



# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

## FACULTAD 7

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

*Módulo Medios de Diagnóstico del Subsistema Web del Sistema  
Integral para la Atención Primaria*

### **Autores:**

*Yisel Martínez Díaz*

*Dailier Moreno Rodríguez*

### **Tutores:**

*Ing. Abdel Alonso Martínez*

*Ing. Alberto Ortega Palacios*

**Ciudad de La Habana, julio de 2010**

**“Año 52 de la Revolución”**

## Resumen

Actualmente, no existe un estándar nacional que permita manejar la información de los exámenes complementarios en las áreas de los Medios de diagnóstico de las instituciones de Atención Primaria de Salud. El procesamiento se realiza de forma manual, resultando engorroso para el personal que allí labora debido a que no se cuenta con un software que gestione todo lo relacionado a estas áreas. Por estas razones el objetivo de la investigación es desarrollar el Módulo Medios de diagnóstico del Subsistema Web del Sistema Integral para la Atención Primaria.

El sistema está basado en tecnologías libres y multiplataforma, desarrollado sobre el lenguaje Java. Para la gestión y almacenamiento de los datos se utiliza PostgreSQL y JBoss Server como servidor de aplicaciones. Se emplean algunas de las tecnologías asociadas al desarrollo con Java como son el framework de los componentes de la interfaz de usuario del lado del servidor Java Server Faces (JSF), el motor de plantilla Facelets, así como el uso del framework de código abierto RichFaces, el cual permite añadir capacidades Ajax dentro de aplicaciones JSF, entre otros.

Con el desarrollo de esta aplicación se espera proveer una atención de salud con calidad, que garantice la seguridad y confiabilidad de la información médica, así como la comunicación entre las áreas de los medios de diagnóstico. Se registrarán los resultados de algunos exámenes complementarios pertenecientes a las áreas definidas realizados a cada paciente permitiendo consultarlos cuando se necesite.

**Palabras Claves:** *Sistema Integral para la Atención Primaria, Módulo Medios de diagnóstico, exámenes complementarios.*

<b>Introducción .....</b>	<b>1</b>
<b>Capítulo 1. Fundamentación Teórica.....</b>	<b>5</b>
1.1    Conceptos Asociados.....	5
1.2    Sistema Nacional de Salud.....	6
1.3    Informatización del SNS.....	7
1.4    Sistema Integral de Atención Primaria de Salud .....	7
1.5    Sistemas automatizados existentes.....	8
1.6    Tecnologías y herramientas utilizadas .....	11
1.6.1 <i>Proceso de Desarrollo</i> .....	11
1.6.2 <i>Estilos Arquitectónicos</i> .....	14
1.6.2 <i>Tecnologías en la capa de Presentación</i> .....	16
1.6.3 <i>Tecnologías en la capa de Negocio</i> .....	18
1.6.5 <i>Tecnologías en la capa de Acceso a Datos</i> .....	19
1.6.6 <i>Lenguaje de Programación</i> .....	20
1.6.7 <i>Herramientas a utilizar</i> .....	20
<b>Capítulo 2. Características del Sistema.....</b>	<b>22</b>
2.1    Flujo actual de los procesos.....	22
2.1.1 <i>Proceso Realizar Análisis de laboratorio clínico</i> .....	22
2.1.2 <i>Proceso Realizar Prueba Cutánea</i> .....	23
2.1.3 <i>Proceso Realizar Endoscopia</i> .....	23
2.1.4 <i>Proceso Realizar Drenaje Biliar</i> .....	23
2.1.5 <i>Proceso Realizar Electrocardiograma</i> .....	24
2.1.6 <i>Proceso Realizar Prueba de Refracción</i> .....	24
2.2    Información que se maneja .....	24
2.3    Objeto de Informatización .....	24
2.4    Modelado del negocio .....	25
2.4.1 <i>Actores involucrados en el proceso de negocio</i> .....	25
2.4.2 <i>Procesos Generales del Negocio</i> .....	27

2.4.3	Diagrama de Proceso de Realizar Análisis de Laboratorio Clínico.....	28
2.4.3.1	Descripción del Proceso.....	28
2.4.4	Diagrama de Proceso Realizar Prueba Cutánea. ....	30
2.4.4.1	Descripción del Proceso.....	30
2.4.5	Diagrama de Proceso de Realizar Endoscopia.....	31
2.4.5.1	Descripción del Proceso.....	32
2.5	Propuesta del Sistema .....	33
2.5.1	Especificación de requerimientos de software.....	33
2.5.2	Requerimientos funcionales .....	33
2.5.3	Requerimientos no funcionales.....	35
2.6	Descripción de prototipos no funcionales.....	37
<b>Capítulo 3. Diseño del Sistema .....</b>	<b>40</b>	
3.1	Modelo de Diseño.....	40
3.1.1	Definición de elementos de diseño .....	40
3.1.2	Patrones de Diseño.....	42
3.1.3	Diagramas de clases del diseño.....	43
3.1.4	Descripción de Clases de Registrar resultados cuantitativos.....	47
<b>Capítulo 4. Implementación.....</b>	<b>56</b>	
4.1	Justificación de interacción con otros sistemas .....	56
4.2	Modelo de implementación.....	56
4.2.1	Diagrama de despliegue.....	56
4.3	Estrategias de codificación. Estándares y estilos a utilizar.....	57
4.4	Tratamiento de Excepciones.....	61

## TABLA DE CONTENIDO

---

Conclusiones .....	63
Recomendaciones .....	64
Referencias Bibliográficas .....	65
Bibliografía.....	67
Glosario de Términos.....	69

## Introducción

La salud pública en Cuba es un logro de la Revolución y del socialismo. Desde el 1ero de Enero de 1959 se comenzó a trabajar por la creación del Sistema Nacional de Salud (SNS) que llevó la acción del trabajador de la salud a los lugares más apartados. El sistema creado comenzó a realizar importantes reformas a partir de los años 60 como parte fundamental de las transformaciones del período revolucionario y en respuesta al respeto más absoluto de uno de los derechos humanos fundamentales de todo ciudadano. [1]

El SNS cubano fue creado con el objetivo de garantizar la equidad y accesibilidad que requieren todos los ciudadanos. Está considerado un sistema complejo, en cuyas relaciones internas y externas descansan los procesos que apoyan la aplicación de planes, programas y acciones que garantizan el control y seguimiento de los objetivos encaminados a garantizar la salud del país.

Está compuesto por tres niveles básicos de atención, la Atención Primaria de Salud (APS) que constituye el primer nivel de contacto del SNS con las personas de la comunidad; la Atención Secundaria de Salud que se brinda a nivel de las instituciones hospitalarias y generalmente son de carácter provincial y la Atención Terciaria de Salud que comprende servicios altamente especializados y sólo se brinda en determinados centros.

En Cuba la APS es la piedra angular del Sistema Nacional de Salud y surge desde el mismo comienzo de la civilización y el desarrollo humano. Su estrategia es abarcadora, compleja e integradora. Promueve la reorganización de los servicios de atención a la salud de forma que la atención secundaria y terciaria, actúen como apoyo y derivación del nivel de atención primaria. [2]

En la APS la lucha social contra la enfermedad se orienta hoy claramente al logro de una medicina preventiva que pretende evitar la aparición de enfermedades. La prevención y curación de las enfermedades del hombre, objeto de la medicina, precisa para su real eficacia de la coordinación de todas aquellas disciplinas que tienen implicaciones en los tres procesos básicos sobre los que se orienta el quehacer médico de cada día, el **diagnóstico**, el reconocimiento de un agente o factor como causa del estado morbo y, por último, el terapéutico.

Con el propósito de mejorar los servicios médicos, la atención al paciente y agilizar los procesos relacionados con el trabajo diario de los galenos, el país se ha trazado como objetivo la informatización del sector de la salud. Por eso en los últimos años un grupo de instituciones cubanas y el propio Ministerio de Salud Pública (MINSAP) han desarrollado sistemas encaminados a lograr este fin.

Una de estas instituciones cubanas es la Universidad de las Ciencias Informáticas (UCI), teniendo como principal objetivo masificar el conocimiento de la informática en todo el país, además de crear software para los diferentes sectores de la sociedad. Existen varias facultades y cada una concentra su producción hacia un perfil específico. La Facultad 7 es la encargada de producir software para la salud, está estructurada por varios departamentos y uno de ellos es el encargado de informatizar los principales procesos de la APS.

El objetivo fundamental del Departamento de APS consiste en la creación de un Sistema Integral de Atención Primaria de Salud (alasiAPS) que abarque los puntos principales del espectro de la APS, elevando la salud en la comunidad, la prevención, prescripción, investigación, recuperación y predicción de los problemas del paciente. Este sistema gestiona información de algunos de los procesos llevados a cabo en la APS, agrupándolos por módulos entre los que se pueden mencionar el de Medicina Familiar, Clínico Quirúrgico, Enfermería y Salud Mental entre otros.

Cuando los pacientes requieren de algún examen complementario se dirigen al departamento de Medios de diagnóstico correspondiente, pero antes deben haber sido atendidos por la recepcionista, la que se encarga de recoger los datos del paciente, así como los exámenes asociados empleando para esto demasiado tiempo. Teniendo en cuenta el número de pacientes que acuden diariamente a los policlínicos y el trabajo realizado por la recepcionista se puede decir que la atención e inscripción de los pacientes se realiza con lentitud.

Actualmente en los departamentos de medios de diagnóstico el manejo de la información de los exámenes complementarios resulta complejo debido a la falta de un estándar nacional por el cual los laboratorios puedan regirse. Esto trae como consecuencia que la información sea tratada de distintas formas en los diferentes laboratorios del país, provocando que los reportes que se generen estén evaluados en diferentes unidades de medidas y el intervalo de referencia de algunos exámenes pueda variar.

Además, la información en estos momentos se encuentra en formato duro resultando engorroso para el médico enviarla a determinados centros, especialistas o a otros locales dentro del mismo policlínico, lo que trae consigo que esta pueda deteriorarse debido a su excesiva manipulación. Esto también puede provocar pérdida de la información afectando tanto al paciente como al médico que se vería en la obligación de volver a realizar los exámenes, lo que provocaría gastos adicionales de recursos, que serían innecesarios, pues muchas veces hay falta de materiales que imposibilitan la realización de análisis a la población. En la actualidad el país no cuenta con un sistema informático que gestione la información necesaria para los procesos de diagnóstico y que permita la integración con el sistema alasSIAPS.

Luego de analizar la anterior situación problemática se plantea el siguiente **Problema a Resolver** ¿Cómo facilitar la gestión de la información de los procesos que requieren exámenes complementarios en las instituciones de la Atención Primaria de Salud?

En función del problema científico identificado se determina como **Objeto de Estudio** el proceso de gestión de la información de los medios de diagnóstico en la Atención Primaria de Salud. El **Campo de Acción** abarcado es el proceso de gestión de la información utilizada por los exámenes complementarios en la Atención Primaria de Salud.

Como **Objetivo General** se plantea: Desarrollar el módulo de Medios de Diagnóstico del Sistema Integral para la Atención Primaria, que facilite la gestión de información relacionada con los exámenes complementarios de las instituciones de la Atención Primaria de Salud.

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

- ✓ Realizar análisis valorativo del estado del arte relacionado a la gestión de información de los Medios de Diagnóstico en la Atención Primaria de Salud.
- ✓ Identificar los procesos actuales relacionados con la gestión de los Medios de Diagnóstico en la Atención Primaria de Salud.
- ✓ Caracterizar la arquitectura definida por el departamento Atención Primaria de Salud.
- ✓ Realizar teniendo en cuenta las tecnologías y herramientas propuestas, los Flujos de Trabajo Modelado de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.
- ✓ Implementar las funcionalidades propuestas para gestionar los Medios de Diagnóstico en la Atención Primaria de Salud.

**Resultados esperados:** Con el modelado del negocio y la captura de los requisitos funcionales y no funcionales, se logrará el diseño de los prototipos no funcionales del sistema que permitirán una mejor implementación de los procesos de Laboratorio Clínico y Parasitología, Endoscopía, Drenaje Biliar, Electrocardiograma, Optometría y Laboratorio de Alergia asociados al área de Medios de diagnóstico en la Atención Primaria de Salud.

El contenido de este trabajo está estructurado por cuatro capítulos recogiendo toda la información utilizada en la investigación:

**CAPÍTULO 1. Fundamentación Teórica:** Contiene los aspectos esenciales para entender toda la información referente a los medios de diagnóstico. Se describen los conceptos fundamentales asociados a los medios de diagnóstico, se hace una valoración crítica de los sistemas informáticos utilizados en la actualidad en la salud que se refieren a los medios de diagnóstico en la APS, se hace un estudio de los lenguajes, tendencias, herramientas y tecnologías utilizadas para el diseño de aplicaciones web.

**CAPÍTULO 2. Características del Sistema:** Describe los principales procesos vinculados al problema a resolver, definiendo los procesos identificados a través del Modelamiento del Negocio que representa las necesidades del cliente, se realiza la especificación de los Requerimientos funcionales y no funcionales, se identifican los procesos que requieren informatización, se define de forma general una propuesta del sistema a través del modelamiento del sistema.

**CAPÍTULO 3. Diseño del Sistema:** Muestra los elementos básicos del diseño del sistema mediante los diagramas de clases del diseño, se hace una definición de los elementos del diseño, se explican algunos de los patrones de diseño y se hace una descripción detallada de una de las clase del diseño.

**CAPÍTULO 4. Implementación:** En este capítulo se muestran aspectos relacionados con la construcción del sistema, la justificación de la integración con otros módulos, se definen los componentes y se realiza el diagrama de despliegue como parte del modelo de implementación, se describen los estándares de diseño, codificación y tratamiento de errores utilizados en la realización de la aplicación web.

## Capítulo 1. Fundamentación Teórica

El objetivo fundamental de este capítulo es abordar distintos aspectos que se utilizan como soporte teórico del sistema diseñado. Se exponen a través de una descripción de los conceptos asociados al problema, de la estructura organizativa del Sistema Nacional de Salud (SNS), así como de los conceptos principales para comprender la gestión de la información de los medios de diagnóstico. Se describen brevemente algunos sistemas a nivel nacional e internacional que pueden servir como referencia a la hora de elaborar esta solución. Además de exponer las características de las herramientas, metodologías y tecnologías escogidas para la solución del problema, así como argumentar la elección de las mismas.

### 1.1 Conceptos Asociados

**Medios de Diagnósticos:** es un instrumento, procedimiento o conjunto de procedimientos a través de los cuales se llega a determinar el padecimiento de un paciente, clasificándose en diferentes especialidades diagnósticas. Por ejemplo Laboratorio Clínico, Endoscopia, Optometría, Electrocardiograma, etc.

**Laboratorio Clínico:** es un lugar físico que se encuentra especialmente equipado con diversos instrumentos y elementos de medida o equipo, en el que se llevan a cabo análisis clínicos que tienen como meta la prevención, el diagnóstico y el tratamiento de las enfermedades.

**Exámenes Complementarios:** son los diferentes análisis que se realizan en los medios de diagnóstico.

**Drenaje Biliar:** procedimiento que se usa para drenar la bilis y aliviar la presión que causa una obstrucción en los conductos biliares.

**Endoscopia:** especialidad que se encarga de diagnosticar padecimientos estomacales en pacientes.

**Optometría:** especialidad que se encarga de diagnosticar los padecimientos visuales de los pacientes.

**Electrocardiograma:** procedimiento que se usa para medir la actividad eléctrica del corazón en forma de cinta gráfica continua.

## 1.2 Sistema Nacional de Salud

El MINSAP es el organismo rector del SNS. Encargado de dirigir, ejecutar y controlar la aplicación de la política del estado y del gobierno en cuanto a la salud pública, el desarrollo de las Ciencias Médicas y la industria Médico Farmacéutica.

Desde el propio triunfo revolucionario se comenzó a trabajar por la creación del SNS que llevó la acción del trabajador de la salud a los lugares más apartados. El SNS tiene un carácter estatal y social, permitiendo una accesibilidad de forma gratuita a estos servicios que se extienden a todo lo largo del país, con una activa participación de la comunidad. También como parte de sus principios rectores se encuentra la colaboración internacional permitiendo extender estos servicios no sólo a los lugares más intrincados de la geografía nacional, sino a cualquier parte del mundo que lo necesite. El SNS se organiza en tres niveles de atención a la población: Atención Primaria, Atención Secundaria y Atención Terciaria.

- **Atención Primaria de Salud (APS):** Da solución aproximadamente al 80 % de los problemas de salud de la población y a lo que corresponda con las acciones de promoción, investigación, prevención, prescripción y protección de la salud en la comunidad. Sus actividades están relacionadas fundamentalmente con los Centros de Salud Familiar o de la Comunidad, Unidades Móviles, Unidades Rurales, así como en Centros Especializados en Atención Primaria de Salud (CEAPS) que cumplen objetivos concretos en la sociedad.
- **Atención Secundaria de Salud (ASS):** Este nivel da cobertura a cerca del 15 % de los problemas de salud. Se llevan a cabo acciones de salud más complejas y especialidades médicas a mayor profundidad que en el nivel anterior. Comprende la atención médica brindada en los diferentes tipos de instituciones hospitalarias entre ellas Hospitales Municipales, Generales y Especiales.
- **Atención Terciaria de Salud (ATS):** El nivel terciario debe abarcar alrededor del 5 % de los problemas de salud. Se brindan servicios de muy alta complejidad, con la óptima utilización de los recursos y medios existentes y el desarrollo de la investigación. A este nivel pertenecen los Institutos y Hospitales Especializados de Alta Tecnología.

## 1.3 Informatización del SNS

Durante los últimos años un grupo de instituciones cubanas y el propio MINSAP han desarrollado sistemas encaminados a lograr la informatización de la salud. En todos los casos el objetivo ha sido proveer al SNS de información confiable, consistente y oportuna para la toma de decisiones y el mejoramiento de los procesos médicos asistenciales, garantizando de esta manera el incremento en la calidad y seguridad de la atención médica a la población.

El SNS se traza una nueva estrategia para alcanzar la informatización del sector alineada al proceso de informatización de la sociedad cubana, partiendo de la APS y tomando como eje al policlínico y utilizando para su construcción las últimas tecnologías de la información y las comunicaciones para el desarrollo de aplicaciones orientadas a servicios empleando para este fin software libre. Se obtendrán soluciones con un alto nivel de integración e interoperabilidad que permitirán garantizar la información para la toma de decisiones y el mejoramiento continuo de la calidad de la atención médica cubana.

## 1.4 Sistema Integral de Atención Primaria de Salud

El Sistema Integral de Atención Primaria de Salud (alasiAPS) en sentido general abarcará 14 áreas claves de la Atención Primaria de Salud que serán implementadas. El sistema contará con dos subsistemas, un subsistema web y uno de escritorio, soportados por una arquitectura multiplataforma y en tres capas, el lenguaje de programación Java y el patrón Modelo-Vista-Controlador.

El subsistema web se concibe como una aplicación de Planificación-Control y Seguimiento de pacientes con un enfoque comunitario, la cual puede ser desplegada en cada uno de las unidades médicas de manera centralizada para un área específica o en una zona que posea un conjunto de entidades dirigidas bajo una misma jerarquía administrativa.

Se recomienda el sistema de escritorio para las entidades de escasa infraestructura tecnológica, de computadoras de pocas prestaciones, con ancho de banda reducidos y de conexión restringida. Se propone ser utilizado para unidades de pocas terminales.

## 1.5 Sistemas informáticos existentes

Con el desarrollo de las TIC (Tecnologías de la Información y las Comunicaciones) el uso de la informática como herramienta de ayuda a la medicina es una realidad en auge. En la actualidad existen varios sistemas informáticos dirigidos a la salud, de ellos un pequeño grupo se centran en la informatización de la gestión de la información relacionada con los medios de diagnóstico. A continuación se especifican algunos de estos sistemas:

### **Galen Lab**

Soluciones implementadas por la empresa cubana de desarrollo de software Softel. Está dirigido a la gestión de los medios de diagnóstico, facilitando la solicitud de exámenes, registro y evaluación de los resultados obtenidos, así como la generación de información estadística. Diseñado para ser utilizado por los técnicos, médicos y personal administrativo de medios de diagnóstico para optimizar el trabajo y elevar la eficiencia. Cuenta con un sistema de ayuda en línea que brinda al usuario toda la información que necesite sobre el proceso que está efectuando y garantiza un estricto control de la información, diferenciando para cada uno de ellos las opciones disponibles según su responsabilidad.[3]

Esta aplicación está desarrollada en visual Basic, los módulos que componen este producto están diseñados para ser ejecutados solamente en las estaciones de trabajo que tengan el sistema operativo Windows, como sistema gestor de bases de datos (SGBD) se utilizó el SQL Server. [4] Además por ser una aplicación de escritorio tiene la desventaja que debe ser instalado en cada una de las estaciones de trabajo para que pueda ser usado por el personal de salud que labora en los diferentes departamentos del laboratorio.

### **Medysistem Laboratorio**

Está diseñado para cumplir con el proceso diario que requieren los laboratorios de análisis clínicos. Brinda un servicio de excelencia ya que tendrá control desde que el paciente se presente en recepción hasta el momento que se le entreguen sus resultados, pasando por los respectivos departamentos del laboratorio. Le ayudará a tener un óptimo control administrativo, porque con el sistema es controlado los cortes de caja, expedición de facturas, entrega de resultados, reportes estadísticos. [5]

A pesar de estas características es un software propietario que contempla sólo el laboratorio, y no gestiona toda la información referente a los exámenes complementarios en los policlínicos.

## **XLab**

Es un sistema mexicano para el apoyo operacional y el control administrativo integral de un laboratorio de análisis clínicos. Proporciona una herramienta eficiente para el manejo de pacientes, sus resultados y la administración de las diversas funciones de un laboratorio. Cuenta con un acceso controlado para compartir la información de los pacientes así como para la transcripción de resultados, la facturación de los cargos del laboratorio y su cobranza en forma expedita. Incluye un control de inventarios para el manejo de los suministros utilizados dentro del servicio.

El sistema XLab tiene la capacidad de conectarse a un sistema hospitalario de forma automática o por medio de interfaces para acceder o actualizar la información de pacientes, el expediente clínico así como la información administrativa de facturación y cobranza. También hace posible la utilización de interfaces para la conexión y transferencia automática de información desde equipos clínicos automatizados. [6]

Esta aplicación tiene como desventaja que para poder funcionar debe estar instalado el sistema operativo Windows. Solamente gestiona la información del laboratorio, por lo que no brinda toda la información necesaria para la integración con otros servicios. Otra desventaja de este sistema es que no permite realizar reportes de la información gestionada en el laboratorio clínico y esto dificulta a los directivos de esta área la administración y el control.

## **VisionDat**

Es un servicio de software para mantener expedientes clínicos electrónicos. El sistema de control de consulta para profesionales de la visión se basa en modelos clínicos de uso internacional, que han sido analizados y plasmados en un diseño simple para el uso de todos aquellos que se dedican al cuidado visual. El software le permite registrar:

- Datos generales
- Estudio socioeconómico
- Exámenes preliminares
- Historia clínica
- Estudio refractivo
- Estudio oftalmológico

Este sistema ofrece sus servicios a través de las ventas de Unidades de Uso (UU). Cada unidad corresponde a una consulta completa de un paciente en una fecha determinada. Cada UU cuesta 0.25 centavos US Dlls. y la compra mínima que puede hacer como suscriptor es de 100 UU, es decir, \$25.00 US Dlls.

El suscriptor es el propietario de la base de datos que se obtiene a través del servicio de software. Es también quien administra el sistema y realiza la renovación del servicio a través de la adquisición de las UU y los usuarios son los individuos que están autorizados para acceder a VisionDat Software, en su consultorio u oficina; son los auxiliares de consultorio, el profesional que realiza la consulta y aquellos colaboradores que cada suscriptor decida autorizar para capturar y acceder a los datos. Existen niveles de permiso para cada usuario que el suscriptor define. [7]

### **Neomedic Endoscopía**

Este programa permite capturar imágenes y video directamente del endoscopio: Pentax, Olympus, etc. Imprime reportes con esquemas y permite seleccionar entre múltiples formatos antes de imprimir. Neomedic maneja un expediente por paciente, posee búsquedas de pacientes muy sencillas pero poderosas. Funciona en red y controla el acceso para múltiples usuarios. [8]

### **Requisitos de instalación Indispensables:**

- Pentium 4 a 1 GHz o superior
- 1 GB de memoria RAM
- Monitor VGA o superior, res. mín. 1024 x 768 píxeles y color de alta densidad
- Microsoft Windows® 7, Vista o XP

Según los requisitos de hardware de este sistema, implantarlo en Cuba en estos momentos sería difícil debido a la situación económica actual, que a pesar de destinar gran cantidad de recursos a la informatización, se hace imposible contar con ordenadores que posean estas propiedades o cualidades en todos los policlínicos del país, sin embargo, este software es uno de los más completos en la gestión de la información en el área de endoscopía, contando con varios servicios como son: capturar imágenes, video, editar reportes, etc. que pueden ser implementados en la solución para versiones posteriores, por lo que constituye uno de los sistemas a tener en cuenta en el desarrollo del producto.

En el área de electrocardiograma actualmente los sistemas están orientados a almacenar e interpretar las imágenes, no a la gestión de la información asociada a la actividad de realizar un electrocardiograma, como es la información del paciente y la de los resultados.

De acuerdo con el estudio realizado no existen actualmente sistemas informáticos que recojan la información relacionada con todos los exámenes complementarios realizados en la APS. Estos sistemas gestionan información de un área de los medios de diagnóstico en particular, por lo que no resuelven el problema planteado y en otras áreas como drenaje biliar y laboratorio de alergia no se conocen software que viabilicen las principales actividades realizadas por el personal de salud que en ellas laboran. Además los sistemas anteriormente analizados, con la excepción del Galen Lab, son propietarios y para poder adquirirlos hay que pagar una licencia, provocando gastos innecesarios a la economía del país.

## **1.6 Tecnologías y herramientas utilizadas**

Las herramientas y tecnologías fueron definidas por el Centro de Informática Médica (CESIM) para el desarrollo de aplicaciones en el departamento de Atención Primaria. En esta sección se mencionan algunas de estas tecnologías, metodologías y herramientas utilizadas en el desarrollo del módulo Medios de diagnóstico del SIAPS, así como algunas de sus características más significativas.

### **1.6.1 Proceso de Desarrollo**

Un buen proceso de desarrollo de software puede reducir costos y retrasos de proyectos, así como mejorar la calidad del software. La metodología de desarrollo cobra gran importancia en proyectos empresariales, pues al no utilizarla adecuadamente se puede desembocar en la frustración del equipo de desarrollo y en la insatisfacción de los clientes. Por tanto, el uso de una metodología es necesario para controlar el ciclo de vida de un proyecto.

Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas CMMI se publican en documentos llamados modelos, los cuales contienen el conjunto de prácticas relacionadas que son ejecutadas de forma conjunta para conseguir determinados objetivos. Así es como el modelo CMM establece una medida del progreso, conforme al avance en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El alcanzar estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables.

CMMI consta de 22 áreas de proceso distribuidas dentro de 5 niveles de madurez:

- Nivel 1: Inicial.
- Nivel 2: Administrado.
- Nivel 3: Definido.
- Nivel 4: Cuantitativamente administrado.
- Nivel 5: Optimizado.

Las organizaciones son evaluadas y reciben una calificación de nivel 1-5 siguiendo los niveles de madurez. Este enfoque se denomina “Representación Escalonada”. La Universidad de las Ciencias Informáticas se encuentra desarrollando un proceso de mejora con el objetivo de obtener el Nivel 2 de CMMI. Las áreas de proceso que lo forman son:

- Administración de Requisitos (REQM).
- Planeación del Proyecto (PP).
- Monitoreo y Control del Proyecto (PMC).
- Medición y Análisis (MA).
- Aseguramiento de la Calidad de Procesos y Productos (PPQA).
- Administración de la Configuración (CM).
- Administración de Acuerdos con Proveedores (SAM).

El objetivo de la Administración de Requisitos es gestionar los requisitos de los elementos del proyecto y sus componentes e identificar inconsistencias entre estos requisitos, el plan de proyecto y los elementos de trabajo. En este proceso se deben gestionar todos los requisitos del proyecto, tanto técnicos como no técnicos. Estos requisitos han de ser revisados conjuntamente con la fuente de los mismos así como con las personas que se encargarán del desarrollo posterior. Para llevar a cabo estas actividades es utilizado el documento IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos realizado por la universidad y cuyo objetivo es definir el proceso de administración de requisitos.

Este documento establece el ciclo de vida a seguir asociado a los proyectos involucrados en el proceso de mejora, el cual consta de 9 fases y se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico del área Administración de Requisitos. La descripción del libro IPP-3510 incluye la definición de roles, sus responsabilidades y las habilidades en la ejecución de las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución

de dichas actividades. La información contemplada en cada producto típico de trabajo está registrada según lo definido en la metodología Proceso Unificado de Desarrollo de Software (RUP).

El RUP es una metodología de desarrollo de software orientada a objetos que proporciona un método disciplinado para asignar las tareas y responsabilidades dentro del equipo de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que resuelva las necesidades del usuario dentro de un cronograma predecible y al menor costo posible. Se caracteriza por ser iterativo e incremental, centrado en la arquitectura y guiado por casos de uso y dividir el proceso en ciclos de desarrollo que se agrupan en fases en las cuales las actividades se distribuyen entre 9 flujos de trabajo. Cada fase finaliza con un hito donde se debe tomar una decisión importante.

Para la descripción de los productos de trabajo resultantes de cada uno de los flujos de trabajo de RUP así como de las fases del ciclo vida definido en el libro anteriormente mencionado se utiliza el Lenguaje Unificado de Modelado (UML), con la especificación del uso del estándar de modelado de procesos de negocio (BPMN) para la descripción de los procesos de negocio.

### **BPMN (Business Process Management Notation)**

Es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas de los procesos. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes, además de proveer a los usuarios de una notación libre. Es fácil, legible y entendible por todos los involucrados del negocio.

El objetivo principal de BPMN es brindar una notación rápidamente comprensible por todo el personal implicado en el negocio. BPMN proporciona a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica, facilitando a las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar.

Sus principales objetivos son:

- Proveer una notación que sea fácilmente entendida por todos los usuarios, desde el analista de negocio, el desarrollador técnico y hasta el propio personal del negocio.
- Asegurar que los lenguajes para la ejecución de los procesos de negocio puedan ser visualizados con una notación común.

- Comunicar una amplia variedad de información a una amplia variedad de audiencias.

## **Lenguaje Unificado de Modelado**

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. El modelo gráfico de UML tiene un vocabulario que identifica los elementos, las relaciones, y los diagramas. No define un proceso de desarrollo específico, sólo se trata de una notación. A través de UML, se unifican las mejores técnicas para modelar y documentar los elementos de un sistema de software. Ofrece diferentes diagramas que son utilizados para visualizar los distintos aspectos del sistema.

### **1.6.2 Estilos Arquitectónicos**

La programación para muchos es considerada un arte, pero en sus inicios se tornaba un poco difícil para la mayoría de las personas, pero con el tiempo se han ido descubriendo y desarrollando formas, guías generales y estilos de programación, con el objetivo de resolver aquellos problemas. A este conjunto de patrones y abstracciones presentes en la programación de hoy, que guían la construcción del software se le conoce como Arquitectura de Software y establece los fundamentos para que todo el personal implicado en el desarrollo del sistema, trabaje en una línea común que permita alcanzar los objetivos propuestos y satisfacer las expectativas del cliente.

#### **Arquitectura Cliente/Servidor.**

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Características del Cliente:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Características del Servidor:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación.
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.

## Arquitectura en tres capas

La utilización de la plataforma Java Enterprise Edition (JEE 5.0) permite una concepción del sistema en n-capas, en específico se usará una arquitectura en tres capas, cuyo objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles.

**Capa de presentación:** es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

**Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

**Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

## Patrón Modelo-Vista-Controlador (MVC)

El patrón MVC separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño donde se mezclan los componentes de interfaz y

de negocio, entonces la consecuencia será que, cuando se necesite cambiar la interfaz, se tendrá que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error. Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. [10]

## 1.6.2 Tecnologías usadas en la capa de Presentación

### Java Server Faces (JSF)

La tecnología Java Server Faces es un marco de desarrollo de los componentes de la interfaz de usuario, válido para todas aquellas aplicaciones web basadas en la tecnología Java.

### Richfaces

RichFaces es un marco de código abierto que añade a las aplicaciones capacidad de Ajax en JSF, sin recurrir a Java Script. RichFaces aprovecha el framework Java Server Faces, incluyendo su ciclo de vida, la validación, los medios de conversión y la gestión de los recursos estáticos y dinámicos. Los componentes de RichFaces con soporte Ajax y aspecto altamente personalizable pueden ser fácilmente incorporados a aplicaciones JSF.

### Ajax

Es el acrónimo para Asynchronous Java Script + XML y el concepto es: Cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en *background*, los datos que son usados para actualizar la página sólo re-renderizando la página y mostrando u ocultando porciones de la misma. [13]

AJAX incorpora:

Presentación basada en estándares usando XHTML y CSS.

- Exhibición e interacción dinámicas usando el Document Object Model.
- Intercambio y manipulación de datos usando XML y XSLT.
- Recuperación de datos asincrónica usando XML Http Request.

### Ajax4Jsf

Es una librería open source 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 Java Script. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control

de cualquier evento de usuario, etc. En definitiva Ajax4jsf permite dotar a la aplicación JSF de contenido mucho más profesional con muy poco esfuerzo. [14]

## 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. Permite que JSP (Java Server Pages) y JSF (Java Server Faces) puedan funcionar conjuntamente en una misma aplicación web. Estos no se complementan naturalmente. JSP procesa los elementos de la página de arriba a abajo, mientras que JSF dicta su propio re-rendering (ya que su ciclo de vida está dividido en fases marcadas). Facelets llena este vacío entre JSP y JSF, siendo una tecnología centrada en crear árboles de componentes y estar relacionado con el complejo ciclo de vida JSF.

Las principales ventajas de Facelets son:

Construcción de interfaces basadas en plantillas.

- Rápida creación de componentes por composición.
- Fácil creación de funciones y librerías de componentes.
- Facelets provee un proceso de compilación más rápido que JSP.
- Provee templating, lo cual implica reutilización de código, simplificación de desarrollo y facilidad en el mantenimiento de grandes aplicaciones.

## Lenguaje de Marcado de Hipertexto Extensible (XHTML)

Es una versión más estricta y limpia de HTML, que nace con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. [15]

## Extensible Markup Language (XML)

No es sólo un lenguaje, es una forma de especificar lenguajes, de ahí el término de extensible. Es un lenguaje de etiquetas no predefinidas previamente, es decir, el programador es el que las crea en cada caso. El XML ahorra tiempos de desarrollo y proporciona ventajas, dotando a webs y a aplicaciones de una forma realmente potente de guardar la información. Se ha convertido en un formato universal para el

intercambio de información estructurada entre diferentes plataformas. En la actualidad permite la compatibilidad entre sistemas para compartir la información.

## **Java Script**

Es un lenguaje de programación interpretado, es decir, no requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. Tiene como principal característica ser un lenguaje independiente de la plataforma.

### **1.6.3 Tecnologías usadas en la capa de Negocio**

#### **Servidor de Aplicaciones**

JBoss es un servidor de aplicaciones para Java. Es muy reconocido por ser de los primeros servidores de aplicación empresarial gratuito y open source. Al estar basado en Java, puede ser utilizado en cualquier sistema operativo que lo soporte, ofreciendo una plataforma de alto rendimiento para aplicaciones java, aplicaciones Web y Portales.

#### **Enterprise Java Beans (EJB)**

Es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando el lenguaje de programación java. Permite a los desarrolladores enfocarse en la lógica del negocio sin tener que emplear tiempo en la conformación del código de la infraestructura. Desde el punto de vista del desarrollador es un pedazo de código java que se ejecuta en un entorno especializado denominado contenedor EJB que proporciona un conjunto de servicios. El servicio de persistencia es proporcionado por un framework especializado denominado proveedor de persistencia.

#### **Plain Old Java Object (POJO's)**

Son las siglas utilizadas por programadores java para enfatizar el uso de clases simples y que no dependen de un framework en especial.

## 1.6.5 Tecnologías usadas en la capa de Acceso a Datos

### Java Persistence API (JPA)

El Java Persistence API fue desarrollado por el grupo de expertos de EJB 3.0, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes; incluso fuera de la plataforma Java EE.

En su definición, se han combinado ideas y conceptos de los principales frameworks de persistencia como Hibernate, Toplink y JDO y de las versiones anteriores de EJB. Todos estos cuentan actualmente con una implementación JPA.

### Framework Hibernate

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada se puede generar bases de datos en cualquiera de los entornos soportados: Oracle, DB2, MySql, etc. Y lo más importante de todo, es open source, lo que supone, entre otras cosas, que no se tiene que pagar nada por adquirirlo.

Hibernate ofrece también un lenguaje de consulta de datos llamado **HQL** (*Hibernate Query Language*). Puede ser usado para desarrollar y distribuir aplicaciones de forma gratuita. Es utilizado en las aplicaciones Java EE.

### JBoss Seam

JBoss Seam es una nueva y poderosa aplicación para la construcción de la próxima generación de aplicaciones Web 2.0 de la unificación y la integración de tecnologías, como Java Script asíncrono y XML (AJAX), Java Server Faces (JSF), Enterprise Java Beans (EJB3), Java Portlets y Business Process Management (BPM).

Es un framework que integra y unifica los distintos estándares de la plataforma Java EE, pudiendo trabajar con todos ellos siguiendo el mismo modelo de programación. Ha sido diseñado intentado simplificar al máximo el desarrollo de aplicaciones, basando el diseño en POJOs con anotaciones. Estos componentes se usan desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa. El núcleo principal de Seam está formado por las especificaciones EJB3 y JSF.

## 1.6.6 Lenguaje de Programación

El lenguaje de programación seleccionado fue Java, este es un lenguaje orientado a objetos desarrollado por Sun Microsystems. La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar. [16]

Es distribuido, multiplataforma, compilado, seguro y posee una arquitectura neutral. Proporciona un conjunto de clases potente y flexible. Elimina muchas de las características de C y C++, lo que lo hace más manejable desde el punto de vista del programador.

## 1.6.7 Herramientas a utilizar

### Visual Paradigm for UML 6.4 Enterprise Edition

El resto de los artefactos: modelo conceptual, diagrama de clases del diseño, modelo de datos, modelo de despliegue y los prototipos de interfaz serán modelados aplicando UML y utilizando la herramienta case Visual Paradigm for UML 6.4 Enterprise Edition. Es válido aclarar que se utiliza la versión 6.4 pues incorpora dentro de las funcionalidades que brinda para el modelado el user interface, el cual permite realizar un esbozo de los prototipos de interfaz de usuario, logrando así una mejor comprensión del sistema que se desea desarrollar, además facilita el trabajo de los diseñadores de interfaces de usuarios y el de los implementadores.

### Eclipse Ganymede

Para el desarrollo del software se utilizará el Eclipse, que es un entorno integrado (IDE) para desarrollo de aplicaciones con java. Está soportado por IBM, es un proyecto open source, multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Se está convirtiendo en el estándar de facto de los entornos de desarrollo para Java. Y es que Eclipse no es tan sólo un IDE, se trata de un marco de trabajo modular ampliable mediante complementos (*plugins*). De hecho, existen complementos que permite usar Eclipse para programar en PHP, Perl, Python, C/C++, etc.

### PostgreSQL Server 8.3.

Como gestor de base de datos se utiliza PostgreSQL, este es el gestor más avanzado de código abierto en la actualidad, ofrece servicios de control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando

también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl y Python). Es extensible, brinda gran estabilidad y confiabilidad y es multiplataforma. Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/PGSQL. Este lenguaje es comparable al lenguaje procedural del sistema de gestión de base de datos relacional Oracle, PL/SQL. Una de sus ventajas es la excelente documentación que existe, además de contar con una comunidad de usuarios y desarrolladores a los que acudir en caso de tener problemas.

En este capítulo para lograr la comprensión de los procesos identificados y un mejor entendimiento del trabajo en general se definió qué son los medios de diagnóstico, así como los conceptos de electrocardiograma, drenaje biliar, endoscopia, laboratorio clínico, entre otros.

Se realizó un estudio de los principales sistemas informáticos dirigidos a los Medios de Diagnóstico, a nivel nacional se analizó el Galen Lab que es un sistema para la gestión de los medios de diagnóstico y en el ámbito internacional se estudiaron algunas aplicaciones dirigidas a la gestión de información en algunas de las áreas de los medios de diagnóstico, como es el caso del Xlab, VisionDat, Neomedic Endoscopia, entre otros, estos sistemas son propietarios y monoplataformas y gestionan información de un área de los medios de diagnóstico en particular, llegando a la conclusión que los mismos no resuelven el problema planteado.

Además, se abordó sobre las tecnologías y lenguajes usados con la tecnología java, como el framework JSF, Facelets y RichFaces, se hizo referencia a la guía utilizada para el desarrollo del software la cual está basada en el libro IPP:3510 y se mencionaron las herramientas que serán utilizadas a lo largo del desarrollo del sistema propuesto.

## Capítulo 2. Características del Sistema

En el presente capítulo se describen las características del sistema a construir y se detallan los principales procesos del negocio identificados en el área de Medios de diagnóstico. De los procesos se obtendrán los documentos que contienen la información gestionada para el correcto funcionamiento de esta área, algunos de ellos son los modelos del negocio y los diagramas correspondientes, describiéndose los actores y trabajadores que intervienen en los mismos, además se podrá tener una visión del sistema a partir de las funcionalidades requeridas y las restricciones que se imponen.

### 2.1 Flujo actual de los procesos

Los medios de diagnóstico son usados para diagnosticar diferentes patologías y para realizar estudios que determinen el tipo de tratamiento que se le debe administrar a un paciente, al igual que el seguimiento del mismo. Están formados por diferentes técnicas, por ejemplo la endoscopía que es la encargada de diagnosticar pacientes con enfermedades digestivas, el drenaje biliar que es un procedimiento terapéutico para el tratamiento de la obstrucción de la vía biliar, el electrocardiograma que es el encargado de diagnosticar enfermedades cardiovasculares, entre otras técnicas.

Cuando el médico consulta al paciente determina si es necesario orientarle algún examen, en caso afirmativo, el paciente debe dirigirse al departamento del policlínico donde se realizan los mismos. Estos departamentos pertenecen a una de las áreas que componen los medios de diagnóstico en la APS, en todas las áreas se realizan diferentes actividades para lograr el correcto funcionamiento de las mismas, algunas de ellas son la recepción, procesamiento de las muestras y la entrega de resultados. A continuación se describen los procesos relacionados con la investigación y la información que cada uno maneja.

#### 2.1.1 Proceso realizar análisis de laboratorio clínico

Este proceso se encarga de realizar los análisis orientados previamente a los pacientes. Desde que el paciente llega a recepción es atendido por la recepcionista, quien una vez que consultó la orden de análisis le recoge sus datos personales. De esta forma el paciente pasa a la lista de trabajo con el tipo de análisis que se realizará, el nombre del mismo y el tipo de muestra necesaria para dicho examen. En caso necesario el paciente se toma las muestras necesarias para que estas sean analizadas por el personal

calificado para desarrollar esta actividad y en el caso de que sea para el departamento de heces fecales y orina entrega las muestras. Después de la obtención de los resultados estos tienen que ser archivados y dados a conocer al paciente. El personal del laboratorio en la descripción de los resultados tiene que hacerle saber a las personas si este se encuentra en estado normal, alterado o en caso que la muestra se deteriore hay que comunicar la repetición del análisis.

### **2.1.2 Proceso realizar prueba cutánea**

Cuando el paciente llega a la consulta de alergia, la doctora alergista le recoge sus datos personales e interroga al paciente para conocer sobre sus síntomas y padecimientos, también ella debe saber si el paciente ha estado otras veces en esa consulta. De acuerdo al interrogatorio realizado con anterioridad la doctora llega a un posible diagnóstico pero orienta las pruebas necesarias para una total seguridad, estas son prueba de puntura, de parche, de escarificación y de reacción intradérmica. Después de realizadas estas el alergista analiza los resultados para poner un tratamiento médico y le orienta cuándo será el próximo encuentro para dar seguimiento a la enfermedad. Por último los resultados, el diagnóstico y el tratamiento orientado son archivados en el libro de control de pacientes.

### **2.1.3 Proceso realizar endoscopia**

Cuando el paciente acude a la consulta el médico realiza un interrogatorio acerca de los síntomas que presenta para conocer su estado de salud. El médico registra esos datos en un informe, le realiza la endoscopia al paciente y según lo que él observe determina si es necesario extraer una muestra y emitir una orden de frotis, en caso negativo registra los resultados en la libreta de control de pacientes.

En caso necesario esta muestra es preparada por la enfermera para ser analizada en el microscopio y si lo que se orienta es una biopsia dicha muestra es llevada al hospital. Si fue necesario una orden de frotis el resultado se le da a conocer al médico en esta misma orden y después él es quien da las conclusiones del diagnóstico y archiva los resultados.

### **2.1.4 Proceso realizar drenaje biliar**

El médico emite una orden para realizar el drenaje biliar. La enfermera registra los datos del paciente en el libro de control de turnos. Cuando se le realiza el drenaje biliar al paciente, se obtiene una muestra de bilis que es analizada detalladamente y el resultado se escribe en la remisión del paciente y se archiva en el

libro de la enfermera. El paciente es el encargado de recoger este resultado para que el médico lo analice y le oriente el tratamiento adecuado.

## **2.1.5 Proceso realizar electrocardiograma**

El médico recoge algunos datos del paciente y emite una orden para realizar el ECG. La enfermera también registra los datos del paciente en su libro y ella es la encargada de realizarle el electrocardiograma al paciente. El resultado se imprime y es el médico quien lo analiza y le explica al paciente el porqué de sus síntomas.

## **2.1.6 Proceso realizar prueba de refracción**

El paciente llega a la consulta de optometría con una orden médica, el técnico recoge sus datos y le realiza la refracción. Si el paciente es menor de 38 años se le da turno para la próxima semana para volver a realizarle las pruebas y luego registrarle los resultados, en caso contrario se registran los resultados del paciente el mismo día que se le realiza la prueba.

## **2.2 Información que se maneja**

A través de la investigación de los procesos anteriormente descritos, se identificaron una serie de documentos que recogen la información necesaria para la correcta realización de las actividades realizadas en estos procesos, a continuación se especifican:

Hoja de cargo: Archiva datos personales del paciente.

Libreta de control de paciente: Recoge información personal del paciente, el diagnóstico y el tratamiento.

Lista de trabajo: Datos del paciente y los análisis que se va a realizar.

Hoja de control de resultados del laboratorio: Recoge los resultados de los análisis indicados a los pacientes.

Orden médica: Es utilizada para registrar los resultados de los exámenes.

## **2.3 Objeto de Informatización**

A partir de los procesos anteriormente mencionados se propone informatizar la gestión de la información relacionada con el registro de resultados y la recepción de los pacientes por ser las actividades más

significativas de acuerdo al levantamiento de procesos y teniendo en cuenta que son claves para el mejoramiento de la atención al paciente, permitiendo prestar un servicio mucho más rápido.

El proceso relacionado con la recepción incluye la búsqueda de los pacientes y una vez encontrados permite la inscripción para ser atendidos en el departamento de medios de diagnóstico asignado de acuerdo a los exámenes orientados y el registro de resultados de los exámenes permite buscar un paciente para registrar los resultados de los exámenes orientados al mismo. Además, el sistema implementado le permitirá al médico orientar los exámenes que se están realizando en ese momento en la institución, así como consultar los resultados de los exámenes asignados a un paciente para llegar a un diagnóstico, de esta forma se logrará un mejor desempeño del personal vinculado con algunas de las actividades que se realizan en las instituciones de salud.

## 2.4 Modelado del negocio

El modelado del negocio es una técnica para comprender los procesos de negocio de la organización y la determinación de los requisitos del futuro sistema. Este es una de las fases del ciclo de vida utilizado para el desarrollo del software, el cual determina los procesos del negocio, identificando quiénes participan y las actividades que requieren informatización. El modelado de negocio a través de procesos aporta una visibilidad de las actividades que se realizan en los mismos, lo que permite una mayor flexibilidad y agilidad para la adaptación al cambio. Al conocer los procesos de negocio se puede integrar la información en un solo sistema para obtener un mejor flujo de información entre departamentos o áreas.

La notación utilizada para el modelado del negocio fue BPMN, esta ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes, además de proveer a los usuarios de una notación libre. Es fácil, legible y entendible por todos los involucrados del negocio.

### 2.4.1 Actores involucrados en el proceso de negocio.

Roles	Funciones
Paciente	Es la persona a la cual se le brindan los principales servicios del área Medios de diagnóstico, es el más beneficiado con los

	resultados del negocio.
Recepcionista	Es el encargado de atender a los pacientes que asisten al laboratorio clínico.
Técnicos de laboratorio	Son los encargados de realizar las extracciones de sangre, analizar las muestras e informarles el resultado a los pacientes.
Gastroenterólogo	Encargado de registrar los datos del paciente, realizarle la endoscopia, emitir una orden de frotis en caso necesario y registrar los resultados del paciente.
Enfermera de drenaje biliar	Verifica la remisión del paciente, le toma los datos al paciente en el registro de control, le realiza el drenaje biliar al paciente y registra los resultados del drenaje biliar.
Técnico optometrista	Es el encargado de verificar la remisión del paciente, tomarle los datos, realizarle la refracción y registrar los resultados del paciente.
Alergista	Es el encargado de inscribir al paciente en el libro de control de pacientes, le realiza las pruebas necesarias, analiza los resultados, le pone un tratamiento al paciente y registra los resultados.
Enfermera	Verifica la remisión del médico, registra los datos en el libro del paciente, le realiza el electrocardiograma al paciente, registra los resultados del paciente.

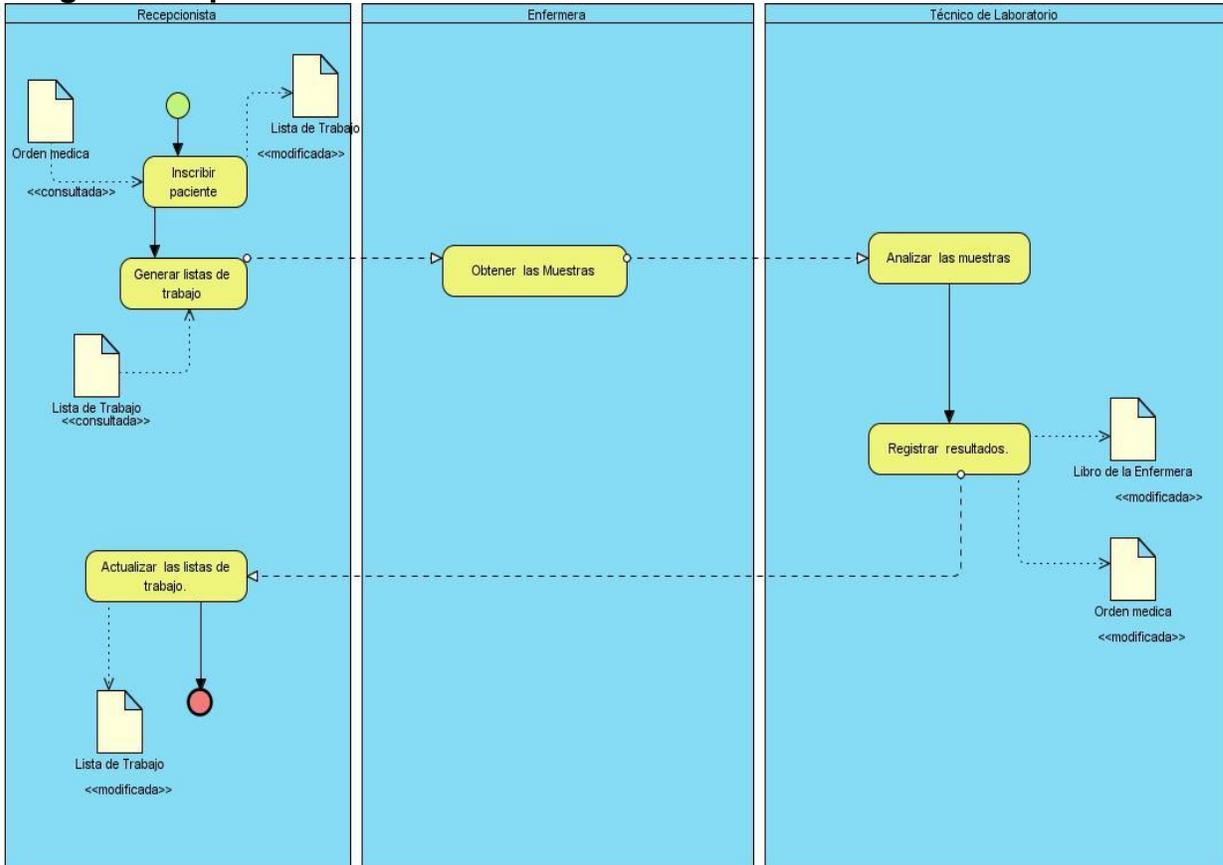
**Tabla 2.4.1 – Actores involucrados en el proceso de negocio.**

### 2.4.2 Procesos generales del negocio.

A través del modelado de los procesos se definieron las siguientes áreas con sus respectivos procesos.

1. Laboratorio Clínico y Parasitología
  - ❖ Realizar análisis de laboratorio clínico.
2. Laboratorio de Alergia
  - ❖ Realizar prueba cutánea.
3. Endoscopía
  - ❖ Realizar endoscopia.
4. Drenaje Biliar.
  - ❖ Realizar drenaje biliar.
5. Electrocardiograma
  - ❖ Realizar electrocardiograma.
6. Optometría
  - ❖ Realizar consulta de optometría.

## 2.4.3 Diagrama de proceso de realizar análisis de laboratorio clínico



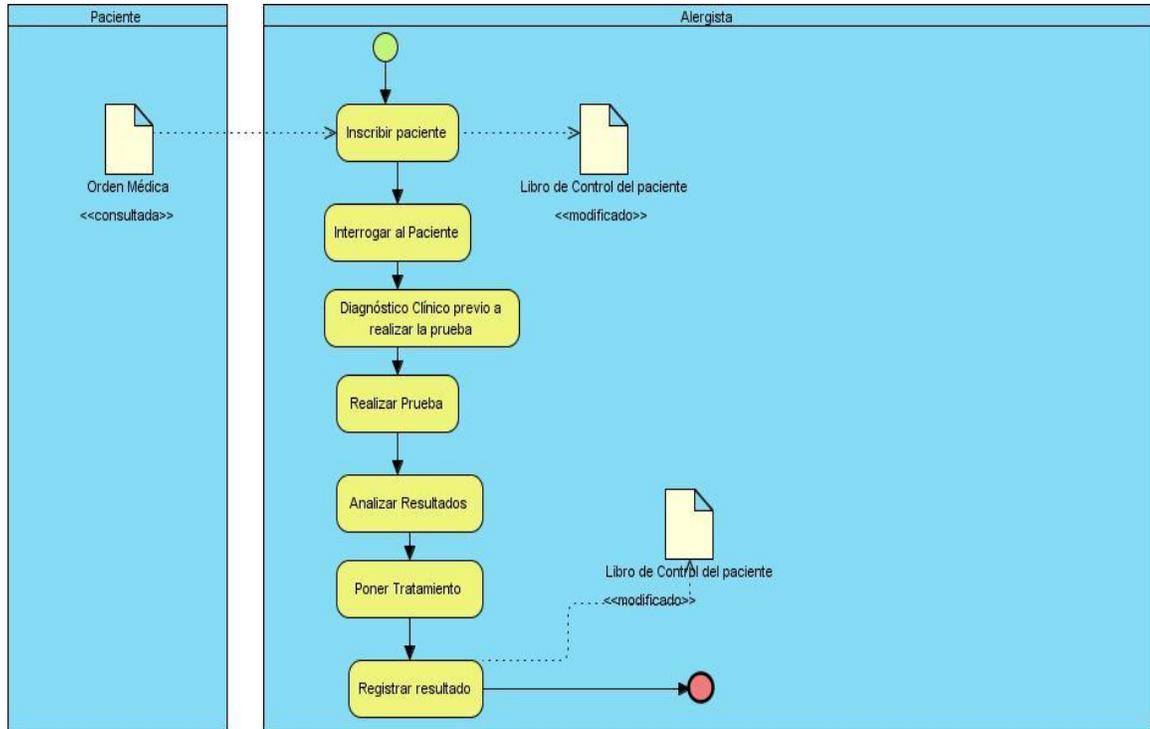
### 2.4.3.1 Descripción del proceso

Nombre:	Realizar análisis de laboratorio clínico.
Objetivos:	Elevar la calidad y eficiencia de los procesos en el laboratorio clínico y correcta información de sus resultados con el fin de apoyar los servicios clínicos en el diagnóstico, tratamiento y seguimiento de pacientes y al control epidemiológico y a la salud pública.
Evento(s) que lo generan:	-Que el paciente acuda con una orden médica a realizarse los análisis.
Precondiciones:	-Presentar orden del médico para que se realice los análisis -Cumplir con las condiciones físicas necesarias para la realización del análisis, de acuerdo a su tipo (por ejemplo: estar en ayuna, tomar la primera orina del día, etc.)

Poscondiciones:	-Quedan registrados en la lista de trabajo todos los resultados de los análisis de cada paciente de acuerdo al código que se le fue asignado.
Reglas de Negocio:	No procede.
Responsables:	Técnico de laboratorio.
Cientes internos:	CESIM.
Cientes externos:	Pacientes.
Entradas:	-Datos del paciente. -Orden médica
Salidas:	-Lista de trabajo. -Libro de la enfermera.
Actividades:	Act1 Inscribir paciente. Act2 Generar listas de trabajo. Act3 Obtener las muestras. Act4 Analizar las muestras. Act5 Registrar resultados. Act6 Actualizar las listas de trabajo.

**Tabla 2.4.3.1 Descripción del proceso realizar análisis**

## 2.4.4 Diagrama de proceso realizar prueba cutánea.



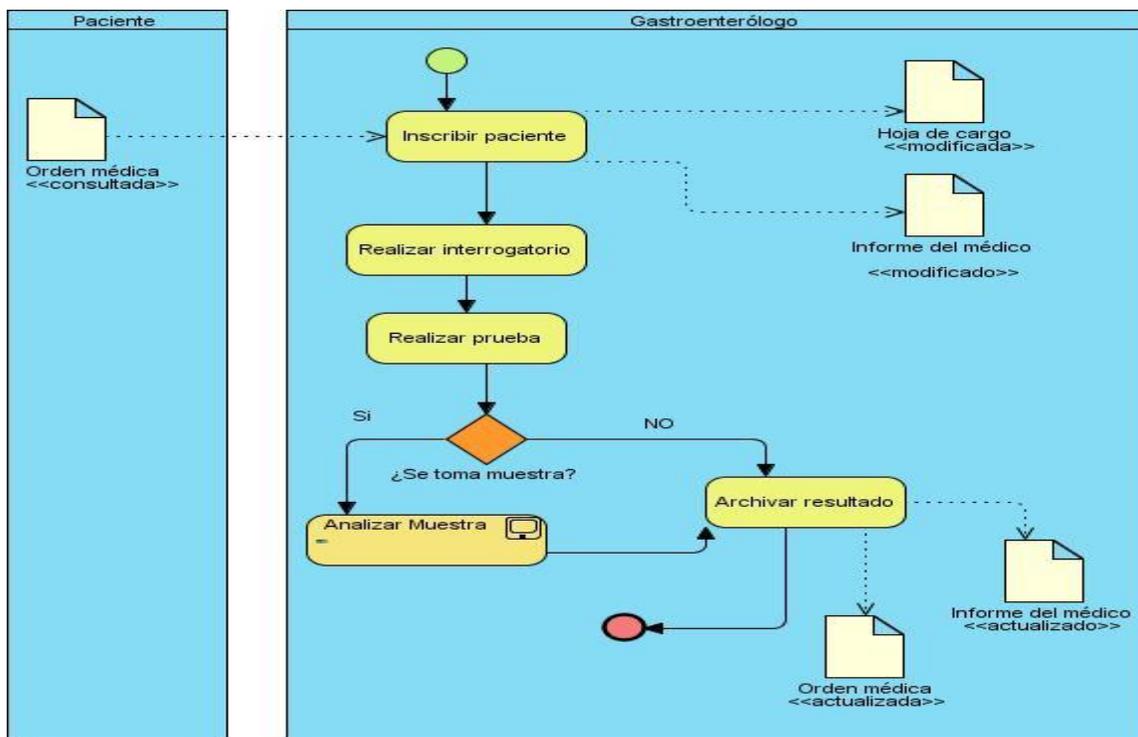
### 2.4.4.1 Descripción del proceso

Nombre:	Realizar prueba cutánea.
Objetivos:	Diagnosticar enfermedades alérgicas como asma, urticaria, alergia a los alimentos y dermatitis de contacto. Determinar el tipo de alergia en los pacientes aeroalérgicos (rinitis, asma, dermatitis atópica), alimentos (dermatitis atópica, anafilaxis, urticaria aguda o de contacto) o reacciones a medicamentos, venenos (anafilaxis por picadura de insectos).
Evento(s) que lo generan:	Que el paciente acuda con una orden médica.
Precondiciones:	-Que el paciente acuda con las condiciones físicas necesarias (estar en ayuna) y que presente la orden médica.
Poscondiciones:	-Quedan registrados los resultados de las pruebas.
Reglas de Negocio:	No procede.

Responsables:	Alergista
Cientes internos:	CESIM
Cientes externos:	Paciente
Entradas:	-Orden médica.
Salidas:	- Libro de control del paciente
Actividades:	Act1 Inscribir paciente. Act2 Interrogar al paciente. Act3 Diagnóstico clínico previo a realizar la prueba. Act4 Realizar prueba al paciente. Act5 Analizar resultados. Act6 Poner tratamiento. Act7 Registrar resultado.

**Tabla 2.4.4.1 Descripción del proceso realizar prueba cutánea.**

### 2.4.5 Diagrama de proceso de realizar endoscopia



### 2.4.5.1 Descripción del proceso

Nombre:	Realizar endoscopia
Objetivos:	Diagnosticar padecimientos estomacales en pacientes que presentan o no una trayectoria con enfermedades de este tipo. Descartar úlceras pépticas, gastritis, neoplasia, inflamaciones en las paredes del estómago, gastritis hemorrágicas, malformaciones a través del tubo digestivo hasta el duodeno.
Evento(s) que lo generan:	Que el paciente acude con una orden médica.
Precondiciones:	-Presentar orden del médico para que se realice la endoscopia -Cumplir con las condiciones físicas necesarias para la realización del análisis, de acuerdo a su tipo (por ejemplo: estar en ayuna, no ingerir alimentos después de las 6.00pm,etc)
Poscondiciones:	Quedan registrado los resultados de la endoscopia en el informe del médico y en la remisión
Reglas de Negocio:	No procede.
Responsables:	Técnico de laboratorio
Clientes internos:	CESIM
Clientes externos:	Paciente
Entradas:	-orden médica
Salidas:	-Hoja de cargo (Nombre, Edad, dato clínico, Fecha, firma, descripción). - Informe del médico (Nombre, Edad, dato clínico, Fecha, firma, descripción de lo que encontró en el estómago, las conclusiones y el tratamiento a seguir). - orden médica
Actividades:	Act 1 Inscribir paciente. Act2 Realizar interrogatorio. Act3 Realizar prueba. Sub1 Analizar muestra. Act1 Emitir orden de frotis. Act2 Llevar las muestras.

	Act3 Analizar muestras. Act4 Entregar resultado. Act4 Archivar resultados.
--	--

**Tabla 2.4.5.1 Descripción del proceso realizar endoscopia.**

## 2.5 Propuesta del Sistema

Teniendo en cuenta el problema planteado se propuso como solución el desarrollo del módulo Medios de Diagnóstico en el que están presentes todos los procesos de las áreas que determinan o posibilitan el diagnóstico de una enfermedad, estas son: Optometría, Endoscopia, Drenaje Biliar, Electrocardiograma, Laboratorio de Alergia, Laboratorio Clínico y Parasitología.

### 2.5.1 Especificación de requerimientos de software

La especificación de los requerimientos de software es una descripción completa del comportamiento del sistema que se va a desarrollar. Es decir, especifica qué debe hacer el sistema en sus requerimientos. Un requerimiento puede definirse como un elemento necesario dentro de un sistema, que puede representar una condición o capacidad, para satisfacer un contrato, estándar, especificación u otro documento formal. Esta actividad debe hacerse con gran precisión, por el papel primordial que juega en la producción de software, enfocada en la definición de lo que se desea producir, mediante una descripción más clara del comportamiento del sistema.

Los requerimientos se dividen en:

Funcionales: son capacidades o condiciones que el sistema debe cumplir.

No funcionales: son propiedades o cualidades que el producto debe tener.

### 2.5.2 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Durante el levantamiento de procesos se identificaron algunos procesos que son el punto de partida para identificar qué debe hacer el sistema. Los requerimientos funcionales no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

Teniendo en cuenta los procesos del negocio y las actividades candidatas a informatizar es posible definir los siguientes requisitos funcionales.

<b>Requisitos funcionales</b>	
<b>RF1-</b> Buscar paciente	<b>RF8.3-</b> Modificar unidad de medida
<b>RF2-</b> Inscribir paciente	<b>RF8.4-</b> Eliminar unidad de medida
<b>RF3-</b> Buscar paciente inscrito	<b>RF9 Gestionar intervalo de referencia</b>
<b>RF4-</b> Buscar exámenes asociados	<b>RF9.1-</b> Agregar intervalo de referencia
<b>RF6- Registrar resultados</b>	<b>RF9.2-</b> Buscar intervalo de referencia
<b>RF6.1-</b> Registrar resultados de Creatinina	<b>RF9.3-</b> Modificar intervalo de referencia
<b>RF6.2-</b> Registrar resultados de Proteínas Totales	<b>RF9.4-</b> Eliminar intervalo de referencia
<b>RF6.3-</b> Registrar resultados de Hemoglobina	<b>RF10 Gestionar grupo</b>
<b>RF6.4-</b> Registrar resultados de Albúmina	<b>RF10.1-</b> Agregar grupo
<b>RF6.5-</b> Registrar resultados de Glicemia	<b>RF10.2-</b> Buscar grupo
<b>RF6.6-</b> Registrar resultados de Bilirrubina	<b>RF10.3-</b> Modificar grupo
<b>RF6.7-</b> Registrar resultados de Triglicéridos	<b>RF10.4-</b> Eliminar grupo
<b>RF6.8-</b> Registrar resultados de Fósforo	<b>RF11 Gestionar sugerencia</b>
<b>RF6.9-</b> Registrar resultados de Urea	<b>RF11.1-</b> Agregar sugerencia
<b>RF6.10-</b> Registrar resultados de Amilasa	<b>RF11.2-</b> Buscar sugerencia
<b>RF6.11-</b> Registrar resultados de Endoscopia	<b>RF11.3-</b> Modificar sugerencia
<b>RF6.12-</b> Registrar resultados de Drenaje Biliar	<b>RF11.4-</b> Eliminar sugerencia
<b>RF6.13-</b> Registrar resultados de la prueba de Puntura	<b>RF12 Gestionar examen</b>
<b>RF6.14-</b> Registrar resultados de la prueba de Escarificación	<b>RF12.1-</b> Agregar examen
<b>RF6.15-</b> Registrar resultados de la prueba de Parche	<b>RF12.2-</b> Buscar examen
<b>RF6.16-</b> Registrar resultados de la prueba reacción Intradérmica	<b>RF12.3-</b> Modificar examen
<b>RF6.17-</b> Registrar resultados de electrocardiograma (ECG)	<b>RF12.4-</b> Eliminar examen
	<b>RF13 Gestionar tipo examen</b>
	<b>RF13.1-</b> Agregar tipo examen
	<b>RF13.2-</b> Buscar tipo examen

<p><b>RF6.18-</b> Registrar resultados de prueba de refracción</p> <p><b>RF7 Gestionar tipo de muestra</b></p> <p><b>RF7.1-</b> Agregar tipo de muestra</p> <p><b>RF7.2-</b> Buscar tipo de muestra</p> <p><b>RF7.3-</b> Modificar tipo de muestra</p> <p><b>RF7.4-</b> Eliminar tipo de muestra</p> <p><b>RF8 Gestionar unidad de medida</b></p> <p><b>RF8.1-</b> Agregar unidad de medida</p> <p><b>RF8.2-</b> Buscar unidad de medida</p>	<p><b>RF13.3-</b> Modificar tipo examen</p> <p><b>RF13.4-</b> Eliminar tipo examen</p> <p><b>RF14 Gestionar configuración de examen</b></p> <p><b>RF14.1-</b> Buscar configuración de examen</p> <p><b>RF14.2 -</b> Modificar configuración de examen</p> <p><b>RF14.3 –</b> Eliminar configuración de examen</p>
--	---

### 2.5.3 Requerimientos no funcionales

Los requerimientos no funcionales tienen que ver con las propiedades o cualidades que de una forma u otra limitan al sistema y que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto.

### Requerimientos de seguridad de acceso y administración de usuarios

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

Se mantendrá un segundo nivel de seguridad entre estaciones de trabajo, garantizando únicamente la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión. Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

### Requerimientos de restricciones de diseño

La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio. La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar

siendo ejecutados por cada usuario. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

## Requerimientos de Interfaz

- **Interfaces de usuario**

Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de función y menús desplegados que faciliten y aceleren su utilización. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español.

- **Interfaces software**

Se interactuará con el sistema alas HIS para realizar solicitudes y obtener resultados relacionados con la hospitalización del Paciente.

## Requerimientos de hardware

- **Estaciones de trabajo**

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

- **Servidores**

La solución estará conformada fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

## Requerimientos de software

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

## 2.6 Descripción de prototipos no funcionales

Una vez identificados los requerimientos del sistema a construir se diseñaron los prototipos de interfaz de usuario no funcionales, especificándose en cada uno de ellos cómo se implementarían en el futuro sistema. A continuación se explican algunos de estos prototipos:

El registrar resultado cuantitativo estará compuesto por los 10 exámenes del laboratorio clínico, entre ellos se encuentra la glicemia (Ver figura 2.6.1):



El prototipo de interfaz, titulado "Resultado del examen", contiene los siguientes elementos:

- Campo de texto "Examen:"
- Campo de texto "Inscripción:"
- Campo de texto "Sexo:"
- Campo de texto "Intervalo:"
- Campo de texto "Unidad de medida:" con un menú desplegable que muestra "Seleccione".
- Campo de texto "Muestra:" con un menú desplegable que muestra "Seleccione".
- Campo de texto "Resultado:"
- Botón "Aceptar"
- Botón "Cancelar"

**Fig 2.6.1 Prototipo de interfaz registrar resultado de examen cuantitativo**

El sistema mostrará una interfaz con los siguientes datos: examen es el nombre del análisis al cual se le registrará el resultado, el atributo inscripción es el número generado por el sistema que identifica al paciente, el sexo del paciente, se visualizará el intervalo de referencia y la unidad de medida tradicional en la que está dado ese examen y si se selecciona otra unidad de medida el intervalo es convertido de la unidad de medida seleccionada a la tradicional, la muestra del examen será seleccionada por el usuario, y en el campo resultado el usuario escribirá el resultado del examen. Si el usuario da clic en Aceptar queda registrado el resultado del análisis para el paciente y si da clic en Cancelar cancela la operación.

El registrar resultado cualitativo estará compuesto por 8 exámenes correspondientes a las demás áreas identificadas en los medios de diagnóstico entre ellos se encuentra el drenaje biliar (Ver figura 2.6.2):

El prototipo de interfaz, titulado "Registrar Resultados de Drenaje Biliar", presenta un formulario con los siguientes elementos:

- Una tabla con tres columnas: "Inscripción", "Sexo" y "Diagnóstico".
- Un campo de texto etiquetado "Descripción:" con botones de flecha arriba y abajo.
- Un campo de texto etiquetado "Estado:" con botones de flecha arriba y abajo.
- Botones "Aceptar" y "Cancelar" en la parte inferior.

**Fig 2.6.2 Prototipo de interfaz registrar resultado de examen cualitativo**

La interfaz muestra los siguientes datos: inscripción que es el número generado por el sistema que identifica al usuario, sexo del paciente y diagnóstico que no es más que una explicación de porqué el paciente acude a realizarse un drenaje biliar. El usuario deberá llenar los campos de descripción y estado, en el primero debe detallar cómo fue el proceso de drenar la bilis del paciente y si este tuvo algún problema, así como cuál fue el resultado final y en el segundo debe dar una explicación del estado en el cual se encuentra el paciente y confirmar si el diagnóstico previo del médico coincide con el resultado final. Si el usuario da clic en Aceptar queda registrado el resultado del análisis para el paciente y si da clic en Cancelar cancela la operación.

El desarrollo de este capítulo permitió comprender el funcionamiento del negocio, identificándose como procesos a informatizar la recepción de los pacientes y el registro de resultados. Se identificaron los requerimientos funcionales entre los que se pueden mencionar Buscar paciente, Inscribir paciente, Buscar pacientes inscritos, entre otros y los no funcionales comprenden requerimientos de interfaz, de hardware, de software, entre otros, lo que permitió un mejor entendimiento del sistema a construir. El resultado de

estos dos flujos de trabajo y en específico el de requerimientos permiten comenzar con las actividades del flujo de trabajo de Diseño.

## Capítulo 3. Diseño del Sistema

El diseño de un sistema de software da una mayor comprensión de los aspectos relacionados con los requerimientos no funcionales y contribuye a la definición de una arquitectura estable y sólida, creando básicamente una especificación que describe cómo implementar el sistema. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Desempeña un papel muy importante en el desarrollo de un software porque describe todos los aspectos del sistema a construir. Dentro del diseño es donde se fomenta la calidad del proyecto y es la única manera de materializar con precisión los requerimientos del cliente.

El objetivo de este capítulo está dirigido a comprender de una mejor manera los elementos relacionados con el diseño del sistema. Se presentan los diagramas de clases del diseño, especificando la estructura y definición de los elementos que este posee, además se justifica el uso de alguno de los patrones de diseño utilizados.

### 3.1 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, sirve de abstracción de la implementación del sistema y de ese modo es utilizado como una entrada fundamental de las actividades asociadas al flujo de trabajo de implementación.

Este modelo es representado por un sistema de diseño que denota el subsistema de nivel más alto del modelo. La utilización de otro subsistema es una forma de organización de este artefacto, en porciones más manejables. Los subsistemas y clases del diseño representan abstracciones del subsistema y componentes de la implementación del sistema. Estas abstracciones son directas y representan una sencilla correspondencia entre el diseño y la implementación. [17]

#### 3.1.1 Definición de elementos de diseño

En el Modelo de diseño se establece la realización de las funcionalidades en clases incluyendo una orientación hacia el entorno de implementación. Está constituido por los diagramas de clases. Se realizará

un diagrama de clases por cada funcionalidad simple y uno por cada CRUD (gestionar) en el caso de los codificadores. La estructura del diagrama de clases responderá a la arquitectura definida.

Para la realización de los diagramas de clases del diseño se utiliza la extensión de UML para la utilización de estereotipos web. Esta extensión presenta como elementos más significativos tres clases: “Server Page”, “Client Page” y “Form” empleadas para el código servidor, código cliente y formularios respectivamente.

El código servidor se encarga de construir o generar el resultado XHTML que conforma el código cliente, los formularios envían sus datos al código servidor para ser procesados los pedidos, además forman parte del código cliente o resultado XHTML. Es por esto que la relación entre la clase empleada para el código cliente y la clase empleada para el formulario es de composición. Es importante destacar que una página cliente es construida por una sola página servidora; esta a su vez, puede completar su funcionamiento incluyendo algunas de las librerías que componen el framework JSF.

Para la nomenclatura de estas clases en los diagramas de clases se siguió la siguiente estructura: los diagramas de clases del diseño se nombrarán: DCD\_<Nombre de la funcionalidad>, <NombreClaseCliente>.siaps para las páginas clientes, frmsiaps es el nombre que tienen todos los formularios, y las clases servidoras están definidas de acuerdo a una funcionalidad, para el listar se define CL\_<NombreOpción>.java, para el agregar es CC\_<NombreOpción>.java, para el modificar se utiliza CM\_\_<NombreOpción>.java, para el eliminar se usa CE\_\_<NombreOpción>.java y para el exportar CR\_\_<NombreOpción>.java. En los diagramas también aparecen las clases entidades y sus relaciones con las clases controladoras. La clase entityManager representa la clase con la que se relacionan las controladoras para llevar a cabo las operaciones de persistencia de datos.

En la estructura de los diagramas de clases del diseño se manifiesta la aplicación de los patrones de diseño, así como las restricciones que establecen sobre la arquitectura definida. Estos diagramas están estructurados siguiendo el patrón 3 capas, el cual permite la separación de los objetos que componen el diseño del sistema en una capa de presentación, de negocio y una de persistencia o de acceso a datos. En la capa de presentación van a estar las clases y objetos relacionados con la presentación de la información a los usuarios, es decir de la interacción de los usuarios con la aplicación, en la capa de

negocio estarán las clases que corresponden con la lógica de negocio y en la de persistencia las relacionadas con la persistencia de los datos.

### 3.1.2 Patrones de Diseño

Los desarrolladores con experiencia que usan la programación orientada a objetos acumulan un repertorio tanto de principios generales como de soluciones con el objetivo de aplicar ciertos estilos que les guían en la creación de software. Si se codifican estos principios y estilos con un formato estructurado que describa el problema y la solución, y se les da un nombre, podrían llamarse “Patrones”. De manera más simple un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlos en nuevas situaciones. [18]

Los patrones de diseño que se utilizan son los patrones GRASP (Patrones de Software para la asignación General de Responsabilidad), estos describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para entender el diseño. Se pueden mencionar 5 patrones principales que son: creador, experto, alta cohesión, bajo acoplamiento y controlador.

**Experto:** es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo.

**Creador:** el patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que: tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase.

**Alta cohesión:** Expresa que la información que almacena una clase debe de ser coherente y estar en la mayor medida de lo posible relacionada con la clase.

**Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

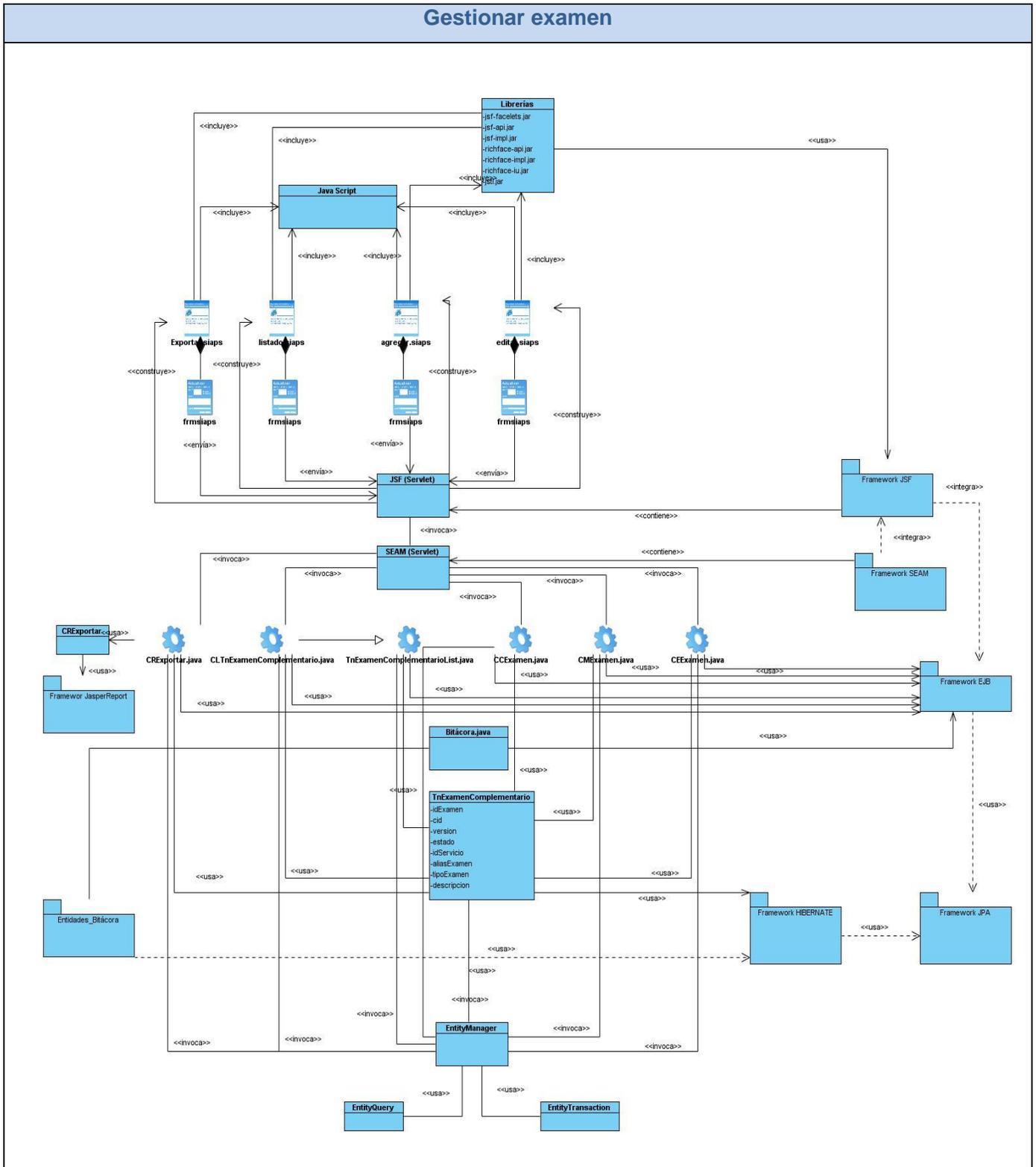
**Patrón Controlador:** Asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las

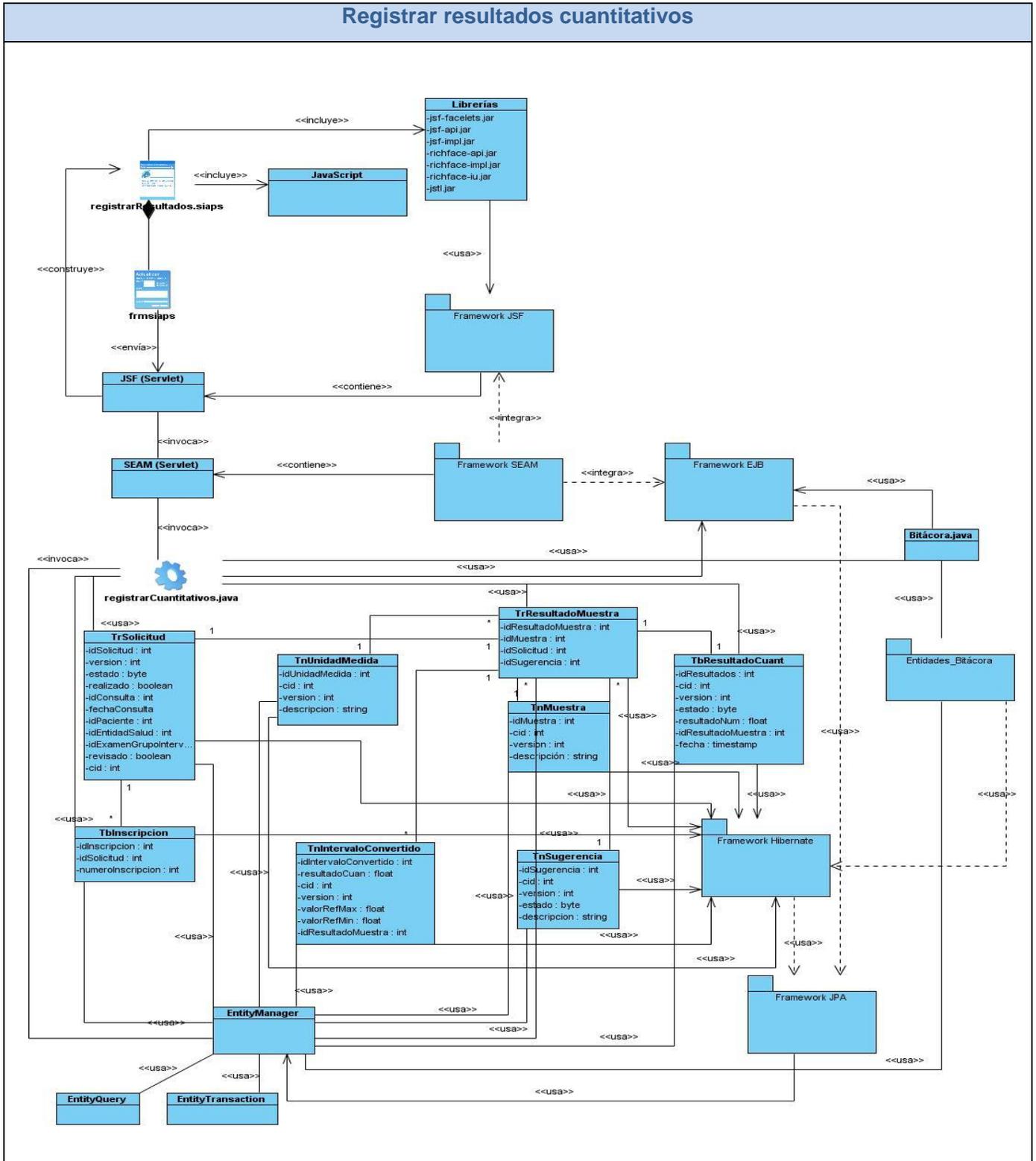
delega en otras clases con las que mantiene un modelo de alta cohesión, es el que recibe los datos del usuario y el que los envía a las distintas clases según el método llamado.

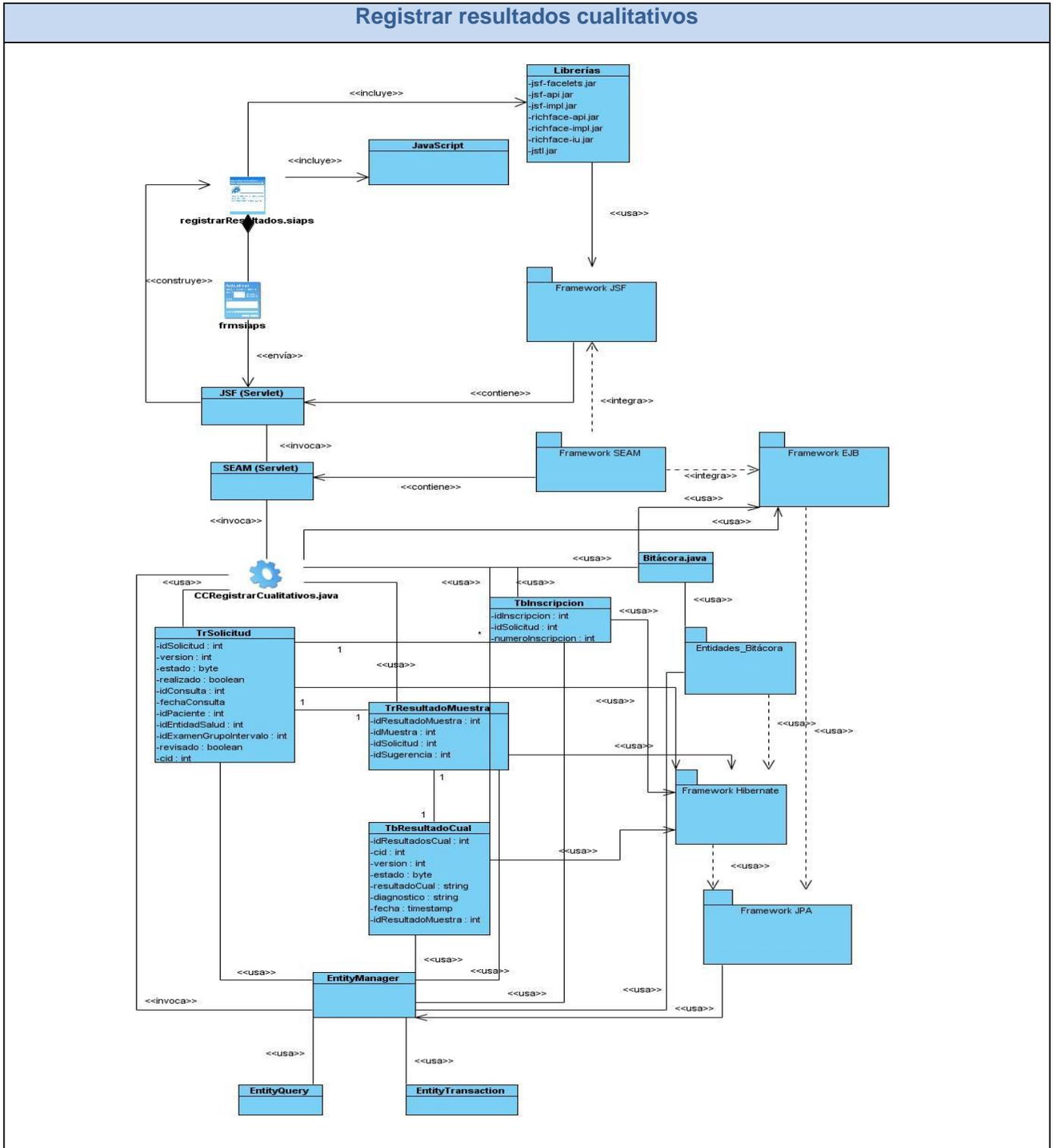
### 3.1.3 Diagramas de clases del diseño

Este diagrama es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente contienen la información de: las clases, asociaciones y atributos, y proporcionan información sobre los tipos de los atributos, navegabilidad y dependencia.

Durante el flujo de trabajo de análisis y diseño se hace uso de los diagramas de clases, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. El UML no define concretamente un elemento denominado "diagrama clases del diseño", sino que se sirve de un término más genérico: que son los "diagrama de clases". A continuación se muestran algunas representaciones.







## 3.1.4 Descripción de las clases del registrar resultados cuantitativos

### Capa de Presentación

#### Clase resultadosCuantitativos.siaps

##### Propósito

Proveer la interacción con el usuario.

##### Descripción

La clase resultadosCuantitativos.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite insertar todos los datos necesarios referentes a registrar los resultados del análisis de creatinina. Posee un conjunto de validaciones en Java Script que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.



resultadosCuantitativos.siaps

### Capa de Negocio

#### Clase registrarCuantitativo.java

##### Propósito

Proveer una respuesta a las peticiones realizadas en la vista.

##### Descripción

La clase registrarCuantitativo.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de insertar los datos necesarios para registrar un análisis de creatinina. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.



registrarCuantitativos.java

## Capa de Datos

### Clase TbResultadoCuantitativo

**Propósito:** Proveer el mapeo con la base de datos.

### Descripción

La clase TbResultadoCuantitativo.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de los resultados de los exámenes cuantitativos. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA

TbResultadoCuant
-idResultados : int
-cid : int
-version : int
-estado : byte
-resultadoNum : float
-idResultadoMuestra : int
-fecha : timestamp

## Descripción de clases comunes

### Capa de Presentación

#### Clase frmsiaps

#### Propósito

Enviar los datos a las páginas servidoras.

#### Descripción

La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario. No tienen operaciones, el método para el paso de los parámetros es \$\_POST.



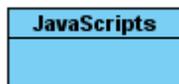
## Clase Java Script

### Propósito

Proveer el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

### Descripción

Permite realizar las validaciones del lado del cliente y construir páginas más dinámicas integradas a un navegador web.



## Clase Librerías

### Propósito

Permite la creación de interfaces.

### Descripción

Contiene un conjunto de librerías que permiten construir una interfaz de usuario con componentes reutilizables y extensibles. Reducen significativamente la carga de construir y mantener aplicaciones web con componentes de interfaz del lado del servidor.



## Clase JSF Servlet

### Propósito

Interceptar las peticiones a las páginas JSF.

### Descripción

Es el controlador de JSF que intercepta las peticiones de las páginas clientes, asociándoles a estas

páginas clases java que recogen la información introducida y que disponen de métodos que responden a las acciones del usuario. Además prepara el contexto JSF antes de enrutar a las páginas correspondientes e interviene en la construcción de la respuesta para generar la vista, luego de ser invocada una petición.



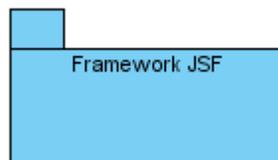
## Paquete Framework JSF

### Propósito

Simplificar el desarrollo de las interfaces de usuario en aplicaciones java basadas en el patrón Modelo-Vista-Controlador.

### Descripción

La tecnología Java Server Faces es un framework de los componentes de la interfaz de usuario y es válido para todas aquellas aplicaciones web basadas en la tecnología Java, está basado en el patrón MVC. Forma parte de la especificación JEE 5.



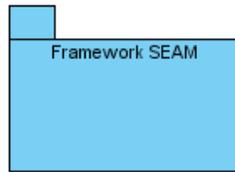
## Paquete Framework SEAM

### Propósito

Integrar la capa de presentación, con la capa de negocio y persistencia.

### Descripción

Conjunto de clases agrupadas en componentes para construir aplicaciones en java. Integra tecnologías como Java Script Asíncrono y XML (AJAX), Java Server Faces (JSF), Java Persistence API (JPA), Enterprise Java Beans (EJB 3.0) y Business Process Management (BPM) en un sistema unificado de completa solución centralizada, con sofisticadas herramientas. Forma parte de la especificación JEE



## Capa de Negocio

### Clase Bitácora.java

#### Propósito

Proveer las funcionalidades para realizar las auditorías del sistema.

#### Descripción

Permite realizar las funciones para almacenar datos como la fecha, hora, usuario, contraseña, entre otros, del usuario que inicia y finaliza la sesión. Además datos como el módulo y funcionalidad accedidos así como las acciones ejecutadas.



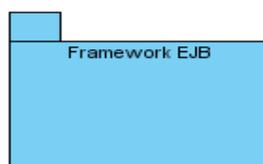
## Paquete Framework EJB

#### Propósito

Encapsular la lógica de negocio que complementa el propósito de la aplicación.

#### Descripción

Es una plataforma para construir aplicaciones de negocio portables, escalables y reutilizables utilizando el lenguaje de programación java. El objetivo de Enterprise JavaBeans (EJB) 3.0 es simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. Forma parte de la especificación JEE 5.



## Capa de Datos

### Clase EntityManager

#### Propósito

Gestionar las entidades proporcionando servicios de persistencia.

#### Descripción

Permite realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) que impliquen entidades.



### Clase EntityQuery

#### Propósito

Agregar consultas que pueden aplicarse a las entidades del modelo.

#### Descripción

Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. Permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición.



### Clase EntityTransaction

#### Propósito

Agrupar las operaciones sobre datos persistentes en una unidad transaccional.

#### Descripción

Permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.



EntityTransaction

A diagram showing a rectangular box with a light blue background and a black border. The text "EntityTransaction" is centered at the top of the box.

## Paquete Entidades Bitácora

### Propósito

Agrupar las entidades que contienen la información de las auditorías del sistema.

### Descripción

Contiene el conjunto de entidades que poseen la información de los usuarios y sus trazas en cuanto a sesión utilizada, módulos accedidos, funcionalidades permitidas, acciones realizadas y atributos modificados. Son utilizadas por la clase Bitácora.java para realizar las auditorías del sistema.



Entidades\_Bitácora

A diagram showing a rectangular box with a light blue background and a black border. The text "Entidades\_Bitácora" is centered at the top of the box.

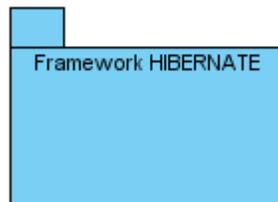
## Paquete Framework HIBERNATE

### Propósito

Proveer el mapeo objeto/relacional con la base de datos.

### Descripción

Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional u Object Relational Mapping (ORM) de código abierto y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada permite generar BD en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2, MySql, entre otras.



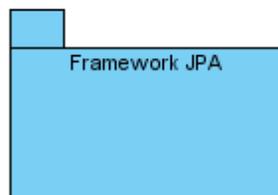
## Paquete Framework JPA

### Propósito

Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.

### Descripción

Conjunto de clase agrupadas en componentes que constituyen la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.



Al concluir el desarrollo de este capítulo se han obtenido algunos artefactos de este flujo de trabajo necesarios para el desarrollo del sistema y que constituyen entradas imprescindibles en el flujo de implementación, como es el caso del modelo de diseño, con los diagramas de clases del diseño. Para un mejor entendimiento del mismo se describen las principales clases que lo componen entre ellas el framework JPA, SEAM, Hibernate, EJB, entre otros, así como las específicas de cada funcionalidad como la clase controladora registrarCuantitativo.java y la página cliente resultadosCuantitativos.siaps.

Se especifican otros aspectos relacionados con el diseño como son los patrones GRASP utilizados como el experto, creador, controlador, alta cohesión y bajo acoplamiento, así como las pautas de diseño utilizadas para la representación de los diagramas de clases.

## **Capítulo 4. Implementación**

La implementación es el centro en la fase de construcción, aunque se lleva a cabo en iteraciones de otras fases, durante la de elaboración crea la línea base de la arquitectura y durante la de transición trata defectos tardíos encontrados en algunas distribuciones del sistema. En este capítulo se empieza con el resultado del diseño posibilitando la implementación del sistema en términos de componentes, se describe cómo los componentes se organizan físicamente de acuerdo a los nodos específicos en el diagrama de despliegue. Además, se aborda sobre el tratamiento de excepciones, así como las estrategias de codificación, estándares y estilos que se utilizan.

### **4.1 Justificación de interacción con otros sistemas**

El módulo Medios de Diagnóstico interactúa con otros módulos que conforman el alasSIAPS, logrando una comunicación a nivel de Bases de Datos. El módulo de Configuración proporciona la seguridad de la aplicación, brinda la información asociada a los departamentos, los servicios médicos y a las entidades de salud.

El módulo Clínico Quirúrgico es el que genera las solicitudes de exámenes a partir de la consulta realizada al paciente por el médico, usa el listado de los exámenes complementarios que se realizan en una entidad de salud para realizar las solicitudes y consultan los resultados de los exámenes orientados y el Visor de Historia Clínica Individual usa los datos de los resultados de los exámenes complementarios que fueron realizados a un paciente.

### **4.2 Modelo de implementación**

Para comenzar a desarrollar el flujo de trabajo de Implementación se comienza con el resultado del diseño, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación. Además de implementar las clases y subsistemas encontrados durante el diseño. Las clases se implementan como componentes de ficheros que contienen código fuente. Uno de los principales propósitos de este flujo de trabajo consiste en desarrollar la arquitectura y el sistema en su conjunto.

### 4.2.1 Diagrama de despliegue

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Este diagrama está compuesto por protocolos, dispositivos y procesadores, los nodos representan a estos dos últimos, además poseen relaciones que representan medios de comunicación entre ellos. Permite comprender la correspondencia entre la arquitectura de software y la arquitectura de hardware.

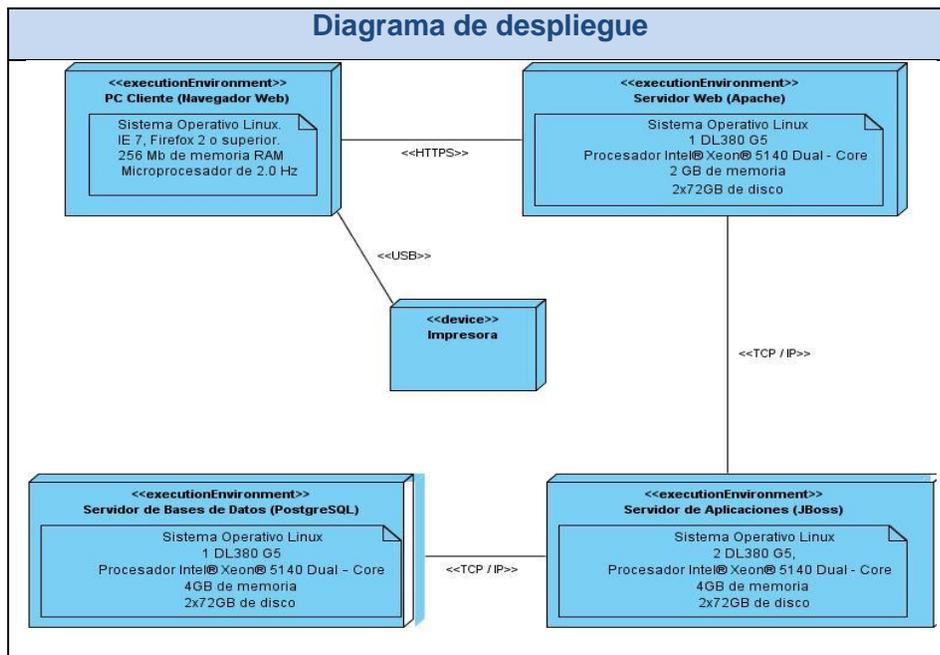


Figura 4.2.1 Diagrama de despliegue

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

El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación, estos son reglas específicas de cada lenguaje de programación cuyo cumplimiento reduce de forma significativa el riesgo de que los desarrolladores introduzcan errores. La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que la aplicación pueda modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Para la implementación del sistema propuesto se utilizaron varios estándares de codificación como son:

Idioma: Se debe utilizar como idioma el español, las palabras no se acentuarán.

## Indentación

Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.

## Inicio y fin de bloque

Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque `{}`. Lo mismo sucede para el caso de las instrucciones `if`, `else`, `for`, `while`, `do while`, `switch`, `foreach`.

## Aspectos generales

El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla. Los inicios (`{`) y cierre (`}`) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar (`{`) en la línea de un código cualquiera, esto requiere una línea propia.

## **Comentarios, separadores, líneas, espacios en blanco y márgenes**

Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con sólo leerlo una vez.

Ubicación de comentarios 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.

## Líneas en blanco

Se emplean antes y después de métodos, clases y estructuras. 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.

## Espacios en blanco

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

Ejemplo: `producto = nomproducto`.

## Aspectos generales

Sobre el comentario:

- Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.

Sobre los espacios en blanco:

No se debe usar espacio en blanco:

- Después del corchete abierto y antes del cerrado de un arreglo.
- Después del paréntesis abierto y antes del cerrado.
- Antes de un punto y coma.

## Variables y constantes

Apariencia de variables:

Las variables tendrán un prefijo para el tipo de datos en minúscula. El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, la cual identificará el tipo de datos al que se refiere, en caso de que sea un nombre compuesto se empleará notación CamellCasing\*\*.

Ejemplo: sNombrePaciente.

Apariencia de constantes:

Todas sus letras en mayúscula. Se deben declarar las constantes con todas sus letras en mayúscula.

## Aspectos generales

Nombres de las variables y constantes. El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.

## Clases y objetos

Objetivo: Nombrar las clases e instancias de forma estándar para todas las aplicaciones.

## Apariencia de clases y objetos

Primera letra en mayúscula: Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación

PascalCasing\*. Ejemplo: MiClase (). Para el caso de las instancias se comenzará con un prefijo que identificará el tipo de dato, este se escribirá en minúscula.

## Apariencia de atributos

Primera letra en minúscula: El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación CamellCasing\*\*.

## Aspectos generales

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

## **Bases de Datos, Tablas, esquemas y Campos**

### Apariencia de la Base de Datos

Las 3 primeras letras corresponden al nombre del departamento, seguido de underscore y a continuación el prefijo bd, luego el nombre completamente en minúscula precedido por underscore, en caso de que sea un nombre compuesto se empleará la misma notación. Los nombres serán cortos y descriptivos.

Ejemplo: APS\_bd\_medio\_diag.

### Apariencia de las tablas

Las 2 primeras letras representan el tipo. Todas las letras en minúscula. El nombre a emplear para las tablas debe comenzar con el prefijo tb seguido de underscore y luego debe escribirse todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizará underscore para separarlo.

Ejemplo: "tb\_inscripcion".

### Tablas que representen relaciones

Las 2 primeras letras representan el tipo, todas las letras deben estar en minúscula. El nombre a emplear para estas tablas de relación debe comenzar con el prefijo tr seguido de underscore y el nombre será la concatenación del nombre de las dos tablas que la generaron separados por underscore todo en minúscula. Ejemplo: "tr\_examen\_grupo\_intervalo".

## Tablas que representen nomencladores

Las 2 primeras letras representan el tipo. Todas las letras deben estar en minúscula. El nombre a emplear para estas tablas de relación debe comenzar con el prefijo tn seguido de underscore. El nombre será corto y descriptivo todo en minúscula. Ejemplo: "tn\_muestra".

Apariencia de los campos: Todas las letras deben estar en minúscula. El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive (literalmente sensible a las mayúsculas/minúsculas) del gestor. Ejemplo: "estado".

## Nombre de los campos

En caso de identificadores. Todos los campos identificadores van a comenzar con el identificador id seguido de underscore y posteriormente el nombre del campo. Ejemplo: id\_muestra.

## Sentencias SQL

Todas las letras deben estar en mayúscula. Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

## Aspectos generales

El nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con sólo leerlos se conozca el propósito de los mismos.

## **4.4 Tratamiento de Excepciones**

Una excepción es un evento que ocurre durante la ejecución de un programa interrumpiendo el flujo normal de las sentencias. Dicho evento puede ser desde serios problemas de hardware, hasta los simples errores de programación y pueden ser tratados mediante una estructura de control que poseen los lenguajes de programación de alto nivel, diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla. A esta estructura de control se le conoce como tratamiento de excepciones.

Validar los datos es una tarea común que se produce a través de cualquier aplicación, desde la capa de presentación a la capa de persistencia. A menudo la misma lógica de validación es implementada en cada capa, resultando lento y propenso a errores. Para evitar la duplicación de estas validaciones en cada

capa, los desarrolladores a menudo utilizan los paquetes de validaciones lógicas en el modelo de dominio, llenando las clases de dominio con código de validación que es en realidad metadatos sobre la propia clase.

En el sistema propuesto se utilizan todas las facilidades que brinda la plataforma para el tratamiento de excepciones. Para cada fragmento de código donde se espere una situación anómala, se definen las excepciones correspondientes para luego ser tratadas evitando la interrupción del sistema. También se emplean un conjunto de tipos de excepciones predefinidas por los marcos de trabajos que se utilizan en el sistema.

El uso de diferentes tecnologías y la integración que existe entre ellas, permiten capturar y controlar posibles situaciones desde diferentes puntos de la aplicación. En las páginas clientes se cuenta con un conjunto de componentes denominados validadores, que permiten establecer tipos de datos y formatos controlando que el envío de los activos al servidor sean los esperados.

Además, el marco de trabajo Seam brinda un potente conjunto de excepciones predefinidas, que conjuntamente con la clase FacesMessages, permite tratar estas situaciones desde las clases controladoras correspondientes y mostrar mediante la clase antes mencionada, los resultados del tratamiento. Seam permite además mediante el fichero de configuración page.xml, todo un flujo de navegación basado en excepciones.

En el capítulo se ha concluido la fase de implementación. Se desarrolló un sistema con la totalidad de las funcionalidades previstas para el correcto funcionamiento de los procesos de atención al paciente del módulo de Medios de diagnóstico de la Atención Primaria de Salud. Se establecieron las estrategias de codificación que deben seguir los programadores, la nomenclatura que deben tener los nombres de las tablas y los atributos de la base de datos. Se especificaron los nodos y dispositivos necesarios para el despliegue del sistema como fueron las computadoras, el servidor web Apache, servidor de aplicaciones JBoss. Se definió la forma en la que el sistema maneja los errores y se especificaron las restricciones del código, las cuales especifican cómo debe trabajar cada uno de los desarrolladores del sistema.

### Conclusiones

La realización del presente trabajo ha posibilitado cumplir con los objetivos propuestos, por lo que se pueden plantear las siguientes conclusiones:

- ✓ Los procesos identificados en el área de Medios de diagnóstico de las instituciones de Atención Primaria de Salud no están completamente estandarizados, dificultando el trabajo del personal que labora en la misma.
- ✓ Luego de realizar el estudio de las tendencias, antecedentes y sistemas informáticos existentes se evidenció la necesidad de crear el módulo Medios de Diagnóstico dado que los sistemas analizados tanto a nivel nacional como internacional no resuelven el problema planteado.
- ✓ La utilización de la arquitectura y tecnologías definidas por el CESIM garantizan un ambiente de desarrollo que permite la creación de un sistema robusto y flexible.
- ✓ Se transitó por las fases de Requerimiento, Diseño e Implementación definidas en el libro de procesos de Administración de requisitos, lográndose el desarrollo del módulo Medios de diagnóstico del sistema alasSIAPS, facilitando la gestión de información en esta área.

### Recomendaciones

Los autores recomiendan:

- ✓ Incorporar otras funcionalidades relacionadas con la elaboración de reportes y control de los reactivos.
- ✓ Lograr en un futuro la comunicación con equipos autoanalizadores para almacenar los resultados de los exámenes de forma automática.
- ✓ Incluir un sistema de ayuda en línea para los usuarios que interactúan con la aplicación.

### Referencias Bibliográficas

1. Aniversario, 50. Justicia Social. [En línea] [Citado el: 18 de noviembre de 2009.]  
[http://aniversario50.cubava.cu/index.php?option=com\\_content&view=article&id=198%3AAsalud&catid=59%3Ajusticia-social&Itemid=56&lang=es](http://aniversario50.cubava.cu/index.php?option=com_content&view=article&id=198%3AAsalud&catid=59%3Ajusticia-social&Itemid=56&lang=es).
2. Biblioteca, Virtual en salud. Revista Cubana de Medicina General. [En línea] Irene Pedomo Victoria, 19 de 2 de 2009. [Citado el: 18 de noviembre de 2009.]  
[http://www.bvs.sld.cu/revistas/mqi/vol19\\_2\\_03/mqi18203.htm](http://www.bvs.sld.cu/revistas/mqi/vol19_2_03/mqi18203.htm).
3. SOFTEL. [En línea][Citado el: 12 de febrero del 2010]  
URL: <http://www.softel.cu/productos.htm>.
4. Propuestas para la Informatización de Centros de Atención Médica. [En línea][Citado el: 12 de febrero del 2010]  
URL: <http://www.softel.cu/doc/CentrosAtencionMedica.pdf>
5. Medysistem Software [En línea][Citado el: 10 de febrero del 2010]  
<http://www.medisystem.com.mx/>
6. Virtus Group [En línea][Citado el: 10 de febrero del 2010]  
<http://www.virtus.com.mx/xlab/index.html>
7. VisionDat Software [En línea][Citado el: 10 de febrero del 2010]  
<https://www.visiondatsoftware.com/pages/home.aspx>
8. Alin Software. Programación aplicada a la vida [En línea][Citado el: 10 de febrero del 2010]  
<http://www.alinsoft.com/neoendo.php>
9. Business Process Management (BPM): articulando estrategia, procesos y tecnología. [En línea] Luis Fernando Sánchez Maldonado [Citado el: 11 de febrero del 2010.]  
<http://www.degerencia.com/articulos.php?artid=611>
10. Patrón "Modelo-Vista-Controlador" [En línea] [Citado el: 11 de febrero del 2010.]  
<http://www.proactiva-calidad.com/java/patrones/mvc.html>
11. Arquitectura Basada en Componentes [En línea] [Citado el: 11 de febrero del 2010.]  
<http://www.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>
12. Desarrollo de Software basado en Componentes [En línea][Citado el: 11 de febrero del 2010.]  
<http://www.lambdasi.com.ar/textocomp.asp?id=9>

13. Ajax: Un nuevo acercamiento a las aplicaciones web [En línea] Maestros del Web [Citado el: 11 de febrero del 2010.]

<http://www.maestrosdelweb.com/editorial/ajax/>

14. Introducción a Ajax4jsf [En línea] Juan Alonso Ramos [Citado el: 11 de febrero 2010]

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>

15. Guía Breve de XHTML [En línea] [Citado el: 11 de febrero del 2010.]

<http://www.w3c.es/Divulgacion/guiasbreves/XHTML>

16. Programación Java [En línea][Citado el: 11 de febrero del 2010.]

<http://www.lenguajes-de-programacion.com/programacion-java.shtml>

17. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de Software.* España : s.n., 2000.

18. **Presman Roger S.** Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001.

## Bibliografía

1. Ajax: Un nuevo acercamiento a las aplicaciones web [En línea] Maestros del Web [Citado el: 11 de febrero del 2010.]  
<http://www.maestrosdelweb.com/editorial/ajax/>
2. Alin Software. Programación aplicada a la vida [En línea][Citado el: 10 de febrero del 2010]  
<http://www.alinsoft.com/neoendo.php>
3. Aniversario, 50. Justicia Social. [En línea] [Citado el: 18 de noviembre de 2009.]  
[http://aniversario50.cubava.cu/index.php?option=com\\_content&view=article&id=198%3Aasalud&catid=59%3Ajusticia-social&Itemid=56&lang=es.](http://aniversario50.cubava.cu/index.php?option=com_content&view=article&id=198%3Aasalud&catid=59%3Ajusticia-social&Itemid=56&lang=es.)
4. *APSI\_SW\_DI\_017\_alasSIAP\_Metodología\_Enfocado\_Procesos V1.0*. La Habana : APS, 2009.
5. Arquitectura Basada en Componentes [En línea] [Citado el: 11 de febrero del 2010.]  
<http://www.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>
6. Biblioteca, Virtual en salud. Revista Cubana de Medicina General. [En línea] Irene Pedomo Victoria, 19 de 2 de 2009. [Citado el: 18 de noviembre de 2009.]  
[http://www.bvs.sld.cu/revistas/mgi/vol19\\_2\\_03/mgi18203.htm](http://www.bvs.sld.cu/revistas/mgi/vol19_2_03/mgi18203.htm).
7. Business Process Management (BPM): articulando estrategia, procesos y tecnología.[En línea] Luis Fernando Sánchez Maldonado [Citado el: 11 de febrero del 2010.]  
<http://www.degerencia.com/articulos.php?artid=611>
8. Desarrollo de Software basado en Componentes [En línea] [Citado el: 11 de febrero del 2010.]  
<http://www.lambdasi.com.ar/textocomp.asp?id=9>
9. Fernández, Vanesa Gómez. *0115\_Modelo de Procesos de Negocio con BPM\_Módulo Medios de Diagnóstico*. La Habana : APS, 2010.
10. F7, Grupo de Arquitectua . *Documento de Arquitectura de Software V1.2*. La Habana : s.n., 2008
11. Guía Breve de XHTML [En línea] [Citado el: 11 de febrero del 2010.]  
<http://www.w3c.es/Divulgacion/guiasbreves/XHTML>
12. Introducción a Ajax4jsf [En línea] Juan Alonso Ramos [Citado el: 11 de febrero 2010]  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>
13. Ivar Jacobson, Grady Booch, James Rumbaugh. *El proceso unificado de desarrollo de Software*. España : s.n., 2000

14. Medysistem Software [En línea][Citado el: 10 de febrero del 2010]  
<http://www.medisystem.com.mx/>
15. Patrón "Modelo-Vista-Controlador" [En línea] [Citado el: 11 de febrero del 2010.]  
<http://www.proactiva-calidad.com/java/patrones/mvc.html>
16. Presman Roger S. Ingeniería de Software, un enfoque práctico. McGraw-Hill. 2001
17. Programación Java [En línea][Citado el: 11 de febrero del 2010.]  
<http://www.lenguajes-de-programacion.com/programacion-java.shtml>
18. Propuestas para la Informatización de Centros de Atención Médica. [En línea][Citado el: 12 de febrero del 2010]  
<http://www.softel.cu/doc/CentrosAtencionMedica.pdf>
19. Sánchez, Yosvanys. *Documento de la arquitectura de software*. La Habana : APS, 2009.
20. SOFTEL. [En línea][Citado el: 12 de febrero del 2010]  
<http://www.softel.cu/productos.htm>
21. Virtus Group [En línea][Citado el: 10 de febrero del 2010]  
<http://www.virtus.com.mx/xlab/index.html>
22. VisionDat Software [En línea][Citado el: 10 de febrero del 2010]  
<https://www.visiondatsoftware.com/pages/home.aspx>

## Glosario de Términos

**Albumina:** el análisis de albúmina se realiza para detectar la presencia de una proteína que se encuentra en gran proporción en el plasma sanguíneo, siendo la principal proteína de la sangre y a su vez la más abundante en el ser humano que también es sintetizada por el hígado.

**Bilirrubina:** el examen de bilirrubina total e indirecta se realiza para medir, detectar o monitorear problemas en el hígado o la vesícula biliar.

**Creatinina:** es un compuesto orgánico generado a partir de la degradación de la creatina (que es un nutriente útil para los músculos). Es un producto de desecho del metabolismo normal de los músculos que usualmente es producida por el cuerpo en una tasa muy constante (dependiendo de la masa de los músculos) y normalmente filtrada por los riñones y excretada en la orina. La medición de la Creatinina es la manera más simple de monitorizar la correcta función de los riñones.

**Diagnóstico:** es el procedimiento por el cual se identifica una enfermedad, entidad nosológica, síndrome, o cualquier condición de salud-enfermedad (el "estado de salud" también se diagnostica). Es un juicio clínico sobre el estado psicofísico de una persona; representa una manifestación en respuesta a una demanda para determinar tal estado.

**Drenaje Biliar:** procedimiento que se usa para drenar la bilis y aliviar la presión que causa una obstrucción en los conductos biliares. Tiene como funcionalidad investigar el funcionamiento de la vesícula y buscar parasitismo, bacterias, líquido de la vesícula (bilis), así como extraer la bilis del hígado.

**Electrocardiograma (ECG):** se encarga de medir el ritmo y la regularidad de los latidos del corazón en forma de cinta gráfica continua, así como el tamaño y posición de las aurículas y ventrículos, cualquier daño al corazón y los efectos que sobre él tienen las drogas. En fin el mismo tiene una función relevante en el cribado y diagnóstico de las enfermedades cardiovasculares, alteraciones metabólicas y la predisposición a una muerte súbita cardiaca.

**Endoscopia:** es una técnica diagnóstica y terapéutica que consiste en la introducción de una cámara o lente dentro de un tubo o endoscopio a través de un orificio natural, una incisión quirúrgica, una lesión para la visualización de un órgano hueco o cavidad corporal.

**Glicemia:** el examen de glicemia mide el comportamiento de la glucosa sanguínea basal, así como la concentración de azúcar en la sangre.

**Hemoglobina:** es una proteína en los glóbulos rojos que transporta oxígeno. Un examen sanguíneo puede determinar qué tanta hemoglobina tiene uno en la sangre.

**Laboratorio de Alergia:** se encarga de detectar sustancias responsables de cuadros alérgicos. Las pruebas alérgicas constituyen una prueba diagnóstica fundamental para luchar contra las alergias, así como para la elaboración de las vacunas.

**Laboratorio clínico y parasitología:** apoya a los servicios clínicos en el diagnóstico, tratamiento y seguimiento de pacientes y al control epidemiológico y a la salud pública.

**Optometría:** el área de optometría se encarga de realizarle pruebas a los pacientes con algún padecimiento visual o a aquellos que quieran sacar o renovar la licencia de conducción.

**Policlínico:** unidad de salud donde se brindan servicios médicos a una población geográficamente determinada perteneciente al nivel asistencial de Atención Primaria de Salud.

**Proteínas totales:** es una medición aproximada de todas las proteínas encontradas en la porción líquida de la sangre. Esta prueba examina específicamente la cantidad total de dos clases de proteínas: albúmina y globulina. Las proteínas son partes importantes de todas las células y tejidos. Por ejemplo, la albúmina ayuda a impedir que se escape líquido fuera de los vasos sanguíneos. Las globulinas son una parte importante del sistema inmunitario.

**Prueba cutánea:** es una prueba para diagnosticar enfermedades alérgicas como asma, urticaria, alergia a los alimentos, dermatitis de contacto, entre otros.

**Prueba de escarificación:** prueba que se encarga de hacer incisiones en la piel de poca profundidad con intenciones de insertar alérgenos.

- **Alérgenos:** es una sustancia que puede inducir una reacción de hipersensibilidad (alérgica) en personas susceptibles.
- **Escarificación:** es la acción de producirse escaras en la piel. Las escaras son cicatrices producidas por cortes superficiales o profundos en la dermis. Estas heridas producen una costra que por lo general es de color oscuro, resultante de la muerte de tejido vivo.

**Prueba de parche:** esta prueba reconoce si una sustancia es la causa de una inflamación cutánea del paciente al contactar con la piel. Esta inflamación se llama dermatitis de contacto. Esta prueba es el único medio que tiene el médico para probar que una sustancia es la causa o el agravante de la dermatitis que padece un paciente. Una vez que se identifica el alérgeno, evitarlo debería favorecer la curación de la dermatitis.

**Triglicéridos:** es un examen de laboratorio para medir la cantidad de triglicéridos, un tipo de grasa en la sangre. El cuerpo produce algunos triglicéridos. Los triglicéridos también provienen del alimento que se consume. Cuando el ser humano come, el cuerpo usa las calorías para obtener energía inmediata. Las calorías sobrantes se convierten en triglicéridos. Si se consumen más calorías de las que el cuerpo necesita, el nivel de triglicéridos puede ser alto.

**Urea:** este examen se realiza principalmente para determinar el equilibrio proteínico de una persona y la cantidad de proteína en la dieta necesaria para pacientes gravemente enfermos. También se utiliza para determinar qué tanta proteína consume una persona. La urea es excretada por los riñones, de manera que la excreción de ésta puede ser un reflejo de la función renal.