad del sistema, así como las estrategias de codificación.

### **4.1 Modelo de Datos**

El Modelo de Datos describe la representación lógica y física de la información persistente manejada por el sistema. Puede ser inicialmente creado a través de la ingeniería inversa de un almacenamiento de datos persistentes que ya exista o puede ser inicialmente creado a partir de un conjunto de clases del diseño (entidades) persistentes en el modelo de diseño. Es usado para definir el mapeo entre las clases del diseño y las estructuras de datos. (56)

Las entidades son objetos de los que el sistema necesita guardar información, las cuales está constituida por uno o más atributos. Los atributos son las características asociadas a una entidad. Estos pueden ser clasificados en obligatorios, opcionales, claves foráneas y claves primarias (estas se dividen en simples y compuestas). Las relaciones, por su parte, muestran la forma en que dos entidades se asocian. Se representan mediante una línea que une a las dos entidades implicadas y manifiestan dos características principales: la cardinalidad y la obligatoriedad. (57)

La cardinalidad se refiere al número de ocurrencias de una entidad con respecto a la otra. La entidad de la que sale la relación tendrá tantas ocurrencias como indique el número asociado a la entidad a donde llega la relación señalando con una flecha el sentido de entrada, de no mostrarse el número de la cardinalidad se asume que la ocurrencia es de solamente una vez.

La obligatoriedad determina que ante la existencia de una entidad puede haber ocurrencias de otras relacionadas con esta. Si la ocurrencia es obligatoria, se representa mediante una línea continua, en caso contrario, se realiza a través de una línea discontinua.



#### 4.1.1 Descripción de las tablas de la base de datos

Atributos comunes en todas las entidades		
Atributos	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	Permite identificar quién realiza alguna acción sobre la entidad.

Tabla 4.1 Descripción de los atributos comunes entre todas las tablas.

enfermedad_calculada		
Atributos	Tipo	Descripción
pronostico	double	Pronóstico de apariciones de una enfermedad.
id_especialidad	long	Identificador de la especialidad medica
Id_enfermedad	long	Identificador de la enfermedad

Tabla 4.2 Descripción de la tabla enfermedad\_calculada.

## 4.2 Modelo de implementación

### 4.2.1 Diagrama de componentes

Un diagrama de componentes muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean estos de código fuente, librerías, binarios o ejecutables. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias a la hora de importar el código y

organizando estos subsistemas en capas. El mismo es utilizado para modelar la vista estática y dinámica de un sistema, además de mostrar organización entre sus componentes. (58)

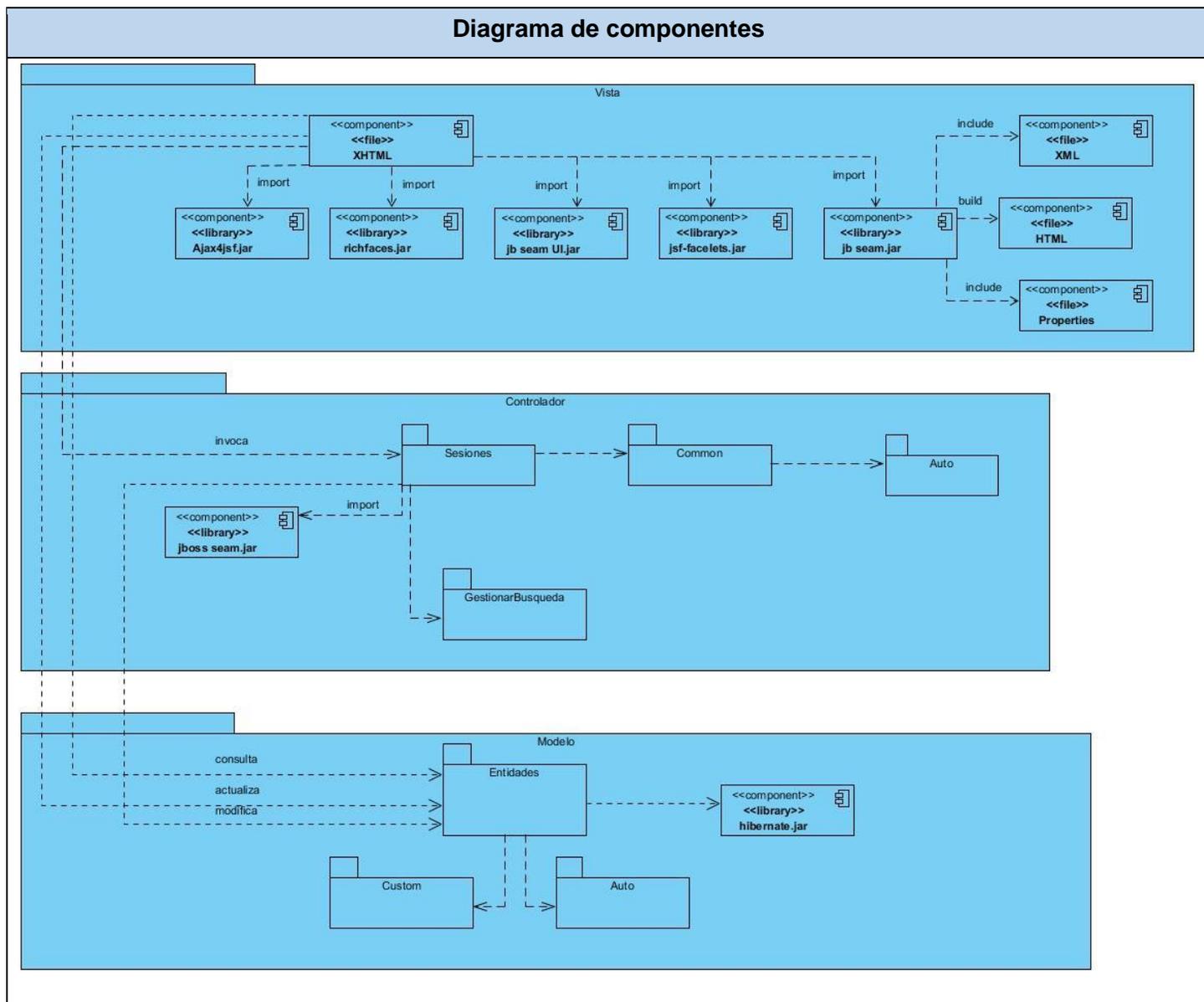


Figura 4.2 Diagrama de componentes.

### 4.2.2 Diagrama de despliegue

El Diagrama de Despliegue muestra las relaciones físicas entre los componentes de hardware y software en el sistema final. Es un grafo de nodos unidos por conexiones de comunicación. Estos muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

La arquitectura en tiempo de ejecución de la solución propuesta se modela haciendo usos de tres elementos de hardware: computadora cliente, servidor de aplicaciones y servidor de base de datos. Por otra parte la comunicación entre los nodos se rige por los protocolos HTTP para la agrupación entre la computadora cliente y el servidor de aplicaciones y TCP/IP para la conexión entre los servidores. (59)

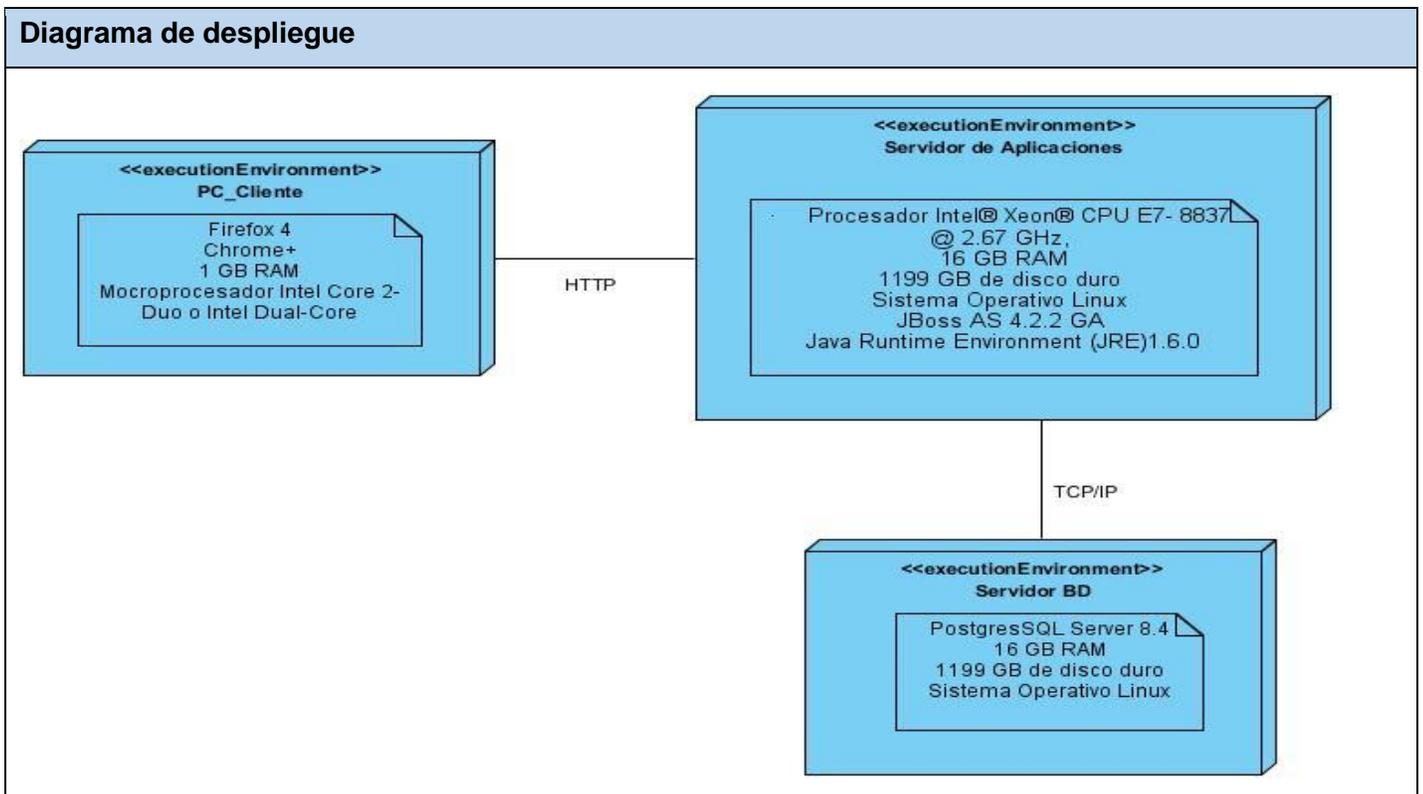


Figura 4.3 Diagrama de despliegue

### **4.3 Seguridad del sistema**

Un sistema informático se considera seguro cuando garantiza la confidencialidad, integridad y disponibilidad de sus datos. Debido a la importancia que poseen el proceso de atención a pacientes, a la necesidad de que estos sean realizados con la calidad requerida y a que el acceso a la información derivada de este proceso solo sea por las personas que se encuentren autorizadas, se hace necesario que se tenga en cuenta una serie de requisitos de seguridad de modo que no pueda afectar la el funcionamiento del componente y la integridad de los datos.

Para fomentar la seguridad, el Sistema de Información Hospitalaria propone un control de acceso a nivel de usuarios y contraseñas, así como una diferenciación de los mismos atendiendo al rol que desempeñan. Dichas contraseñas solo podrán ser cambiadas por los usuarios o por los administradores del sistema. Las actividades de los usuarios dentro del sistema serán registradas mediante trazas.

Para garantizar la fidelidad de los datos la información que sea intercambiada entre el sistema y otros sistemas debe ser cifrada, asegurando que no se lea ni modifique la información confidencial que se maneja. Además, toda la información ingresada al sistema nunca será eliminada físicamente de la base de datos, permitiendo al sistema recuperarse de posibles errores en su manejo.

Independientemente de las políticas de seguridad propias del sistema, estas no son efectivas si no existe organización en el manejo administrativo de la documentación y la autodeterminación en los hospitales, que garantice un uso correcto del sistema informático por las personas encargadas de operarlo. La institución o instituciones correspondientes, en este caso el ministerio y los hospitales, deberán incorporar personal confiable para el manejo del sistema informático y garantizar el respeto por las políticas de acceso que propone el sistema.

### **4.4 Estrategias de codificación. Estándares y estilos a utilizar en desarrollo del sistema**

Un estándar de codificación define un conjunto de reglas establecidas para la escritura del código fuente. Estas reglas son utilizadas como modelo de trabajo para garantizar un estilo de programación homogéneo durante la implementación del sistema y para entender el código por parte de otros programadores. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código adecuado es de gran importancia para la calidad del software y para obtener un buen rendimiento.

Entre las principales ventajas que tiene el uso de estándares se encuentran:

- Asegurar la legibilidad del código entre distintos programadores, facilitando el debugging del mismo.
- Facilitar la portabilidad entre plataformas y aplicaciones.
- Proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado.

Para la implementación de las funcionalidades propuestas se utilizaron varios estándares de codificación como son:

- **Métodos y variables**

Los métodos y variables deben escribirse utilizando el estilo lowerCamelCase, es decir, con la primera letra del nombre en minúsculas y con la primera letra de cada palabra interna en mayúsculas. En los métodos no se permiten caracteres especiales, mientras que en las variables se deben evitar en la medida de lo posible. El nombre ha de ser lo suficientemente descriptivo, de forma que al leerlo, se conozca su propósito.

Ejemplo: `int cantMeses; public double pronosticoFinalSerie()`

- **Comentarios**

Pueden estar ubicados al inicio de cada clase o función y al final de cada bloque de código. Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.

- **Declaraciones**

La declaración de las variables locales a una clase, método o bloque de código se realizan al principio del mismo y no justo antes de utilizarse la variable. Se recomienda una declaración por línea, ya que facilita los comentarios.

Ejemplo: `double pronostico; // pronóstico de ocurrencia`

`string codigo; // código`

- **Líneas y espacios en blanco**

Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

Se recomienda usar espacios en blanco entre estos operadores lógicos y aritméticos para lograr una mayor legibilidad en el código.

Ejemplo: `public double Tendencia()`

```
pronostico = pronostico1
```

La llave de apertura `<<{>>` aparece al final de la misma línea de la sentencia de declaración y la de cierre `<<}>>` empieza una nueva línea, luego de todas las líneas de sentencias.

Ejemplo: `public double Tendencia() {`

```
...
```

```
}
```

- **Clases**

Los nombres de las clases deben ajustarse a la entidad que representan, y su primera palabra debe ser un sustantivo. Si con una sola palabra no se puede nombrar dicha entidad, la segunda palabra debe ser un adjetivo, a menos que la palabra sea compuesta.

Ejemplo:

```
public class EnfermedadCalculada_diagnostico()
```

```
{
```

```
//...
```

```
}
```

- **Propiedades (Properties)**

Ejemplo:

```
public long getIdEsp() {  
    return idEsp;  
}  
  
public void setIdEsp(long idEsp) {  
    this.idEsp = idEsp;  
}
```

### **Conclusiones**

Durante el proceso de codificación se cumplió con los estándares y estilos definidos, lo que permitió obtener una aplicación entendible para todos los programadores y fácil de mantener en el transcurso del tiempo. Se logró alcanzar una aplicación con todas las funcionalidades previstas, que compense las principales necesidades de los clientes.

## Conclusiones

Con el desarrollo de las funcionalidades asociadas al componente para el proceso de emisión de diagnósticos del Sistema de Información Hospitalaria del CESIM, se concluye lo siguiente:

- El análisis de los sistemas relacionados con el campo de acción evidenció que los mismos no cumplen con todas las funcionalidades deseadas, ni permiten su integración al Sistema de Información Hospitalaria del CESIM.
- El análisis de los diferentes métodos y técnicas utilizadas en Sistemas Inteligentes permitió la selección de la técnica de Series Temporales para desarrollar las funcionalidades asociadas al componente de diagnóstico, permitiendo dada una serie de datos, predecir el comportamiento de los diagnósticos emitidos en la consulta.
- El uso del RUP permitió la obtención de los artefactos correspondientes al desarrollo de las funcionalidades asociadas al componente de diagnósticos garantizando, la correcta documentación del proceso de desarrollo.
- El desarrollo de las funcionalidades asociadas al componente de diagnósticos del Sistema de Información Hospitalaria del CESIM permitió garantizar una mejor utilización de este componente durante el proceso de emisión de diagnóstico.

### **Recomendaciones**

Con el objetivo de enriquecer la solución propuesta se proponen la siguiente recomendación para el trabajo presentado:

- Incorporar al componente para el proceso de emisión de diagnósticos, un diccionario de términos médicos, que permita a los especialistas la obtención las enfermedades asociadas al término introducido durante la búsqueda en el componente.

### Referencias Bibliográficas

1. Fernández Puerto, Francisco y Puerto Lara, Flarina. *Sistema de Información*. México : s.n., 2003.
2. Oviedo, Nolberto José. *Sistema asistente de diagnóstico de enfermedades cardiovasculares a través de síntomas*. Buenos Aires : s.n.
3. Loaiza, Roger. De la información a la informática. [En línea] [Citado el: 12 de enero de 2014.] [http://www.bvs.sld.cu/revistas/san/vol2\\_2\\_98/san15298.htm](http://www.bvs.sld.cu/revistas/san/vol2_2_98/san15298.htm).
4. Definición.de. [En línea] [Citado el: 10 de enero de 2014.] <http://definicion.de/inteligencia-artificial/#ixzz2u0CZIUAH>.
5. Microsoft Developer Network. [En línea] [Citado el: 12 de enero de 2014.] <http://msdn.microsoft.com/es-es/library/ms174923.aspx>.
6. cie10.org. [En línea] [Citado el: 20 de noviembre de 2013.] <http://www.cie10.org>.
7. Lozano, Laura y Fernández, Javier. *Razonamiento Basado en Casos: Una Visión General*.
8. —. *Razonamiento Basado en Casos: Una Visión General*.
9. Sánchez Corrales, Yosvannys. *Método computacional para el diagnóstico de la Hipertensión*. La Habana : s.n., 2012.
10. Prezi. [En línea] [Citado el: 21 de enero de 2014.] <http://prezi.com/-hrc-adxqrrs/estadistica-y-probabilidad/>.
11. Hinto, Perry R. *Statistic Splained*.
12. Martínez, Miguel Orlando . Scribd. [En línea] [Citado el: 24 de enero de 2014.] <http://es.scribd.com/doc/218143043/Tema-1-Miguel>.
13. Gonzáles Osuna, Alan, Mancera Gonzáles, Fransisco Israel y Navarro, Raymundo. SlideShare. [En línea] <http://www.slideshare.net/isaacgflores/anlisis-de-series-de-tiempo>.
14. Microsoft Developer Network. [En línea] [Citado el: 24 de enero de 2014.] <http://msdn.microsoft.com/es-es/library/ms174923.aspx>.
15. MedGadget. [En línea] [Citado el: 24 de enero de 2014.] <http://medgadget.es/2012/03/symcat-estudia-sus-sintomas-con-ayuda-de-la-informacion.html>.
16. Hergueta González, Celia. *Sistema de Detección y Tratamiento de Enfermedades Cutáneas*. Madrid : s.n., 2006.

## Referencias Bibliográficas

---

17. Oviedo, Nolberto José. *Sistema asistente de diagnóstico de enfermedades cardiovasculares a través de síntomas*. Buenos Aires : s.n.
18. *Módulo Consulta Externa, Manual de Usuario SIGHO*. México : s.n.
19. Cardozo Díaz, Jorge Luis. *Seade: Sistema experto para el análisis y detección de enfermedades*. Cuba : s.n., 2013.
20. Fernández, Karina y Toledo, Viviana. *DITRITS: Sistema experto para diagnóstico y tratamiento de enfermedades de transmisión sexual*. Cuba : s.n.
21. Gutiérrez, Mariela, Barroso Rodríguez, Yadira y Bedoya Rusenko, Jorge. *Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias*. Cuba : s.n.
22. Venete, Adriana. *Introducción a los Patrones de Arquitectura*. 2010.
23. [En línea] [Citado el: 21 de enero de 2014.] <http://www.lsi.us.es/docencia/get.php?id=1891>.
24. León, Jeimy. *Mis primeros pasos por el mundo de la arquitectura del software*.
25. Reindustria. [En línea] [Citado el: 24 de enero de 2014.] <http://redindustria.blogspot.com/2008/03/arquitectruas-clienteservidor.html>.
26. Oracle.com. *Java Platform, Enterprise Edition (JEE)*. [En línea] [Citado el: 25 de enero de 2014.] <http://www.oracle.com/technetwork/java/javaee/documentation/index.html>.
27. OsmosisLatina. [En línea] [Citado el: 27 de enero de 2014.] <http://www.osmosislatina.com/java/basico.htm>.
28. Arquitectura y Tecnología de Computadores. [En línea] <http://atc.ugr.es/~bprieto/paginas-web8/introduccionx.html>.
29. Álvarez, Miguel Angel y Gutiérrez, Manuel. Programación en Java Script. [En línea]
30. Curso de Java. [En línea] <http://tikal.cifn.unam.mx/~jsegura/LCGII/java2.htm>.
31. Facelets. [En línea] [Citado el: febrero de 3 de 2014.] <http://www.google.com/notebook/public/06237054388688325750/BDSW5QgoQ38aBIMsi>.
32. JBoss. [En línea] [Citado el: 3 de febrero de 2014.] <http://www.jboss.com/products/hibernate/>.
33. Suárez, Joé Manuel. Introducción a Richfaces. [En línea] [Citado el: 6 de febrero de 2014.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflIntro>.
34. DZone Refcardz. [En línea] [Citado el: 5 de febrero de 2014.] <http://refcardz.dzone.com/refcardz/seam-ui>.
35. SeamFramework.org. [En línea] [Citado el: 12 de febrero de 2014.] <http://seamframework.org/>.

## Referencias Bibliográficas

---

36. SeamFramework.org. [En línea] [Citado el: 15 de febrero de 2014.] <http://seamframework.org>.
37. Introducción a EJBs. [En línea] [Citado el: 15 de febrero de 2014.]  
<http://di002.edv.uniovi.es/~dflanvin/docencia/dasdi/teoria/Transparencias/13.%20Introducci%C3%B3n%20EJBs.pdf>.
38. Bauter, Cristina y King, Gavin. *Java Persistence with Hibernate*. 2005.
39. Gómez, Juan. Fundamentos de la metodología RUP. [En línea]  
<http://www.scribd.com/doc/297224/RUP>.
40. [En línea] [Citado el: 20 de enero de 2014.] <http://msdn.microsoft.com/es-es/library/ms174923.aspx>.
41. PostgreSQL. [En línea] [Citado el: 13 de febrero de 2014.]  
[http://www.freedownloadcenter.com/es/Programacion/Base\\_de\\_Datos\\_y\\_Redres/PostgreSQL\\_Maestro.html](http://www.freedownloadcenter.com/es/Programacion/Base_de_Datos_y_Redres/PostgreSQL_Maestro.html).
42. Bermejo Sanz, Laura. *Eclipse como IDE*.
43. SeamFramework.org. [En línea] [Citado el: 15 de febrero de 2014.]  
<http://docs.jboss.org/seam/latest/reference/en-US/html/gettingstartedwithjbossstools.html>.
44. Guia-ubuntu. [En línea] [Citado el: 20 de febrero de 2014.] [http://www.guia-ubuntu.com/index.php/PgAdmin\\_III](http://www.guia-ubuntu.com/index.php/PgAdmin_III).
45. Visual Paradigm. [En línea] [Citado el: 22 de febrero de 2014.] <http://www.visual-paradigm.com/news/vpsuite33/vpum163.jsp>.
46. Jaramillo, Wilmer. Software Libre de Venezuela 777, C.A. [En línea] [Citado el: 22 de febrero de 2014.]  
<http://wilmer.fedorapeople.org/files/presentations/JBoss.pdf>.
47. Vallejo, Ronny. Introducción a las Series Cronológicas. [En línea] [Citado el: 11 de mayo de 2014.]  
<http://rvallejos.mat.utfsm.cl/Time%20Series%20I%202013/TSbook.pdf>.
48. [En línea] [Citado el: 11 de mayo de 2014.]  
<http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/EDescrip/tema7.pdf>.
49. [En línea] [Citado el: 20 de marzo de 2014.]  
[http://www.upct.es/~metodos/Asignaturas/Diplomatura/Introduccion\\_estadistica/2008\\_2009/material\\_didactico/apuntes/TEMA5SERIESTEMPORALES.pdf](http://www.upct.es/~metodos/Asignaturas/Diplomatura/Introduccion_estadistica/2008_2009/material_didactico/apuntes/TEMA5SERIESTEMPORALES.pdf).
50. Tecnología y Synergix. [En línea] [Citado el: 1 de marzo de 2014.]  
<http://synergix.wordpress.com/2008/07/10/modelo-de-dominio>.
51. Presman, Roger. *Ingeniería de Software, un enfoque práctico*.

## Referencias Bibliográficas

---

52. Cáceres Tello, Jesús. Diagrama de Casos de Uso. [En línea] [Citado el: 12 de marzo de 2014.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.
53. Velázquez Carralero, Alejandro Mario. *IH-SW-DR-091 Alas-His\_Documento de Arquitectura del Sistema*. 2008.
54. Ochoa, Raquel. [En línea] [Citado el: 1 de marzo de 2014.] [http://digeset.ucol.mx/tesis\\_posgrado/Pdf/Raquel%20Ochoa%20Ornelas.pdf](http://digeset.ucol.mx/tesis_posgrado/Pdf/Raquel%20Ochoa%20Ornelas.pdf).
55. García, Joaquín. IngenieroSoftware. [En línea] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
56. García, Raúl. Modelo de Datos. [En línea] [Citado el: 15 de marzo de 2014.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos>.
57. [En línea] [Citado el: 20 de marzo de 2014.] [http://www.unirioja.es/cu/arjaime/Temas/02.Modelo\\_E\\_R.pdf](http://www.unirioja.es/cu/arjaime/Temas/02.Modelo_E_R.pdf).
58. Cuz Quispe, Victor Fabio. *Diagrama de Componentes*. 2012.
59. Quisbert Limanchi, Nancy Susana y Marca Hullapar, Hugo Michael. *Diagrama de despliegue*.
61. Ramírez, Sergio, Ponce, Cristina y Nolivos, Indira. *SISTEMA EXPERTO PARA LA DETECCION DE CANCER A LA GLANDULA TIROIDES - SIECAT*. Ecuador : s.n.
62. SeamFramework.org. [En línea] [Citado el: 2 de mayo de 2014.] <http://docs.jboss.org/seam/latest/reference/en-US/html/gettingstartedwithjbosstools.html>.
63. SeamFramework.org. [En línea] [Citado el: 14 de marzo de 2014.] <http://seamframework.org>.

## Bibliografía

- [En línea] [Citado el: 20 de enero de 2014.] <http://msdn.microsoft.com/es-es/library/ms174923.aspx>.
- [En línea] [Citado el: 21 de enero de 2014.] <http://www.lsi.us.es/docencia/get.php?id=1891>.
- Álvarez, Miguel Ángel y Gutiérrez, Manuel. Programación en Java Script.
- Arquitectura y Tecnología de Computadores, Universidad Granada. [En línea] [Citado el: 1 de febrero de 2014.] <http://atc.ugr.es/~bprieto/paginas-web8/introduccionx.html>.
- Bauter, Cristian y King, Gavin. Java Persistence with Hibernate. 2005.
- Bermejo Sanz, Laura. Eclipse como IDE.
- Cáceres Tello, Jesús. Diagrama de Casos de Uso. [En línea] [Citado el: 12 de marzo de 2014.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.
- Cardozo Díaz, Jorge Luis. Seade: Sistema experto para el análisis y detección de enfermedades. Cuba: s.n., 2013.
- cie10.org. [En línea] [Citado el: 20 de noviembre de 2013.] <http://www.cie10.org>.
- Corales, Yovannys Sánchez. Método computacional para el diagnóstico de la Hipertensión Arterial mediante el procesamiento analítico en línea y el razonamiento basado en casos. La Habana : s.n.
- Cruz Quispe, Victo Fabio, y otros. Diagrama de Componentes. 2012 : s.n.
- Curso de Java. [En línea] [Citado el: 2 de febrero de 2014.] <http://tikal.cifn.unam.mx/~jsegura/LCGII/java2.htm>.
- Definición. De. [En línea] [Citado el: 10 de enero de 2014.] <http://definicion.de/inteligencia-artificial/#ixzz2u0CZIUAH>.
- DZone Refcardz. [En línea] [Citado el: 10 de febrero de 2014.] <http://refcardz.dzone.com/refcardz/seam-ui>.
- Facelets. [En línea] [Citado el: 3 de febrero de 2014.] <http://www.google.com/notebook/public/06237054388688325750/BDSW5QgoQ38aBIMsi>.
- Fernández Puerto, Francisco J. y Puerto, Florina Lara. Sistema de Información. México : s.n., 2003.
- Fernández, Karina y Toledo, Viviana. DITRITS: Sistema experto para diagnóstico y tratamiento de enfermedades de transmisión sexual. Cuba : s.n.
- García, Joaquín. IngenieroSoftware. [En línea] [Citado el: 1 de marzo de 2014.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

## Bibliografía

---

- García, Raúl. Modelo de Datos. [En línea] [Citado el: 4 de marzo de 2014.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos>.
- Gómez, Juan. Fundamentos de la metodología RUP. [En línea] [Citado el: 14 de febrero de 2014.] <http://www.scribd.com/doc/297224/RUP>.
- Guia-ubuntu. [En línea] [Citado el: 20 de febrero de 2014.] [http://www.guia-ubuntu.com/index.php/PgAdmin\\_III](http://www.guia-ubuntu.com/index.php/PgAdmin_III).
- Gutiérrez, Mariela, Barroso Rodríguez, Yadira y Bedoya Rusenko, Jorge. Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias. Cuba : s.n.
- Hergueta González, Celia. Sistema de Detección y Tratamiento de Enfermedades Cutáneas Madrid : s.n., 2006.
- Hinto, Perry R. Statistic Splained.
- Jaramillo, Wilmer. Software Libre de Venezuela 777, C.A. [En línea] [Citado el: 22 de febrero de 2014.] <http://wilmer.fedorapeople.org/files/presentations/JBoss.pdf>.
- JBoss. [En línea] [Citado el: 3 de febrero de 2014.] <http://www.jboss.com/products/hibernate/>.
- León , Jeimy. Mis primeros pasos por el mundo de la arquitectura del software.
- Loaza, Roger. De la información a la informática. [En línea] [Citado el: 12 de enero de 2014.] [http://www.bvs.sld.cu/revistas/san/vol2\\_2\\_98/san15298.htm](http://www.bvs.sld.cu/revistas/san/vol2_2_98/san15298.htm).
- Lozano, Laura y Fernández, Javier. Razonamiento Basado en Casos: Una Visión General.
- Lozano, Laura y Fernández, Javier. Razonamiento Basado en Casos: Una visión general.
- MedGadget. [En línea] [Citado el: 24 de enero de 2014.] <http://medgadget.es/2012/03/symcat-estudia-sus-sintomas-con-ayuda-de-la-informacion.html>.
- Microsoft Developer Network. [En línea] [Citado el: 12 de enero de 2014.] <http://msdn.microsoft.com/es-es/library/ms174923.aspx>.
- Microsoft Developer Network. [En línea] [Citado el: 24 de enero de 2014.] <http://msdn.microsoft.com/es-es/library/ms174923.aspx>.
- Módulo Consulta Externa, Manual de Usuario SIGHO.
- Ochoa, Raquel. [En línea] 1 de marzo de 2014. [http://digeset.ucol.mx/tesis\\_posgrado/Pdf/Raquel%20Ochoa%20Ornelas.pdf](http://digeset.ucol.mx/tesis_posgrado/Pdf/Raquel%20Ochoa%20Ornelas.pdf).
- Oliveros, Hugo. TENDENCIAS Y SERIES DE TIEMPO. Universidad de Colombia : s.n.

## Bibliografía

---

- Oracle.com. [En línea] [Citado el: 25 de enero de 2014.]. Java Platform, Enterprise Edition (JEE) <http://www.oracle.com/technetwork/java/javaee/documentation/index.html>.
- OsmosisLatina. [En línea] [Citado el: 27 de enero de 2014.] <http://www.osmosislatina.com/java/basico.htm>.
- Oviedo, Nolberto José. Sistema asistente de diagnóstico de enfermedades cardiovasculares a través de síntomas. Buenos Aires : s.n.
- Oviedo, Norberto José. Sistema asistente de diagnóstico de enfermedades cardiovasculares a través de síntomas. Buenos Aires : s.n.
- PostgreSQL. [En línea] [Citado el: 13 de febrero de 2014.] [http://www.freedownloadscenter.com/es/Programacion/Base\\_de\\_Datos\\_y\\_Red/PostgreSQL\\_Maestro.html](http://www.freedownloadscenter.com/es/Programacion/Base_de_Datos_y_Red/PostgreSQL_Maestro.html).
- Presman, Roger. Ingeniería de Software, un enfoque práctico.
- Prezi. [En línea] [Citado el: 21 de enero de 2014.] <http://prezi.com/-hrc-adxqrrs/estadistica-y-probabilidad/>.
- Quisbert Limanchi, Nancy Susana y Marca Hullapar, Hugo Michael. Diagrama de despliegue.
- Reindustria. [En línea] [Citado el: 24 de enero de 2014.] <http://redindustria.blogspot.com/2008/03/arquitectruas-clienteservidor.html>.
- Sánchez Corrales, Yosvannys. Método computacional para el diagnóstico de la Hipertensión. La Habana : s.n., 2012.
- SeamFramework.org. [En línea] [Citado el: 12 de febrero de 2014.] <http://seamframework.org/>.
- SeamFramework.org. [En línea] [Citado el: 15 de febrero de 2014.] <http://docs.jboss.org/seam/latest/reference/en-US/html/gettingstartedwithjbossstools.html>.
- SeamFramework.org. [En línea] [Citado el: 15 de febrero de 2014.] <http://seamframework.org>.
- Sergio Ramírez Q, Cristina Ponce E, Indira Nolivios A. SISTEMA EXPERTO PARA LA DETECCION DE CANCER A LA GLANDULA TIROIDES - SIECAT. s.l.: Ecuador.
- Soto, Carlos. Sistema Experto para el diagnóstico del síndrome de Guilliam Barre.
- Suárez, José Manuel Sánchez. Introducción a Richfaces. [En línea] [Citado el: 6 de febrero de 2014.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflntro>.
- Tecnología y Synergix. [En línea] [Citado el: 20 de febrero de 2014.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio>.

## Bibliografía

---

- Tello, Jesús Cáceres. [En línea] [Citado el: 3 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/dd409377.aspx>.
- Vallejo, Juan Santaella. Un Sistema Experto: MyCIN.
- Velázquez Carralero, Alejandro Mario. IH-SW-DR-091 Alas-His\_Documento de Arquitectura del Sistema. 2008.
- Venete, Adriana. Introducción a los Patrones de Arquitectura. 2010.
- Visual Paradigm. [En línea] [Citado el: 22 de febrero de 2014.] <http://www.visual-paradigm.com/news/vpsuite33/vpuml63.jsp>.

## Anexos

### 1. Ejemplo ilustrativo del cálculo del método de Series de Tiempo

Como el objetivo de este ejemplo es ilustrar el procedimiento a efectuar en el cálculo de una serie temporal, solo se tomaron 10 observaciones. Es importante resaltar que la fiabilidad de la predicción será mayor a medida que exista mayor número de observaciones antecedentes.

El método inicia utilizando una recta  $y(t) = a + bt$  como la función de ajuste a los datos mostrados, los cuales representan la cantidad de apariciones de una enfermedad en un período de tiempo:

$t$	$e$
1	45
2	38
3	30
4	29
5	18
6	24
7	33
8	17
9	22
10	25
<b>55</b>	<b>281</b>

Tabla A.1 Observaciones correspondientes a 10 períodos.

El objetivo que se persigue con este método es hallar el pronóstico para el período 11.

Para estimar la Tendencia (T) de la serie, primeramente se calculan las componentes  $a$  y  $b$  de la recta  $y(t) = a + bt$  por el método de mínimos cuadrados, cuya fórmula es:

## Anexos

$$a = \frac{\sum e}{n} - (b) \frac{\sum t}{n} \quad b = \frac{\frac{\sum et}{n} - \frac{\sum e}{n} * \frac{\sum t}{n}}{\frac{\sum t^2}{n} - \left(\frac{\sum t}{n}\right)^2}$$

<i>t</i>	<i>e</i>	<i>e * t</i>	<i>t<sup>2</sup></i>
1	45	45	1
2	38	76	4
3	30	90	9
4	29	116	16
5	18	90	25
6	24	144	36
7	33	231	49
8	17	136	64
9	22	198	81
10	25	250	100
<b>55</b>	<b>281</b>	<b>1376</b>	<b>385</b>

Sustitución de los valores en la ecuación de **a** y **b**

$$b = \frac{1376/10 - (281/10)*(55/10)}{385/10 - (55/10)^2} = \text{Dividimos por 10 cada número: } \frac{137.6 - (28.1)*(5.5)}{38.5 - (5.5)^2}$$

$$\text{Seguimos los cálculos así: } b = \frac{137.6 - 154.55}{38.5 - 30.25} = \frac{-16.95}{8.25} = -2.05$$

$$\text{El valor para } a = 281/10 - (-2.05)(55/10) = 28.1 + 11.33 = 39.43$$

Una vez calculado los valores **a** y **b**, son sustituidos en la ecuación de la recta:

Por lo tanto la ecuación final sería  $T = 39.43 + (-2.05)t \Rightarrow T = 39.43 - 2.05t$

Luego se reemplaza el período a pronosticar en el componente  $t$  de la ecuación y se obtienen el siguiente resultado:

$$T = 39.43 - (2.05 * 11)$$

$$T = 16.8$$

El valor hallado es T (Tendencia), que significa que si todo sigue igual, debería aparecer esta cantidad de enfermedades en el período 11.

### Cálculo de la E (Estacionalidad)

Buscando quitarle irregularidad a la cantidad de enfermedades, se utiliza una herramienta llamada Promedio Móvil, este puede ser impar o par, el más utilizado es el Promedio Móvil base 3, y consiste en sacar un promedio primero de las tres cifras iniciales de ventas, o sea sumar y dividir por 3, luego se hace lo mismo con el segundo, tercer y cuarto período, y así sucesivamente.

El resultado de la primera suma de los 3 períodos iniciales quedará ubicado en el segundo período, el cual corresponde a la mitad visual de 3 cifras  $(1 + 2 + 3)/3 = 2$

$t$	$e$	$PM3$
1	45	-
2	38	37,67
3	30	32,33
4	29	25,67
5	18	23,67
6	24	25,00
7	33	24,67
8	17	24,00

9	22	21,33
10	25	-
<b>55</b>	<b>281</b>	

Tabla A.2 Valores del Promedio Móvil.

**Revisando:** 37.67 es el resultado de  $(45 + 38 + 30)/3$ , y luego  $(38 + 30 + 29)/3 = 32.33$  y así sucesivamente.

**Nota:** Pasos realizados hasta el momento.

1) Calcular a y b.

2) Definir la ecuación.

3) Hallar la tendencia de los períodos a pronosticar (T).

4) Hallar el promedio móvil de la cantidad de enfermedades, para suavizar la gráfica (graficar las enfermedades contra el tiempo y en la misma gráfica colocar el promedio móvil con el tiempo y se podrá observar que el promedio móvil ha suavizado la gráfica inicial). En matemáticas el proceso de suavizar la gráfica se llama deflactor.

En la siguiente tabla la columna del promedio móvil se llamará **TC**, ¿por qué?, de los cuatro conceptos: **T, E, C e I**, se ha quitado la **E** y la **I** que eran los que poseían irregularidades.

Ahora se procede a dividir la cantidad de enfermedades (que tienen los 4 conceptos incluidos o sea cantidad de enfermedades = **TECI**) entre **TC**, resultado  $TECI/TC = EI$  (se cancelan las T y las C). Esto se realiza para cada período.

<b>t</b>	<b>e</b>	<b>PM3 (TC)</b>	<b>EI</b>
1	45	-	-

2	38	37,67	1,01
3	30	32,33	0,93
4	29	25,67	1,13
5	18	23,67	0,76
6	24	25,00	0,96
7	33	24,67	1,34
8	17	24,00	0,71
9	22	21,33	1,03
10	25	-	-
<b>55</b>	<b>281</b>		

Tabla A 3. Valores de calculados de TC y EI.

**Nota:** Se han añadido dos pasos más.

5) La columna resultante del promedio móvil se llamará TC.

6) Dividir la cantidad de enfermedades (TECI)/TC, obtener EI y seguir con el paso 7.

7) Ahora ubicar EI en sus respectivos períodos.

Hasta el momento no se había definido los períodos de tiempo con los que se trabajará. Una serie de tiempo se puede trabajar para períodos totales mayores a un año. Se debe tener en cuenta cómo se puede dividir el año:

Semestres=2

Cuatrimestres=3

Trimestres=4

Bimensual=6

Meses=12

*Nota: En el presente ejemplo se trabajara con cuatrimestres.*

El primer período corresponde al tercer cuatrimestre del año 2009, entonces se tendrá lo siguiente:

1	2	3
		-
1,01	0,93	1,13
0,76	0,96	1,34
0,71	1,03	-

Tabla A 4. Ubicación de EI en los períodos correspondientes

Comenzamos en el tercer período ya que las enfermedades del primer período son las del tercer cuatrimestre.

*Paso 8) Calcular promedio aritmético de cada período:*

1	2	3
		-
1,01	0,93	1,13
0,76	0,96	1,34
0,71	1,03	-
2,48	2,92	2,47
<b>0,83</b>	<b>0,97</b>	<b>1,23</b>

Tabla A 5. Promedio aritmético de cada período.

O sea, se suma cada columna y se divide por el número de cifras que tiene cada período, ejemplo primera columna:  $(1.01+0.76+0.71)= 2.48$ , esto dividido entre 3= 0.83

*Paso 9) La suma de los promedios aritméticos obtenidos debe ser igual al número de períodos, en este caso 3, si no lo es, debemos hacer regla de 3.*

1	2	3
		-
1,01	0,93	1,13
0,76	0,96	1,34
0,71	1,03	-
2,48	2,92	2,47
0,83	0,97	1,23
<b>0,82</b>	<b>0,96</b>	<b>1,22</b>

Como el resultado dio 3.03, se aplica regla de 3 con cada período.

*Paso 10) Los resultados obtenidos serán E, o sea 0.82-0.96 y 1.22, ¿por qué E? Porque con el promedio aritmético se le quita la irregularidad (I) a los períodos, luego EI se transforma en E.*

*Paso 11) Ubicar E en los períodos respectivos:*

<i>t</i>	<i>e</i>	<i>PM3</i>	<i>EI</i>	<i>E</i>
1	45	-	-	1,22
2	38	37,67	1,01	0,82
3	30	32,33	0,93	0,96
4	29	25,67	1,13	1,22
5	18	23,67	0,76	0,82
6	24	25,00	0,96	0,96
7	33	24,67	1,34	1,22
8	17	24,00	0,71	0,82
9	22	21,33	1,03	0,96

## Anexos

10	25	-	-	1,22
<b>55</b>	<b>281</b>			

Tabla A 6. Ubicación de los valores E, EI, PM3.

Es importante tener en cuenta que se colocó 1.22 en el primer período, la cual corresponde a la E del período inicial que en este caso era el 3er cuatrimestre, ¿cuál será la E que le corresponde al período 11 a pronosticar? El período 11 cae en el primer cuatrimestre, por lo que le corresponde el valor de Estacionalidad (E)= 0.82.

*Paso 12) Colocar la E de los períodos a pronosticar:*

	<b><i>T</i></b>	<b><i>E</i></b>
11	16,80	0,82

*Paso 13) Multiplicar T x E:*

	<b><i>T</i></b>	<b><i>E</i></b>	<b><i>T X E</i></b>
11	16,80	0,82	13,73

*Paso 15) Los resultados obtenidos serán el pronóstico final de la cantidad de enfermedades para el período 11.*

## 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.

**Biblioteca de componentes:** Provee opciones poderosas con el objetivo de extender la lógica del negocio en los procesos, o para extender la capa de integración. La gran mayoría de bibliotecas de componentes contienen una amplia selección de modelos pre-programados y probados que van desde simples componentes pasivos y las funciones de control, a los modelos más complejos.

**Caching:** Su principal funcionalidad es almacenar temporalmente los datos frecuentemente accedidos más cerca del solicitante de los mismos. El caching se ha usado desde hace mucho tiempo en sistemas de ordenadores y redes para aumentar las prestaciones.

**Clustering:** Es la división de los datos en grupos de objetos similares. Este proceso ayuda a la agrupación estructural de un conjunto de datos.

**Código:** Cifras, clave. En informática se utiliza para referirse a un conjunto de instrucciones en un lenguaje de programación.

**Compilación:** Acción de compilar o reunir en una mismo contexto extractos de diferentes contextos. En informática se refiere a la acción de correr el código escrito en algún lenguaje y hacerlo funcionar.

**Componente:** Parte modular de un sistema de software. Se refiere a la agrupación de varias funcionalidades y estructuras que toman sentido en un contexto determinado.

**Dismorfias:** Es un trastorno somato morfo que consiste en una preocupación importante y fuera de lo normal por algún defecto percibido en las características físicas (imagen corporal), ya sea real o imaginado. Si dicho defecto existe, la preocupación y ansiedad experimentada por estas personas es excesiva, ya que lo perciben de un modo exagerado.

**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.

**HTTP:** Es un protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. Se usa en cada transacción de datos en la web. Es usado por gran cantidad de entidades pues la información que manejan es de suma confidencialidad e importancia y por ello se necesita mantener cifrado el canal de transferencia.

**Infraestructura tecnológica:** Es el conjunto de hardware y software sobre el que se asientan los diferentes servicios que las entidades necesitan tener en funcionamiento para poder llevar a cabo toda su actividad.

**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 inferencia:** Es un programa que, mediante el empleo de los conocimientos puede resolver el problema que está especificado. Lo resuelve gracias a los datos que contiene la base de conocimientos.

**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.

**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.

**Plug-ins:** Programas que deben instalarse en los ordenadores clientes para que otros puedan reconocer y procesar determinados tipos de archivo y realizar determinadas funcionalidades.

**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.

**TCP/IP:** Protocolo de control de transmisión /Protocolo de Internet o un conjunto de protocolos. Representa las reglas de comunicación para internet y se basa en las direcciones IP de cada equipo en la red para poder enrutar paquetes de datos.

**Workspaces:** Es un lugar en para guardar, tener acceso y compartir información.