



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7

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

Módulo Configuración del Subsistema Web para el Sistema
Integral para la Atención Primaria de Salud alas SIAPS

Autores:

Alexis Rodríguez Hernández

Yariel Hernández Espino

Tutores: Ing. Maireny Mendoza Santana

Ing. Yaney Gómez Domínguez

Ciudad de La Habana, Julio de 2010

“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaración de Autoría

Por este medio declaramos que somos los únicos autores de la presente investigación y autorizamos a la Empresa UCI, perteneciente al Ministerio de Informática y las Comunicaciones (MIC) a que hagan el uso que estimen pertinente con el mismo.

Para que así conste, firmamos la presente a los __ días del mes de Julio del ____.

Yariel Hernández Espino

Firma del Autor

Alexis Rodríguez Hernández

Firma del Autor

Ing. Mairenys Mendoza Santana

Firma del Tutor

Ing. Yaney Gómez Domínguez

Firma del Tutor

Datos de Contacto

Ing. Mairenys Mendoza Santana: Graduada de Ingeniería en Ciencias Informáticas en el año 2008, en la Universidad de las Ciencias Informáticas. Presenta la categoría docente de Instructor Recién Graduado, ha impartido la asignatura Segundo Perfil (Software Libre 1er año). Pertenece al Dpto. de Atención Primaria de Salud del Centro Especializado en Soluciones de Informática Médica (CESIM). Se ha desempeñado como responsable de la Práctica Profesional de 2do a 5to año en el Dpto. y como Jefe de Equipo de Desarrollo del Módulo Configuración.

mmendoza@uci.cu

Ing. Yaney Gómez Domínguez: Graduada de Ingeniería en Ciencias Informáticas en el año 2008, en la Universidad de las Ciencias Informáticas. Presenta la categoría docente de Instructor Recién Graduado, ha impartido la asignatura de Ingeniería de Software 3er año y Preparación para Prueba de Nivel de Ingeniería de Software 5to año. Pertenece al Dpto. de Atención Primaria de Salud del Centro Especializado en Soluciones de Informática Médica (CESIM). Se ha desempeñado como Analista Principal del Dpto. y Jefe de Equipo de Procesos

ygomezd@uci.cu

Resumen

A raíz de la decisión de vincular las TIC (Tecnologías de la Informática y las Comunicaciones) a los procesos que se desarrollan dentro del Sistema Nacional de Salud se comienzan a desarrollar herramientas que sirvan de apoyo al personal de salud y a las distintas instituciones médicas, con el objetivo de hacer más eficiente su funcionamiento, posibilitando acceder a la información en tiempo real y disminuir la probabilidad de cometer errores en procesos críticos como el de diagnóstico, aumentando por tanto la calidad del servicio.

El presente informe tiene como objetivo mostrar los pasos a seguir para el desarrollo del Módulo Configuración del Subsistema Web para el Sistema Integral para la Atención Primaria de Salud alas SIAPS.

Se exponen en el documento, las tecnologías y herramientas utilizadas en correspondencia con la arquitectura propuesta por el Centro Especializado en Informática Médica, basadas en estándares abiertos para el desarrollo de la aplicación. Se justifica el uso del proceso de desarrollo seguido para el diseño y construcción del sistema y el lenguaje de modelado gráfico UML (Lenguaje de Modelado Unificado) utilizado en el desarrollo de los flujos de trabajo de Modelado de Negocio, Requerimientos, Diseño e Implementación.

Con el desarrollo de la aplicación se dispondrá del componente Configuración, que incorporan sustanciales beneficios desde el punto de vista administrativo, de gestión y de depuración de procesos.

Palabras Claves: Tecnologías de la Informática y las Comunicaciones, Sistema Nacional de Salud, Módulo Configuración del Sistema Integral para la Atención Primaria de Salud, Lenguaje de Modelado Unificado.

Tabla de Contenidos

Contenido

Introducción	1
CAPÍTULO 1. Fundamentación teórica.....	1
1.1 Sistema Nacional de Salud.....	1
1.2 Informatización del Sistema Nacional de Salud.	2
1.3 Situación problemática y problema a resolver.	3
1.4 Análisis de las soluciones existentes.....	5
1.5 Tendencias y tecnologías.	9
1.6 Arquitectura de Software.....	12
1.7 Servidor de Aplicación.....	17
1.8 Sistema Gestor de Base de Datos.....	18
1.9 Frameworks, librerías y componentes.....	18
1.10 Tecnologías.....	22
1.11 Lenguajes de Programación.....	23
1.12 Metodologías de Desarrollo del Software.	24
1.13 Herramientas a utilizar. [46].....	29
Capítulo 2. Características del Sistema.....	31
2.1 Modelo de Dominio.....	31
2.2 Diagrama del Modelo de Dominio.....	33
2.3 Propuesta de Sistema.....	35
Capítulo 3. Diseño del Sistema.....	41
3.1 Modelo de Diseño.....	41
3.2 Clases del Diseño.....	44
3.3 Diagramas de Clases del Diseño.....	46
Capítulo 4. Implementación.....	52
4.1 Relación con otros sistemas.....	52
4.2 Implementación.....	53
4.3 Estándares de diseño, codificación y tratamiento de excepciones.....	55
Conclusiones	60
Recomendaciones.....	LXI
Referencias Bibliográficas.....	64
Bibliografía	67
Anexos.....	69

Introducción

En pleno siglo XXI y con una invasión de las Tecnologías de la Informática y las Comunicaciones (TIC) en todos los sectores, la demanda de soluciones informáticas se ha hecho cada vez más grande; factibilidad financiera y tecnológica, calidad y poderosas prestaciones figuran entre los deseos más comunes de los clientes, conscientes de que los métodos tradicionales son obsoletos.

En el año 2000 el Ministerio de Salud Pública de Cuba (MINSAP) tomó la decisión de vincular las TIC al proceso de gestión de ese servicio tan importante para el país, con el objetivo de aplicar las novedades que estas proponen y hacer más eficiente su funcionamiento, permitiendo contar con mecanismos novedosos y potentes para la gestión administrativa a todos los niveles.

El Sistema Nacional de Salud (SNS), considera varios niveles de atención médica: primaria, secundaria y terciaria. El primer nivel representa a la Atención Primaria de la Salud (APS), el cual constituye la base del SNS y establece el primer intercambio con los individuos, la familia y la comunidad. Formando parte de la Atención Secundaria se encuentran los servicios hospitalarios, prestados en instituciones en su mayoría provinciales. En cuanto a la Atención Terciaria solo se brinda en algunos centros especializados. **(Ver Anexo 1)**

Dentro del SNS la APS es un punto crítico para el desarrollo de aplicaciones informáticas debido a la diversidad de instituciones que la conforman. El primer nivel de atención de la salud en el marco de la atención médica puede resolver un 80% de los problemas de salud de la población, abordándolos en forma multidisciplinaria dentro de la perspectiva familiar y social. En su núcleo se encuentran grupos de procesos imprescindibles como:

- Historia de Salud Individual (HSI).
- Historia de Salud Familiar (HSF).
- Clínico-Quirúrgico.

- Enfermería.
- Salud Materno Infantil.
- Medios de Diagnóstico.

La introducción de las TIC a este proceso incorpora sustanciales beneficios desde el punto de vista administrativo, de gestión y de depuración de procesos, además del invaluable apoyo y humanización que representa su uso en labores cotidianas, pero a su vez realza aspectos fundamentales dentro del funcionamiento de los procesos identificados en la APS como la adopción de un idioma común que logre la comunicación entre ellos y la correcta manipulación de la información, entre otros.

En la actualidad estas constituyen las principales barreras a la hora de desarrollar sistemas informatizados vinculados a la atención médica. Cuando el sistema de información es monolítico, es decir, consta de un único tipo de desarrollo y no se requiere la integración de otros sistemas, la interoperabilidad puede no ser necesaria, pero la realidad de la mayoría de las organizaciones de salud es la convivencia con múltiples sistemas de información, por lo tanto, la necesidad de integrarlos convierte a la interoperabilidad en un componente clave.

Sin interoperabilidad no existe un sistema nacional de salud integrado, tampoco acceso en tiempo real a la información crítica del paciente en casos de accidentes. Para pensar en un sistema de salud diferente, la interoperabilidad es el componente fundamental que permite juntar todos los datos de los pacientes, sin importar el lugar donde recibió algún tipo de cuidado (hospitales, consultorios médicos, consultorios odontológicos, laboratorios, departamentos de imágenes, etc.)

Lograr que los sistemas informáticos sean capaces de comunicarse entre ellos y exista una sincronía total, no puede ser posible sin un idioma común. Es imposible hacer que todas las entidades adopten el mismo lenguaje, así que se debe crear un nuevo idioma para las aplicaciones informáticas. La utilización de estándares internacionalmente establecidos y la creación de otros puede ser la solución definitiva, además al tratarse de sistemas de información de un volumen significativo de datos como los que se manejan en

la APS también deben tenerse en cuenta los elementos generales utilizados por cada uno de los flujos, para centralizarlos y evitar así la múltiple existencia de funciones con un mismo objetivo, acelerando el proceso de desarrollo y evitando la introducción de incoherencias.

El correcto manejo de la información médica asociada a cada individuo y su protección se hacen necesarios para desarrollar herramientas informáticas para la salud, al ser dicha información valiosa, confidencial y de única incumbencia al personal de salud pertinente y a los pacientes, así mismo la información de las instituciones desde el punto de vista administrativo y de otras categorías.

El proyecto alas SIAPS, encargado de la informatización de la APS, definió una primera etapa donde se desarrollaron cuatro módulos:

- ✓ **Módulo de Medicina Familiar del SIAPS**
- ✓ **Módulo de Medios Diagnósticos del SIAPS**
- ✓ **Módulo de Clínico Quirúrgico del SIAPS.**
- ✓ **Módulo de Enfermería del SIAPS**

Como sistema integrado el alas SIAPS es incapaz de:

- Controlar el crecimiento exponencial de las interfaces de comunicación debido a la incorporación de nuevos módulos a la aplicación.
- Permitir la comunicación a nivel de codificadores y estándares liberándose de las barreras del lenguaje natural.
- Proveer y centralizar la seguridad del sistema.
- Controlar las incoherencias y retrasos generados por la duplicidad de funcionalidades correspondientes a procesos comunes.

Dada la situación anterior el **problema** radica en ¿cómo obtener la configuración base necesaria para el desempeño de los diferentes procesos gestionados por el SIAPS en las unidades de salud correspondientes a la APS?

El **objeto de estudio** se centra en el proceso de gestión de la información de la APS en el SNS. Y el **campo de acción** apunta al proceso de configuración orientado a la seguridad, la interoperabilidad y la estandarización.

Par dar solución al problema antes mencionado se propone como **objetivo general**: desarrollar un módulo de configuración dentro del SIAPS que implemente las funciones necesarias para garantizar la seguridad, gestión de nomencladores y otras características generales que complementan la integridad funcional del sistema.

Entre los **objetivos específicos** se plantean:

- Analizar los aspectos teóricos conceptuales en el Sistema de Salud Nacional cubano y a nivel internacional, específicamente referentes a la configuración.
- Realizar un estudio de las principales tendencias y tecnologías informáticas actuales y una selección de alguna de ellas para el posterior desarrollo de la aplicación.
- Realizar el diseño de un sistema que controle el proceso de la gestión de la configuración y que permita el correcto funcionamiento e integración de los demás módulos del Sistema Integral para la Atención Primaria de Salud.

Las **tareas** propuestas se definen como:

1. Realizar análisis valorativo de las soluciones informáticas existentes en el Sistema de Información para la Salud (SISALUD) y otras a nivel nacional e internacional relacionadas con el tema.
2. Caracterizar la Arquitectura Java EE, aprobada por el CESIM para el desarrollo de las aplicaciones en los diferentes departamentos.
3. Realizar teniendo en cuenta las tecnologías y herramientas propuestas, los flujos de trabajo Modelado de Negocio”, “Gestión de Requerimientos”, “Diseño” e “Implementación”.

4. Implementar las funcionalidades del módulo de Configuración para el SIAPS:

4.1 Ubicación Geográfica

4.2 Áreas de Salud

4.3 Nomencladores Asistenciales

Una vez culminadas las tareas para la liberación del Módulo Configuración del Sistema Integral para la Atención Primaria de Salud se espera obtener varios **beneficios** que potenciarán el funcionamiento y las prestaciones de los restantes módulos en desarrollo, entre las cuales se pueden enunciar:

- Se realizará el diseño e implementación de los nomencladores :
 - Nomencladores Asistenciales
 - Ubicación Geográfica
 - Áreas de Salud
- Se utilizarán estándares internacionales para el diseño de las Bases de Datos que permitan cumplir con el objetivo de interoperabilidad, aumentando la profesionalidad y la calidad del producto.
- Se facilitará el funcionamiento de los restantes módulos del SIAPS.
- Se aplicarán correctas políticas de seguridad que cumplan con los estándares internacionales y con los de defensa y Seguridad Nacional.
- Se garantizará la imprescindible comunicación de los sistemas con cualquier herramienta similar.

La estructura del presente documento se compone de cuatro capítulos en los que se recoge la información obtenida desde el punto de vista investigativo y lo referente al diseño e implementación del Módulo Configuración del Subsistema Web del Sistema Integral para la Atención Primaria de Salud.

Capítulo 1. Fundamentación Teórica: Contiene los aspectos esenciales para entender el entorno del problema a resolver. Se muestran los antecedentes que dieron origen al problema en cuestión, se define el objeto de estudio, la situación problémica, el estado del arte de los sistemas informáticos en la atención primaria que gestionan la información relacionada con la configuración y los aspectos generales del Sistema Nacional de Salud, para comprender la gestión de la información del componente Módulo de Configuración del SIAPS.

CAPÍTULO 2. Características del Sistema: Presenta los argumentos principales de los procesos vinculados al problema a resolver, como son la descripción de los conceptos fundamentales asociados al dominio del problema y la identificación de los requerimientos mediante la especificación de requerimientos funcionales y no funcionales.

CAPÍTULO 3. Diseño del Sistema: Muestra los elementos básicos del diseño del sistema, mediante la justificación del uso de patrones y los diagramas de clases del diseño, las descripciones de las clases más significativas, el diagrama de clases persistentes y el modelo de datos con la explicación de las tablas y atributos.

CAPÍTULO 4. Implementación: Describe los componentes fundamentales de todo el proceso de implementación, argumentando la justificación de la integración con otros módulos, los métodos, estándares de diseño, codificación y excepciones, así como la representación general del diagrama de despliegue.

CAPÍTULO 1. Fundamentación teórica

En este capítulo se exponen los principales aspectos relacionados con el problema a resolver, se analizan los antecedentes que dieron origen al mismo, se define el estado del arte de los sistemas de configuración y sus precedentes en aplicaciones vinculadas a la salud, para comprender la gestión de la información del componente Módulo de Configuración del SIAPS.

1.1 Sistema Nacional de Salud.

La forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado, es lo que se conoce como Sistema Nacional de Salud (SNS). La Organización Mundial de la Salud lo define como: “Un sistema que engloba todas las organizaciones, instituciones y recursos cuyo principal objetivo es llevar a cabo actividades encaminadas a mejorar la salud”. [1]

De acuerdo con la complejidad de las acciones preventivas, curativas y de rehabilitación, así como la especialización de los servicios de salud brindados, los diferentes niveles de atención médica se han organizado en: [2]

- ✓ **Atención Primaria de Salud (APS):** Da solución aproximadamente al 80 % de los problemas de salud de la población y que correspondan con las acciones de promoción y protección de la salud. Aunque sus actividades se realizan en cualquier unidad del SNS, están relacionados fundamentalmente con las que se realizan en clínicas urbanas o rurales, dispensarios y postas médicas.
- ✓ **Atención Médica Secundaria:** Este nivel da cobertura a cerca del 15 % de los problemas de salud, su función fundamental es tratar al hombre ya enfermo, tanto desde el punto de vista individual como colectivo, pero también desempeña funciones de rehabilitación, promoción y prevención de la salud. Se llevan a cabo acciones de salud más complejas y especializadas. Comprende la atención médica brindada en los distintos hospitales.
- ✓ **Atención Médica Terciaria:** El nivel terciario debe abarcar alrededor del 5 % de los problemas de salud, relacionados con secuelas o aumento de las complicaciones de determinadas dolencias.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Se brindan servicios de muy alta complejidad, con la óptima utilización de los recursos y medios existentes en los mismos y el desarrollo de la investigación. A este nivel pertenecen los institutos y hospitales especializados.

El SNS tiene como propósito básico elevar gradualmente la salud de la población, de modo que contribuya al aumento de su calidad de vida, al incremento de la calidad de los servicios, la satisfacción de la población y de trabajadores de la salud y al perfeccionamiento de la gestión y la elevación de la eficiencia en los centros de salud.

1.2 Informatización del Sistema Nacional de Salud.

La informatización del SNS está dada por el conjunto de métodos, técnicas, procedimientos y actividades gerenciales dirigidas al manejo de la información en salud, la cual comprende la información sobre el estado de salud de la población, la información sobre el conocimiento de las ciencias de la salud y la información en general para la toma de decisiones, clínico-epidemiológicas, operativas y estratégicas. [3]

En el año 1997 se comienzan a dar los primeros pasos para la introducción de las TIC en el SNS, trazándose las estrategias para el desarrollo de una infraestructura no solo tecnológica; sino de desarrollo. Como parte del proceso de informatización de la sociedad cubana el SNS ocupa un puesto crítico, por lo que se han destinado cuantiosos recursos, personal y empresas a desarrollar herramientas que cumplan con los objetivos trazados.

1.2.1 Alas SIAPS.

Para la integración de los componentes, servicios o sistemas que se desarrollen en pos de la informatización de la Atención Primaria de Salud, se concibe el Sistema Integral para la Atención Primaria de Salud (alás SIAPS), para atender las necesidades de los clientes del nivel primario de atención de salud como plataforma única para la gestión, procesamiento y transmisión de la información clínica en la APS.

Actualmente los módulos que se encuentran en desarrollo son:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- ✓ **Módulo de Configuración del SIAPS:** Gestiona todas las configuraciones, de temas, roles, usuarios, funcionalidades, módulos, unidades de salud y áreas de salud, personal de salud, ubicación, gestión de codificadores o estándares internacionales, nomencladores médicos, las poblaciones o zonas de APS, así como la seguridad del sistema.
- ✓ **Módulo de Medicina Familiar del SIAPS:** Gestiona los procesos básicos y fundamentales de la APS, es el núcleo del sistema, contiene los datos generales del paciente, Historia Clínica Individual, Historia Clínica Familiar, las planificación-seguimiento y control del paciente, los ingresos en el hogar y seguimiento diario, las remisiones, la captación del recién nacido, la dispensarización del paciente o clasificación en grupos homogéneos para la atención diferenciada del paciente, en la atención primaria de salud.
- ✓ **Módulo de Medios Diagnósticos del SIAPS:** Gestiona la información de los laboratorios clínico y parasitología, microbiología, imagenología, endoscopia, drenaje biliar, electrocardiograma (ECG), laboratorio de alergia, optometría y audiometría en la Atención Primaria de Salud.
- ✓ **Módulo de Clínico Quirúrgico del SIAPS:** Gestiona la información de la planificación-control y seguimiento de las especialidades médicas (Nefrología, Medicina Interna, Cardiología, Angiología, Urología, Dermatología, Endocrinología, Hematología, Reumatología, Otorrino, Ortopedia, Neurología), genética, cirugía menor, cirugía menor ambulatoria, cirugía mayor, quimioterapia, atención integral al paciente diabético y oftalmología en la Atención Primaria de Salud .
- ✓ **Módulo de Enfermería del SIAPS:** Gestiona la información asociada a los vacunatorios, esterilización, las especialidades de enfermería en la atención primaria de salud y centros comunitarios de enfermería en la Atención Primaria de Salud.

1.3 Situación problémica y problema a resolver.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Cuando se definió y comenzó el desarrollo de la herramienta alas SIAPS, se identificaron un grupo de puntos vitales que debían ser analizados como limitaciones y tratados como tal, para lograr el funcionamiento de la misma:

- La existencia de un grupo significativo de procesos distintos que a la vez coexistían compartiendo valiosos datos indispensables.
 - ✓ El SNS cubano no es monolítico, comparte información con múltiples procesos alternos.
 - ✓ Necesidad de acceso en tiempo real a la información crítica.
- Heterogeneidad en el flujo de la información y en la forma que esta es tratada.
 - ✓ Descentralización de la información.
- Actividades comunes vinculadas a distintos procesos.
 - ✓ Manejo independiente de funciones generales por cada sistema.
 - ✓ Duplicidad de funciones elementales.
- Barreras de comunicación desde el punto de vista del lenguaje.
 - ✓ Falta de un idioma común de comunicación entre aplicaciones.
 - ✓ Poca o ninguna utilización de los estándares internacionales
- Seguridad de la información
 - ✓ Información crítica correspondiente a pacientes e instituciones.

Como resultado de todo lo anterior surge el problema a resolver de desarrollar un sistema informático que sirva de soporte, complementando los elementos anteriormente mencionados, a los sistemas en desarrollo por las SIAPS.

1.4 Análisis de las soluciones existentes.

Los sistemas de configuración están estratégicamente diseñados para brindar soporte a un conjunto de aplicaciones con un objetivo específico, el caso de la APS no es distinto a otros procesos, por lo que la aplicación de un sistema de configuración es crítica. Algunos de los elementos analizados a la hora de hablar de configuración son inherentes al negocio o el ámbito en que se desarrolla el sistema, estos conforman la línea base o los principios que definen qué es configuración, incorporando posteriormente los elementos característicos y definiendo la configuración real que se desea manejar.

1.4.2 Sistemas de manejo de configuración en aplicaciones informáticas.

Software para Consultorios Médicos [6]

País de origen: Argentina

Es una aplicación desarrollada por la empresa Grandi y Asociados para el manejo de la información de salud en los consultorios médicos. Esta herramienta posee un módulo de configuración general.

Módulo Configuración Consultorios

Mediante menú especial de mantenimiento se podrán incorporar nuevos médicos, configurar los colores de pantalla por profesional, cargar nuevos tratamientos, medicamentos, diagnósticos; dando flexibilidad al software, permitiendo así adaptarlo a los requerimientos de cada consultorio, clínica o centro de salud.

Características destacadas:

- Incorporación de nuevos médicos y modificación de datos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Codificación internacional de enfermedades.
- Diagnósticos.
- Estudios médicos.
- Listado de intervenciones quirúrgicas.
- Tratamientos médicos.
- Obras sociales y mutuales.
- Códigos de funciones.
- Medicamentos.
- Laboratorios.

Entre sus limitantes se presentan las siguientes características:

- ✓ Su estructura no completa las funcionalidades necesarias para satisfacer las características del SNS cubano.
- ✓ Imposibilidad de modificación del sistema (Software Propietario).

SIGHO (Sistema de Información para la Gerencia Hospitalaria):

País de origen: México.

Es un software basado en la Norma Oficial Mexicana NOM-168-SSA1-1998 referente al resguardo y uso del expediente clínico electrónico para facilitar las actividades de gerencia dentro del hospital y se apoya en estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos tales como el CIE-10 y CIE9MC. Permite realizar registros individuales alrededor de la historia clínica electrónica en cada uno de los módulos que componen al SIGHO relacionados con la atención al paciente

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Se compone de 14 módulos (2 administrativos y 12 relacionados con la atención al paciente)

El módulo de configuración contiene las opciones de configuración de parámetros necesarios para el correcto funcionamiento de los módulos del SIGHO, concentra los catálogos de información que en común requieren los módulos como son Estados, Jurisdicciones, Municipios, Localidades, Servicios y Áreas Médicas, etc.

Permite llevar a cabo la administración de usuarios que implica la creación de las cuentas de los usuarios que utilizarán el sistema, su clave de acceso y privilegios sobre opciones y operaciones del sistema. El objetivo de este módulo es configurar la información general necesaria para trabajar con el SIGHO. Este módulo es el primero que se debe ejecutar y en el cual se realizarán las operaciones básicas de la configuración para la correcta utilización del SIGHO.

Sus limitantes están dadas por las siguientes características:

- ✓ Su estructura no completa las funcionalidades necesarias para satisfacer las características del SNS cubano.
- ✓ Imposibilidad de modificación del sistema (Software Propietario).

Estos sistemas no son factibles para solucionar el problema analizado pues no cumplen con la totalidad de las funcionalidades que se requieren y son sistemas propietarios, además que todas las aplicaciones referentes al Sistema Nacional de Salud deben converger en una única red usando de esta manera herramientas de desarrollo fundamentalmente libres o código abierto. Esto involucra grandes ventajas; el uso de las tecnologías libres es económico y el bajo coste o nulo de los productos libres permite ampliar sus infraestructuras sin que se vean mermados sus intentos de crecimiento por no poder hacer frente al pago de grandes cantidades dado el costo de las licencias.

Sistema de Información Hospitalaria (alás HIS)

Este sistema está orientado a satisfacer las necesidades de generación de información para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria. Permite la optimización de los recursos humanos y materiales, además de minimizar los inconvenientes burocráticos que afrontan los pacientes. Su principal función es la de apoyar las actividades en los niveles operativos, tácticos y estratégicos dentro de un hospital, haciendo uso de las computadoras para obtener, almacenar, procesar y comunicar información clínica y administrativa.

Esta aplicación aún en desarrollo tiene entre sus módulos un Módulo de Configuración cuya función fundamental es:

- ✓ Gestionar la información general necesaria para trabajar con el Sistema de Información Hospitalaria alás HIS.
- ✓ Gestionarla información relacionada con la administración de usuarios que implica la creación de cuentas de usuarios que utilizarán el sistema, clave de acceso y privilegios sobre las opciones y operaciones del sistema.
- ✓ Actualización de la información referente a los servicios y departamentos con que cuenta el hospital, las posibles referencias a otras instituciones por servicios no disponibles y el control de las camas y ubicaciones; permitiendo así una mayor flexibilidad ante cambios funcionales en el sistema y ofreciendo una visión general de la entidad.

Este módulo comparte un grupo de funcionalidades que son capaces de satisfacer algunas de las necesidades del Módulo Configuración del Subsistema Web para el Sistema Integral para la Atención Primaria de Salud alás SIAPS, permitiendo:

- ✓ Gestionar las Entidades.
- ✓ Gestionar Personal de Salud.
- ✓ Gestionar ATC (Clasificación Anatómica, Terapéutica y Química).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- ✓ Gestionar los Estados del Paciente.
- ✓ Gestionar Grupos Etareos. (Revisar si no es etarios)
- ✓ Gestionar Seguridad.

Por lo que se decide heredar este conjunto de componentes para agilizar el desarrollo de la aplicación.

1.5 Tendencias y tecnologías.

Los elementos tecnológicos como lenguaje, sistema gestor de bases de datos (SGBD), arquitectura, patrones, herramientas y metodologías, relacionados con las políticas anteriormente mencionadas y seleccionados para la construcción del sistema, forman parte de las orientaciones del CESIM para el desarrollo de aplicaciones médicas.

1.5.2 Internet.

Internet es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP. Garantiza que redes físicas heterogéneas funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California y una en Utah, EE.UU.

Internet ha revolucionado al mundo y probablemente sea el invento más importante del siglo XX. Este fenómeno ha venido a dinamizar la forma de vida de millones de personas que hoy en día utilizan la red en diferentes actividades cotidianas. Tiene un impacto profundo en el trabajo, el ocio y el conocimiento a nivel mundial. Gracias a la web, millones de personas tienen acceso fácil e inmediato a una cantidad extensa y diversa de información en línea. [7]

Debido a la concepción misma del Módulo Configuración del Subsistema Web para el Sistema Integral para la Atención Primaria de Salud al SIAPS, el uso de la red y la Internet como su expresión máxima es crítico. Las potencialidades de este sistema descansan en su adaptabilidad y posibilidad de expansión, brindadas por su concepción y el uso de la red.

1.5.3 Software Libre

El término se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software: [8]

1. La libertad de usar el programa, con cualquier propósito (libertad 0).
2. La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
3. La libertad de distribuir copias, con lo que se puede ayudar al vecino (libertad 2).
4. La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, se debería tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no se tiene que pedir o pagar permisos. [9]

La política del MINSAP y del CESIM está enfocada en todas las soluciones informáticas que se desarrollen en el uso de las potencialidades del Software Libre, aspecto que no limita la comerciabilidad de los productos y favorecen el desarrollo.

Sistema Operativo Linux.

Linux es la denominación de un sistema operativo y el nombre de un núcleo. Es uno de los paradigmas del desarrollo de software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona puede libremente usarlo, modificarlo y redistribuirlo.

Es un sistema operativo descendiente de UNIX. UNIX es un sistema operativo robusto, estable, multiusuario, multitarea, multiplataforma y con gran capacidad para gestión de redes. Linux fue creado siguiendo estas características, por Linus Torvalds, con la colaboración de cientos de asociados. Se distribuye bajo la Licencia Pública General de GNU. [10]

1.5.4 Servicios Web.

Clarifican el sentido de accesible y precisan más qué quiere decir orientado a Internet. Son componentes que se pueden integrar en sistemas distribuidos complejos. Una aplicación software identificada por una URI, cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Un servicio Web soporta interacciones directas con otros agentes software utilizando mensajes XML intercambiados mediante protocolos basados en Internet.

Un servicio web es un sistema de software diseñado para soportar la interacción máquina a máquina sobre una red. Posee una interfaz que las máquinas pueden procesar WSDL (del inglés, Web Services Description Language, Lenguaje de Descripción de Servicios Web), está disponible en un URL (del inglés, Uniform Resource Locator, Localizador Uniforme de Recursos) y es accesible a través del HTTP (del inglés, HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto). Un servicio web viene siendo una rutina en Internet. [11]

Entre sus ventajas se pueden denotar las siguientes:

- ✓ Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- ✓ Los servicios web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- ✓ Al apoyarse en HTTP, los servicios web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- ✓ Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos, puedan ser combinados fácilmente para proveer servicios integrados.
- ✓ Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Los servicios web soportan una serie de protocolos como FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol) y HTTP (Hypertext Transfer Protocol), siendo este último el más usado debido a que el puerto 80 casi siempre está abierto, esto permite el acceso a ellos al igual que a una página web. La diferencia entre una página y un servicio web, es que la página la visita cualquier individuo interesado, mientras que el servicio lo visitan además aplicaciones que lo requieren. Las rutinas de los servicios web se actualizan de forma transparente para el programador y para el encargado de mantenimiento de la aplicación. [12]

Mediante un servicio web se pueden implementar a un programa, funciones imposibles de contemplar bajo el uso de rutinas de librerías, como por ejemplo, incorporar un buscador de páginas web. Por otro lado, la carga de CPU (Unidad central de procesamiento) que supone la ejecución de una rutina, desaparece al usar servicios web. La carga se reparte por Internet, sobre el servidor del servicio web. Esto es un comienzo de "Computación Distribuida". [13]

Es necesario reconocer, que los Servicios Web, van a ser utilizados principalmente por programadores Web (PHP, Perl, ASP, JSP...) ya que su máxima funcionalidad se da casi siempre ligada a Internet. [14]

1.6 Arquitectura de Software.

La Arquitectura de Software aporta una visión abstracta de alto nivel, consiste en un conjunto de patrones y abstracciones coherentes, que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. Establece los fundamentos para que los analistas, diseñadores, programadores y otros trabajen en una línea común, que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.

En la arquitectura de software se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, otros pueden ser: la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información, y las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información.

1.6.2 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.

La arquitectura cliente/servidor, es una forma de dividir y especializar programas y equipos de cómputo, a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema. En una arquitectura monolítica no hay distribución; los tres niveles tienen lugar en el mismo equipo, en el modelo cliente/servidor, en cambio, el trabajo se reparte entre dos ordenadores. [15]

Ventajas del modelo Cliente/Servidor.

- ✓ El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- ✓ Se reduce el tráfico de red considerablemente. El cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.
- ✓ El sistema es fácil de escalar.

1.6.3 Arquitectura en 3 capas.

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos, como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración. Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema. [16]

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Ventajas de la Arquitectura en 3 capas.

- ✓ Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- ✓ No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costos de mantenimiento.
- ✓ Mayor sencillez de los clientes.

La arquitectura de tres capas es la más común en sistemas de información que además de tener una interfaz de usuario, contemplan la persistencia de los datos. Cada capa o nivel es un proceso separado y bien definido corriendo en plataformas separadas. Las tres capas o niveles son: presentación, negocio y almacenamiento. **(Ver Anexo 2).**

Capa de Presentación: Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando 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. Comprende las responsabilidades de lógica de presentación:

- ✓ Navegabilidad del sistema.
- ✓ Validación de datos de entrada.
- ✓ Formateo de los datos de salida.
- ✓ Internacionalización.
- ✓ Renderizado de presentación.

Capa de Negocio: Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio)

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

pues 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, almacenar o recuperar datos de él. Esta capa comprende las siguientes responsabilidades: [17]

- ✓ Las responsabilidades de lógica de negocio (o dominio) del sistema.
- ✓ Resultado del análisis funcional:
 - Conjunto de reglas de negocio que abstraen el mundo real.
- ✓ La capa de negocio ha de ser independiente de la capa de presentación y viceversa (en la medida de lo posible).

Capa de Datos: Es donde residen los datos. Está formada por uno o más sistemas gestores de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Comprende las responsabilidades de lógica de persistencia de las entidades que maneja el sistema en desarrollo. [18]

1.6.4 Desarrollo de Software Basado en Componentes (DSBC).

La complejidad de los sistemas computacionales actuales ha llevado a buscar la reutilización del software existente. El desarrollo de software basado en componentes permite reutilizar piezas de código pre elaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión. [19]

Los componentes son los "ingredientes de las aplicaciones", que se juntan y combinan para llevar a cabo una tarea. El paradigma de ensamblar componentes y escribir código para hacer que estos componentes funcionen se conoce como Desarrollo de Software Basado en Componentes. [20]

Los modelos y plataformas de componentes proporcionan los mecanismos adecuados para poder tratar la complejidad de los problemas que aparecen en la construcción de aplicaciones para sistemas abiertos y

distribuidos, como pueden ser la heterogeneidad, la sincronización, la dispersión, fallos en la seguridad, retrasos y problemas en las comunicaciones, la composición tardía y otros.

Beneficios del Desarrollo de Software Basado en Componentes

El uso de este paradigma posee algunas ventajas: [21]

- ✓ Reutilización del software. Permite alcanzar un mayor nivel de reutilización de software.
- ✓ Simplifica las pruebas. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
- ✓ Simplifica el mantenimiento del sistema. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
- ✓ Mayor calidad. Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

1.6.5 Arquitectura Orientada a Servicio (SOA).

La Arquitectura Orientada a Servicio (SOA, Service Oriented Architecture) establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular. [22]

La Arquitectura Orientada a Servicios es una filosofía de diseño que permite un mejor alineamiento de las Tecnologías de Información (IT) con las necesidades de negocio, permitiendo a empleados, clientes y

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

socios comerciales responder de forma más rápida y adaptarse adecuadamente a las presiones del mercado. [23]

Entre los beneficios de SOA se destacan: [24]

- ✓ La reducción de costos y tiempo en el desarrollo de aplicaciones ya que SOA permite reutilizar los módulos de aplicaciones existentes y el código nuevo para generar nuevas aplicaciones. Como consecuencia también se reducen los costos mantenimiento.
- ✓ Las metodologías que aterrizan el concepto de SOA facilitan la integración entre aplicaciones nuevas así como con los sistemas existentes.
- ✓ Desarrollo de aplicaciones más productivas, flexibles, más seguras y manejables para gestionar procesos de negocio críticos a medida que evolucionan o cambian las necesidades del negocio.
- ✓ Fortalecimiento y consolidación de los procesos de negocio a través de aplicaciones que comparten servicios comunes.

1.7 Servidor de Aplicación.

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, y esto hace que el desarrollo de las aplicaciones del empresario sean mucho más simples. [25]

Una de las facilidades que este servidor presenta es que puede ser instalado sobre varios ambientes, tales como Windows o GNU/Linux.

1.8 Sistema Gestor de Base de Datos.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones alcanza los niveles de otras bases de datos comerciales. [26]

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

La última serie de producción es la 8.4, siendo la última versión disponible en el momento actual la 8.4.2. Sus características técnicas lo hacen uno de los gestores de bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. Postgres SQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

1.9 Frameworks, librerías y componentes.

Los Frameworks se pueden considerar como soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). Aceleran el proceso de desarrollo, permiten reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

1.9.2 Java Server Faces (JSF)

La tecnología Java Server Faces es un marco de desarrollo de los componentes de la interfaz de usuario del lado del servidor y es válido para todas aquellas aplicaciones web basadas en la tecnología JAVA. JSF

funciona traduciendo las distintas acciones del usuario en eventos que son respondidos por el servidor regenerando la página original y reflejando los cambios necesarios para la acción realizada.

1.9.3 RichFaces

RichFaces es un framework de código abierto que añade capacidad JavaScript asíncrono y XML Ajax dentro de aplicaciones JSF existentes sin recurrir a JavaScript. RichFaces incluye ciclo de vida, validaciones, conversores y la gestión de recursos estáticos y dinámicos. Los componentes de RichFaces están contruidos con soporte Ajax y un alto grado de personalización que puede ser fácilmente incorporado dentro de las aplicaciones JSF.

Permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes listos para usar, altamente configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia, lo que mejora grandemente la experiencia de usuario

1.9.4 Ajax.

El framework de extensión Ajax, que es el acrónimo para Asynchronous JavaScript + XML y el concepto es: cargar y renderizar una página, luego mantenerse en esa página mientras los scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma. [27]

Ajax incorpora: [28]

- ✓ Presentación basada en estándares usando XHTML y CSS.
- ✓ Conversión y validación de campos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- ✓ Establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario, un modelo orientado a eventos y, combinado con Facelets, se elimina la necesidad de motores de renderización, mejorando el rendimiento en general además de que brinda la capacidad añadida de la tecnología de plantillas de Facelets.

1.9.5 Ajax4JSF.

Ajax4jsf es una librería código abierto (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. [29]

1.9.6 Java script.

Es un lenguaje de scripting basado en objetos no tipado y liviano, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

1.9.7 Facelets.

Facelets es un framework simplificado de presentación en donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF entre otras cosas. Permite crear plantillas para construir un árbol de componentes de forma que se puedan definirse como composición de otros.

1.9.8 JBoss Seam.

JBoss Seam es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB). Con Seam basta agregar anotaciones propias de éste a los objetos Entidad y Session de EJB, logrando con esto escribir menos código Java y XML. Otra característica importante es

que puedes hacer validaciones en los POJOs (Plain Old Java Object) como además manejar directamente la lógica de la aplicación y de negocios desde sus sessions beans.

Seam provee una mayor granularidad de contextos de estado. La principal, quizás es el contexto conversacional, así como el asociado a procesos del negocio, con estos se logra un uso más eficiente de la memoria evitando memory-leaks. Integra además el concepto de workspaces permitiendo que el usuario tenga en varios tabs o ventanas del navegador actividades del negocio con contextos completamente aislados. Seam integra transparentemente la administración de procesos del negocio vía jBoss jBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones (workflows) y complejas interacciones con el usuario (pageflows). [30]

1.9.9 Hibernate.

Hibernate es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación mediante archivos declarativos (XML) que permiten establecer estas relaciones. Genera sentencias SQL a partir de sentencias HQL (Hibernate Query Language). A través de la implementación del estándar JPA que provee Hibernate 3.3, se puede realizar el acceso a datos.

Las características más importantes de las colecciones en Hibernate son: [31]

- ✓ Colecciones ordenadas: Hibernate soporta la implementación de colecciones ordenadas a través de los interfaces `java.util.SortedMap` y `java.util.SortedSet`. Si se desea, se puede definir un comparador en la definición de la colección. Los valores permitidos son `natural`, `unsorted` y el nombre de la clase que implementa `java.util.Comparator`.
- ✓ El colector de basura de las colecciones: Las colecciones son automáticamente persistidas cuando son referenciadas por un objeto persistente y también son borradas automáticamente cuando dejan de serlo.

1.10 Tecnologías.

1.10.2 CSS

Hojas de estilo cascada (Cascading Style Sheets) es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas.

La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. [32]

1.10.3 HTML

(Lenguaje de Marcado de Hipertexto) es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. [33]

1.10.4 XHTML

(Lenguaje de Marcado de Hipertexto Extensible) es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0. Los tipos de documentos de la familia XHTML están basados en XML, y diseñados fundamentalmente para trabajar en conjunto con aplicaciones de usuario basados en XML. [34]

1.10.5 XML

Son las siglas de Extensible Markup Language, una especificación/lenguaje de programación desarrollada por el W3C (World Wide Web Consortium). XML es una versión de SGML (Standard Generalized Markup Language), diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones. [35]

El lenguaje unificado EL (Expression Language) utilizado en JSF está especialmente diseñado para soportar el sofisticado modelo de componentes de interfaz de usuario, que permite realizar validaciones y conversiones, propagar datos de los componentes a los objetos y recoger los eventos de los componentes. Con lo cual para soportar todo esto el lenguaje ofrece estas funcionalidades:

- ✓ Evaluación diferida de expresiones
- ✓ Invocación de métodos
- ✓ Recoger y asignar datos.

1.11 Lenguajes de Programación.

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

El lenguaje Java tiene como ventaja que es un lenguaje multiplataforma. Este es un lenguaje que se ha extendido, cobrando, cada día, más importancia tanto en el ámbito de Internet como en la informática en general. [36]

Java permite programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema, facilita la integración de múltiples frameworks que ofrecen muchas facilidades: persistencia, seguridad, logs, internacionalización, BPM, testing, tareas asincrónicas, web services, pantallas con elementos ricos de interfaz y relativa facilidad de uso: múltiples implementaciones de JSF por Apache MyFaces, Oracle ADF, RichFaces.

1.12 Metodologías de Desarrollo del Software.

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 CMMI 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".

Dichas organizaciones pueden también ser evaluadas por áreas de proceso en vez de por niveles de madurez, al adquirir los niveles de capacidad en cada una de las Áreas de Proceso, obteniendo el "Perfil de Capacidad" de la organización. A esta visión de la organización se le conoce como "Representación Continua."

Cada área de proceso dentro de los niveles de capacidad posee un conjunto de objetivos genéricos y específicos. Los objetivos genéricos asociados a un nivel de capacidad establecen lo que una organización debe alcanzar en ese nivel de capacidad. El logro de cada uno de esos objetivos en un área

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

de proceso significa mejorar el control en la ejecución de la misma. Estos objetivos tienen un conjunto de prácticas genéricas que se aplican a cualquier área de proceso porque puede mejorar el funcionamiento y el control de cualquier proceso.

Los objetivos específicos se aplican a una única área de proceso y localizan las particularidades que describen qué se debe implementar para satisfacer el propósito del área de proceso. A su vez, cada objetivo específico está formado por un conjunto de prácticas específicas y sub prácticas. Las prácticas específicas son actividades que se considera importantes en la realización del objetivo específico al cual están asociadas, describen las actividades esperadas para lograr dicho objetivo en un área de proceso y las sub prácticas constituyen descripciones detalladas que sirven como guía para la interpretación de una práctica genérica o específica..

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.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

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.

Ciclo de vida básico:

Estudio Preliminar: Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo.

Modelación del Negocio: Se comprende como funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para la descripción y modelado de negocio pueden ser utilizadas diferentes técnicas como el Modelado de Casos de Uso del Negocio y Business Process Modeling Notation (BPMN).

Requisitos: Se desarrolla un modelo del sistema que se va a construir. Incluye un conjunto de casos de uso, servicios que describen todas las interacciones que tendrán los usuarios con el software, estos responden a los requisitos funcionales del sistema.

Análisis y Diseño: Se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados en esta etapa son más formales y específicos de una implementación. Durante esta fase son desarrollados el documento de arquitectura, diagramas de clases, diagramas de entidad relación, diagrama de despliegue entre otros.

Implementación: Se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares, a partir de los resultados del análisis y diseño.

Pruebas Internas: Se verifica el resultado de la implementación, probando según sea necesario, cada construcción incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Pruebas de Liberación: Se llevan a cabo las pruebas diseñadas e implementada por el Laboratorio Industrial de Pruebas de Software a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

Despliegue: Se procede a la entrega de la solución, así como a la instalación, configuración, prueba y puesta en marcha del software en el entorno real del cliente. También deben realizarse en este periodo la capacitación y acompañamiento a clientes para asegurar que adquieran los conocimientos necesarios en la manipulación del software.

Soporte: Se ofrece un servicio para resolver conflictos y problemas de usabilidad y rendimiento del software entregado al cliente, suministrándole actualizaciones y parches a errores.

La descripción del IPP-3510:2009 Libro de Proceso para la Administración de Requisitos 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 (RUP).

El Proceso Unificado de Desarrollo de Software (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 finalizan 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 IPP- 3510:2009 Libro de Proceso para la Administración

de Requisitos se utiliza UML, con la especificación del uso del estándar BPMN para la descripción de los procesos de negocio.

1.12.2 ¿Qué es un Proceso de Negocio?

Un proceso de negocio consiste en una colección de actividades que son realizadas coordinadamente en un ambiente técnico y organizacional. La conjunción de estas actividades logra un objetivo del negocio. Los procesos de negocio toman una o más tipos de entradas (precondiciones) y crea una salida (pos condición) que posee un determinado valor para el cliente. [39]

1.12.3 ¿Qué es BPM?

BPM significa Business Process Management o Gestión de procesos de negocios, es una metodología empresarial para el modelado, integración, monitoreo y mejora continua de procesos de negocios a través de la gestión sistémica de los mismos. [40]

1.12.4 ¿Qué es un software BPM?

Un software BPM permite el modelado, documentación, ejecución y monitoreo de los procesos de negocios de su organización. [41]

1.12.5 Notación Utilizada para Modelar los Procesos del Negocio.

Notación para el Modelado de Procesos de Negocio (BPMN) es un nuevo estándar de que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. 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. Su principal objetivo es proveer una notación que sea fácilmente entendida por todos los usuarios, desde el analista de negocio, el desarrollador técnico y los administradores del negocio.

1.12.6 Lenguaje Unificado de Modelado (UML). [45]

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un modelo del sistema, incluyendo tanto los aspectos conceptuales tales como procesos de negocio así como los aspectos concretos como expresiones de lenguajes de programación y otros. Permite una integración fuerte entre las herramientas, los procesos y los dominios y no precisa un proceso de desarrollo determinado.

1.13 Herramientas a utilizar. [46]

Toda metodología de desarrollo de sistemas debe sustentarse sobre la base de una herramienta case que garantice un mejor análisis del sistema y la calidad del resultado final, a su vez estas herramientas proporciona el potencial para mejorar la productividad del analista, facilitan el proceso de desarrollo más eficaces y mejorar la calidad del sistema.

La herramienta seleccionada para el sustento de la metodología fue Visual Paradigm suite 3.1, esta Herramienta CASE da soporte al modelado visual de UML 2.1 y BPMN.

La herramienta escogida para el modelado de los procesos de negocio fue Business Process Visual ARCHITECT 2.1 Analyst Edition (BP-VA) la cual forma parte de la suite de Visual Paradigm 3.1. Es una herramienta que soporta las últimas notaciones de modelo de negocios OMG's (BPMN). BP-VA está diseñada para visualizar, comprender, analizar, mejorar y documentar los procesos de negocio, flujo de documentos y la información en la organización. Permite la generación de un código BPEL compatible con JBPM de JBoss que es de vital importancia para el desarrollo.

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 6.1 Enterprise Edition. Es válido aclarar que se utiliza la versión 6.1 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.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Conclusiones

En este capítulo se expusieron las principales características de la problemática existente, así como de los sistemas desarrollados relacionados, se analizaron las principales tecnologías y herramientas propuestas a nivel mundial para el desarrollo de este tipo de software, concluyendo que :

- ✓ El producto alas SIAPS no tiene forma de manejar eficientemente la interoperabilidad tanto semántica como sintáctica de sus módulos.
- ✓ No existe en una aplicación que de solución íntegra al problema propuesto.
- ✓ El uso de funcionalidades gestionadas por alas HIS facilitará el proceso de desarrollo del módulo.
- ✓ La tecnología propuesta generará una primera versión robusta y actualizada del sistema.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Capítulo 2. Características del Sistema

En este capítulo se presentan los argumentos principales de los procesos vinculados al problema a resolver, se identifican y describen los principales conceptos asociados al dominio del problema y son representados mediante el Modelo del Dominio, se refleja también el resultado del levantamiento de Requerimientos, a través de la especificación de requerimientos funcionales y no funcionales.

2.1 Modelo de Dominio

Es el primer paso en el proceso de desarrollo de software es alcanzar un alto nivel de conocimientos sobre el problema en cuestión. En este punto los especialistas tienen que interactuar con personal que, aunque perfectamente capacitado en su campo de acción, tiene poco o ningún dominio de la terminología informática, son los ingenieros informáticos los que deben ser capaces de entrar en su entorno e identificar el problema para luego ofrecer una solución eficiente.

En el caso de establecer la configuración de un sistema para la APS, la definición del negocio se torna compleja debido a que el mismo no se encuentra claramente definido y sus fronteras no están bien delimitadas. Por lo anteriormente planteado se identifican solamente un conjunto de conceptos los cuales son descritos mediante un Modelo de Dominio, que es un subconjunto del Modelo de Negocio y constituye un modelo conceptual donde se destacan como componentes fundamentales los conceptos y las relaciones entre ellos, el mismo puede considerarse además como un diccionario visual de las abstracciones relevantes, con vocabulario e información estrictamente del dominio.

2.1.1 Conceptos Fundamentales.

Dominio Área de Salud:

Área de Salud:

Estructura fundamental del sistema sanitario, responsable de la gestión unitaria de los centros y establecimientos del servicio de salud en su demarcación territorial y de las prestaciones sanitarias y programas sanitarios por ellos desarrollados.

Centro Laboral:

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

El concepto hace referencia a los centros laborales que se encuentran dentro de un área de salud.

Dirección:

El concepto hace referencia a la ubicación lógica (calles, entre calles, números, etc) de las viviendas y centros laborales de un área de salud.

Zona de APS:

El concepto hace referencia a zonas básicas (territorio de influencia de un centro de salud) de un área de salud.

Localidades por Áreas:

El concepto hace referencia a las localidades atendidas por un área de salud.

Tipo de Centro:

El concepto hace referencia a los tipos de centros laborales que pueden existir.

Tipo de Zona de APS:

El concepto hace referencia a los tipos de zonas básicas de un área de salud en dependencia de si comprenden viviendas o centros laborales.

Ubicación de Zona de APS:

El concepto hace referencia a la ubicación geográfica de las zonas básicas de un área de salud.

Unidades del Área:

El concepto hace referencia a las unidades de salud donde se prestan servicios médicos que abarca un área de salud.

Clasificación Internacional de Atención Primaria:

Taxonomía de los términos y expresiones utilizadas habitualmente en medicina general. Recoge los motivos (o razones) de consulta, los problemas de salud y el proceso de atención. Es un tipo de clasificación de terminología médica de ámbito internacional.

CIAP:

Representa los códigos de la clasificación CIAP.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

CIAP Capítulo:

Representa los capítulos de la clasificación CIAP que hacen referencia a un aparato o sistema orgánico.

CIAP Componente:

Representa los componentes de la clasificación CIAP que hacen referencia específica o inespecíficamente a signos o síntomas; procedimientos administrativos, diagnósticos, preventivos o terapéuticos; resultados de pruebas complementarias; derivaciones, seguimiento y otras razones de consulta; o enfermedades y problemas de salud.

CIAP Subcomponente:

Representa los subcomponentes de la clasificación CIAP.

CIAP Abreviatura:

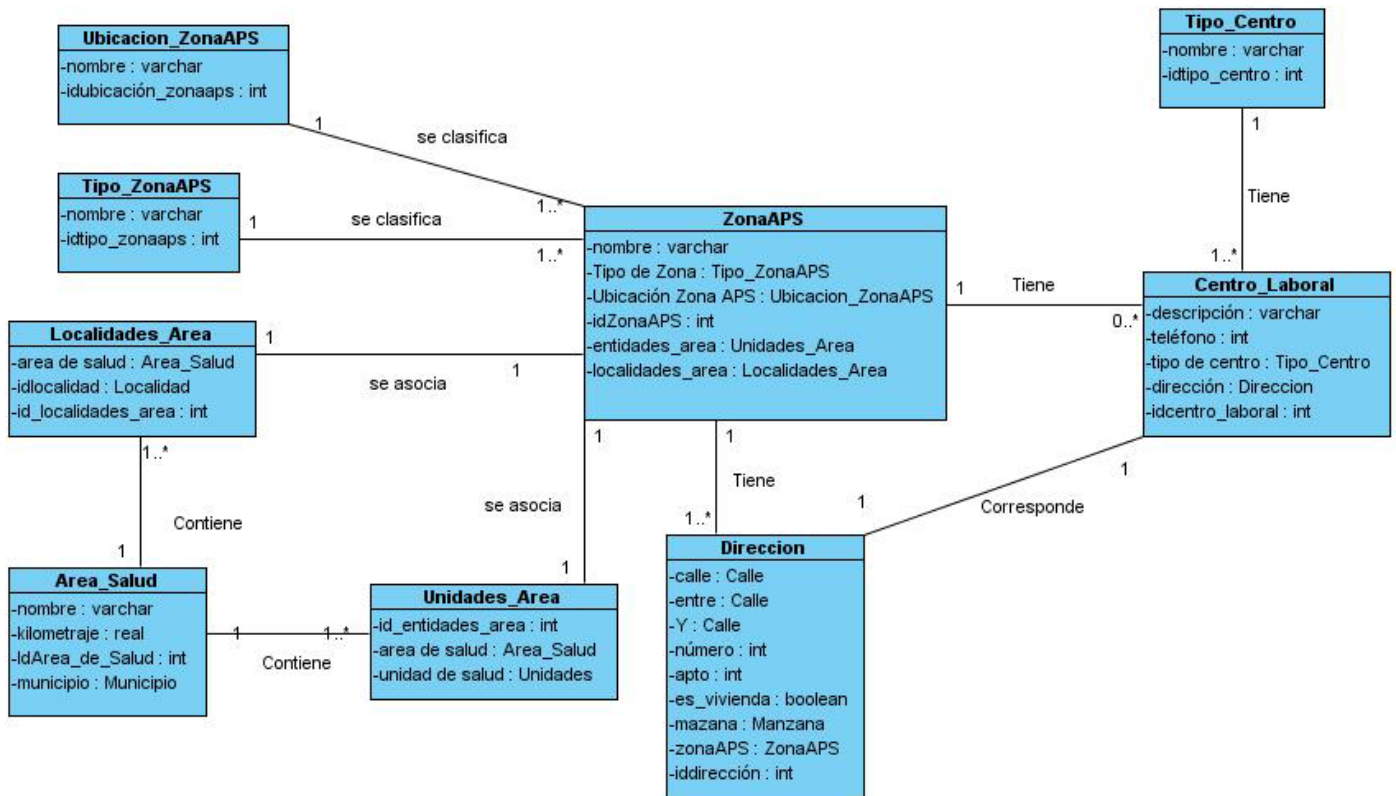
Representa las abreviaturas definidas para referirse a los términos de CIAP.

2.2 Diagrama del Modelo de Dominio

Un diagrama del modelo de dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un diagrama que describe clases de software u objetos software con responsabilidades, se trata de una representación de conceptos puramente del dominio, donde se establece una terminología estrictamente del dominio. Aprovechando las bondades de los diagramas UML para representar cosas y conceptos, el diagrama del modelo del dominio se presenta en forma de diagrama de clases, donde figuran los principales conceptos y roles del sistema analizado. A continuación se muestran algunos de los Diagramas de Modelo de Dominio de cada uno de los grupos definidos dentro del ámbito de la configuración:

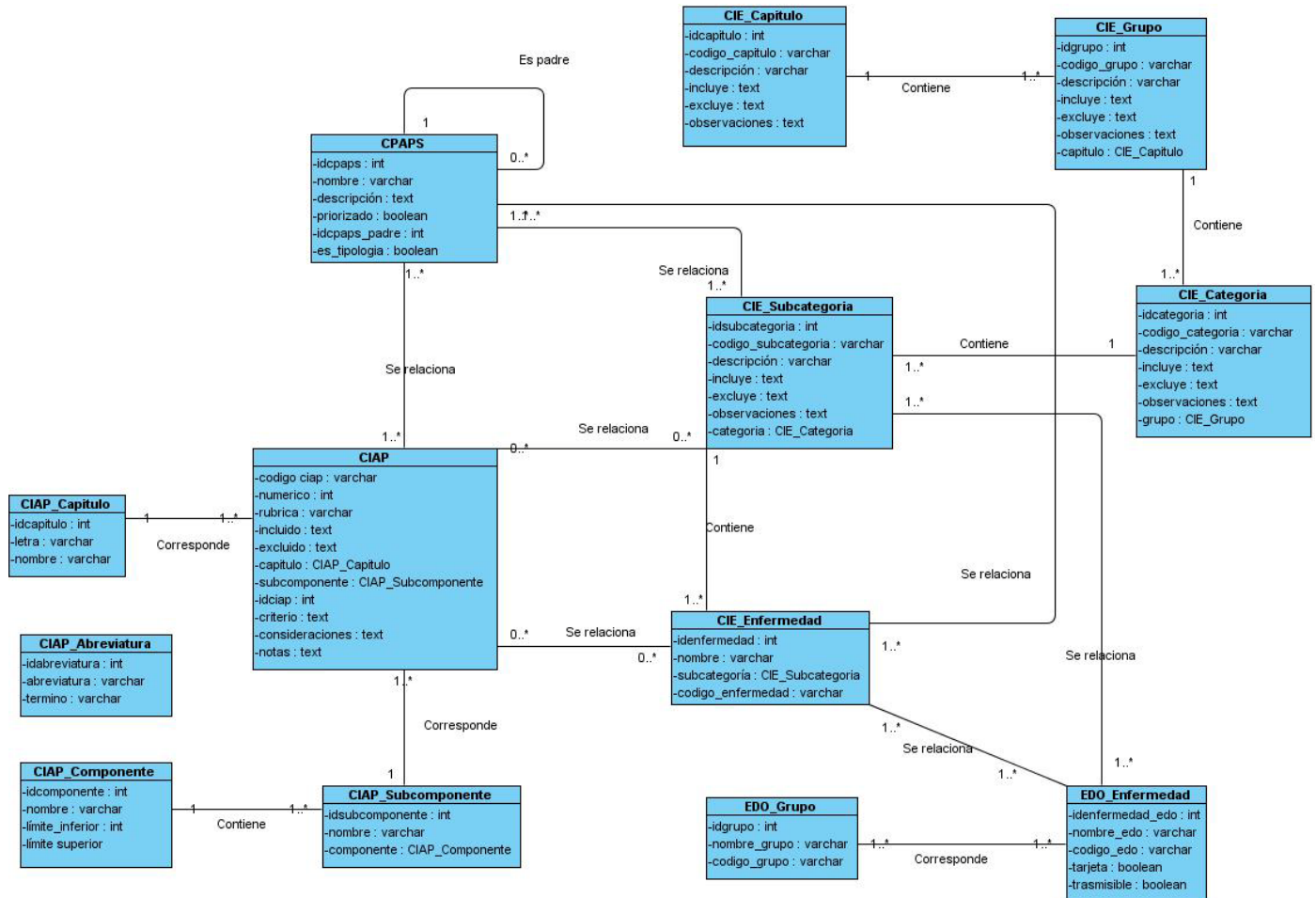
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Modelo de Dominio Área de Salud:



CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Modelo de Dominio Nomencladores Asistenciales:



Se desarrolló otro modelo de dominio:

- ✓ Ubicación Geográfica.

2.3 Propuesta de Sistema

Establecer un punto de partida desde un análisis claro de las necesidades del cliente es vital para el buen desarrollo de la aplicación, cuando un producto se entrega sin haber dado este paso y surgen cambios, los costos se disparan, lo que genera problemas con los clientes. Manteniendo una comunicación con los

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

beneficiarios del trabajo, no solamente se logra acelerar el proceso de desarrollo y minimizar los costos de producción; sino hacer que el equipo funcione mejor y aumente su potencial de desarrollo. Decidir que se va a construir es el punto crítico en el desarrollo de un software, no solamente lo convierte en el contrato con el cliente; sino que define la meta del esfuerzo.

2.3.2 Especificación de Requerimientos de Software.

El proceso para la especificación de requerimientos se encarga de establecer los servicios que un sistema debe suministrar, así como las restricciones bajo las cuales debe operar el mismo, siendo la meta final construir un documento con la Especificación de Requerimientos de Software (ERS). El papel de la ERS es esencial para el logro de un producto de software correcto, cualquiera que sea su área de aplicación.

2.3.2.1 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, los mismos no alteran la funcionalidad del producto. A partir del estudio e investigaciones realizadas en el negocio del problema, se obtuvieron los siguientes requerimientos funcionales que ha de cumplir el sistema, con vista a la satisfacción final del cliente:

No	Requisitos Funcionales	Descripción
RF1	Configurar Área de Salud	Se agregan los datos de una Área de Salud, se buscan los datos, se eliminan se muestran y se modifican.
RF2	Gestionar Dirección.	Se agregan los datos de una Dirección, se buscan los datos, se eliminan se muestran y se modifican.
RF3	Gestionar Zona APS	Se agregan los datos de una Zona APS, se buscan los datos, se eliminan se muestran y se modifican.
RF4	Gestionar Centro Laboral	Se agregan los datos de un Centro Laboral, se buscan los datos, se eliminan se muestran y se modifican.
RF5	Gestionar Codificador Tipo de Zona	Se agregan los datos de un Tipo de

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

		Zona, se listan, se eliminan se muestran y se modifican.
RF6	Gestionar Codificador Ubicación Zona	Se agregan los datos de un Tipo de Zona, se listan, se eliminan se muestran y se modifican.
RF7	Gestionar Codificador Tipo de Centro	Se agregan los datos de un Tipo de Centro, se listan, se eliminan se muestran y se modifican.
RF8	Gestionar Datos Capítulos CIAP	Se agregan los datos de un Capítulo, se buscan los datos, se eliminan se muestran y se modifican.
RF9	Gestionar Datos Componente	Se agregan los datos de un Componente, se buscan los datos, se eliminan se muestran y se modifican.
RF10	Gestionar Datos Subcomponente	Se agregan los datos de un Subcomponente, se buscan los datos, se eliminan se muestran y se modifican.
RF10	Gestionar Datos Abreviatura	Se agregan los datos de un Abreviatura, se buscan los datos, se eliminan se muestran y se modifican.
RF11	Gestionar Datos CIAP	Se agregan los datos de un CIAP, se buscan los datos, se eliminan se muestran y se modifican.

2.3.2.2 Requerimientos no Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez 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. En el caso de la configuración los requerimientos no funcionales son vitales para la aceptación del producto, pues están inseparablemente vinculados a su dominio.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

No	Categoría	Requisitos no Funcionales
RNF1	Usabilidad	<p>El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:</p> <ul style="list-style-type: none">• Usuarios normales: 20 días• Usuarios avanzados: 30 días
RNF2	Fiabilidad	<p>En los servidores de los policlínicos y en el Centro de Datos Nacional del MINSAP se garantizará una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos. Se garantizarán además, políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema.</p>
RNF4	Fiabilidad	<p>Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.</p>
RNF5	Fiabilidad	<p>Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.</p>
RNF6	Fiabilidad	<p>Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.</p>
RNF10	Fiabilidad	<p>El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.</p>
RNF11	Eficiencia	<p>El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.</p>

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RNF 15	Soporte	Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados.
RNF18	Soporte	Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.
RNF 19	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.
RNF 21	Interfaz	Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con menús desplegables que faciliten y aceleren su utilización. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español.
RNF 25	Rendimiento	El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.
RNF 26	Rendimiento	El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.
RNF 27	Hardware	En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

		navegador actualizado y que siga los estándares web, se recomienda IE 7, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo Linux.
RNF 28	Hardware	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.
RNF29	Software	El sistema debe correr en sistemas operativos Windows, Unix 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.

Conclusiones

Después de haber analizado el dominio del problema, se identificaron los principales conceptos asociados y los elementos necesarios para conformar los requerimientos funcionales y no funcionales, imprescindibles para la realización del diseño del módulo, obteniéndose así la primera entrada para el desarrollo del mismo.

Capítulo 3. Diseño del Sistema

El propósito del presente capítulo es definir la estructura y elementos del diseño, describir requerimientos funcionales en términos de clases y sus objetos, representándolos gráficamente en Diagramas de Clases, así como fundamentar los patrones de diseño empleados.

3.1 Modelo de Diseño

Constituye el refinamiento del análisis, soporta los requisitos funcionales y no funcionales y las restricciones que suponen, define cómo cumple el sistema sus objetivos. Además, impone una estructura del sistema que se debe conservar lo más fielmente posible cuando se de forma al sistema. Es la entrada al flujo de trabajo de implementación.

Propósitos fundamentales del flujo de trabajo:

- ✓ Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- ✓ Crear una entrada apropiada y un punto de partida para actividades de implementación, mediante la captura de los requisitos o subsistemas individuales, interfaces y clases.
- ✓ Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- ✓ Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando se utilizan interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

3.1.1 Fundamentación del uso de Patrones.

Un patrón es una solución recurrente para un problema en un contexto o la respuesta al problema dentro de un contexto que ayuda a resolver las dificultades. Los Patrones de Diseño son directrices y principios

CAPÍTULO 3. DISEÑO DEL SISTEMA

estructurados que describen un problema común y entregan una buena solución ya probada a la que se le da un nombre. La utilización de los mismos en la concepción del módulo, ayudará a diseñar correctamente en menos tiempo, a construir clases reutilizables, facilitará la documentación y conducirá a la definición de una arquitectura pequeña, simple y comprensible.

Modelo Vista Controlador.

La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño que mezcle los componentes de interfaz y de negocio, entonces los cambios que se realicen en las interfaces producirán modificaciones en los componentes de negocio, lo que proporcionará mayor trabajo y más riesgo de errores. El módulo se encuentra diseñado en las tres capas correspondientes al MVC:

- Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma.
- Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación Web, la "Vista" es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.
- Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario.

Patrones de Asignación de Responsabilidades (GRASP).

Experto.

Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para

CAPÍTULO 3. DISEÑO DEL SISTEMA

hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece el hecho de tener sistemas más robustos y de fácil mantenimiento.

Alta Cohesión.

La cohesión es la medida de la fuerza que une a las responsabilidades de una clase. Una clase con baja cohesión es aquella que hace muchas cosas no afines o muchas tareas, lo que trae como consecuencias dificultades para entender, reutilizar y conservarla. Son delicadas y las afectan constantemente los cambios. Una clase con alta cohesión mejora la claridad y la facilidad de su uso, su mantenimiento se simplifica y es fácil de reutilizar. A menudo se genera un bajo acoplamiento.

Bajo Acoplamiento.

El acoplamiento es la medida de la fuerza con que una clase está conectada a otras clases, a las que conoce y recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras, lo que trae consigo que los cambios una clases ocasionen cambios en otras, sean más difíciles de entender y de reutilizar porque se requiere la presencia de otras clases de las que dependen. El diseño del módulo está regido por la creación de dependencias escasas, un ejemplo de esto lo constituye la definición de métodos del negocio poco dependientes de otros, lo que reduce el impacto del cambio y favorece la reutilización.

Creador

Este patrón como su nombre lo indica es el que crea, el guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando:

- B contiene a A.
- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor).

CAPÍTULO 3. DISEÑO DEL SISTEMA

- B usa a A.

Los patrones anteriormente expuestos son ampliamente utilizados en la concepción del módulo:

Teniendo en cuenta que el desarrollo está completamente basado en el paradigma orientado a objetos el patrón Experto se identifica perfectamente en cada una de las clases controladoras CL(clases listado) encargadas de manejar los listados de elementos , CC(clases controladoras) encargadas de la introducción de nuevos elementos, CM (clases modificar) encargadas de la edición de un elemento y CE(clases eliminar) encargadas de eliminar elementos, cada una de ellas relacionada a un concepto de los gestionados. Esta misma distribución hace claro el uso del patrón Alta Cohesión pues cada una de estas clases solo realiza un conjunto de tareas específicas sin suplantar las responsabilidades de otra.

Las relaciones entre clases están limitadas solamente a la dependencia de conceptos, por lo que cada clase solamente se relaciona con aquella cuyos conceptos dependen completamente de ella asegurando la mínima comunicación posible y potenciando un Bajo Acoplamiento (un CIAP contiene un Capítulo). Como se ha descrito existe dependencia entre conceptos por lo que algunas clases, claramente identificadas, están encargadas de la creación de objetos de otras que están en su dominio conceptual (un CIAP contiene un Subcomponente).

3.2 Clases del Diseño.




Se ha definido que la aplicación posea una estructura centrada en la flexibilidad ante los cambios y la reutilización. Mediante la conservación de dicha estructura durante la implementación, se obtendrá un sistema mantenible como un todo: será flexible a los cambios en los requerimientos e incluirá elementos que podrán ser reutilizados cuando se construyan sistemas similares.

3.2.1 Diagrama de Clases del Diseño.

Los diagramas de clases muestran el diseño del sistema desde un punto de vista estático, a través de una colección de elementos declarativos, como clases, colaboraciones y sus relaciones.

CAPÍTULO 3. DISEÑO DEL SISTEMA

Al ser el módulo una aplicación web, la misma se modelará con estereotipos definidos para este tipo de aplicaciones según la extensión de UML, lo que proporciona una mayor comprensión de las funcionalidades del sistema y permite distinguir qué atributos, operaciones y relaciones están activas en el servidor y cuáles están activas cuando el usuario está interactuando con la página en el navegador cliente. A continuación se brinda una explicación de cómo son usados estos estereotipos en el diseño de la propuesta del sistema, qué representa cada cual y las relaciones entre clases:

	<<Server Page>>: 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.
	<<Client Page>>: Es una página web que se ejecuta del lado del cliente sobre un navegador. Permite capturar los datos que serán persistidos en la base de datos, además de mostrar información útil al usuario.
	<<FormHTML>>: Es una colección de elementos de entrada que están contenidos en la página cliente y se comunican con las clases controladoras mediante submit.

Relaciones utilizadas entre clases:

<<usa>> definición: aprovechamiento, uso, empleo, manejo, utilidad, costumbre o tradición.

En el esquema se muestra en las relaciones entre las clases servidoras y los frameworks EJB e HIBERNATE con su respectiva relación con el JPA, pues las clases servidoras utilizan los mismos.

<<invoca>> definición: invocación, atracción o apelación.

En el esquema se muestra en las relaciones entre el Framework JSF y el integrador SEAM (Servlet), a su vez entre las clases servidoras y el manejador de entidades (Entity Manager) de JPA.

CAPÍTULO 3. DISEÑO DEL SISTEMA

<<envía>> definición: someter, presentar, someterse, entregarse, rendirse o entregar.

En el esquema se muestra en las relaciones entre los formularios y el Framework JSF, se refiere al envío de datos.

<<contiene>> definición: contiene.

En el esquema se muestra en las relaciones entre los Frameworks y sus subcomponentes, JSF y sus librerías para la presentación, HIBERNATE y JPA, SEAM y EJB.

<<accede>> definición: acceso, ataque o entrada.

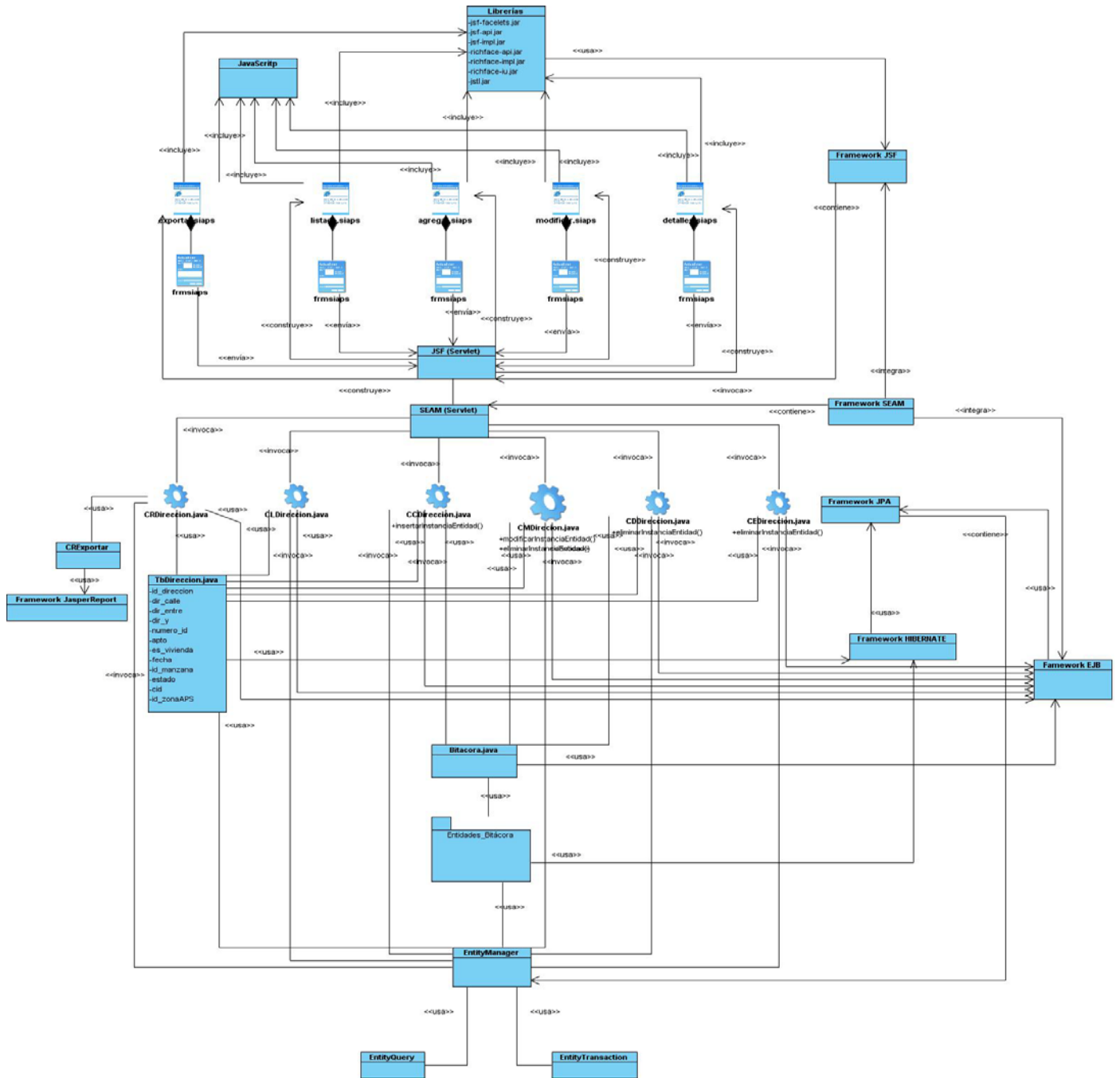
En el esquema se muestra en la relación entre el manejador de entidades Entity Manager y la Base de Datos.

3.3 Diagramas de Clases del Diseño.

Los diagramas de clases del diseño, son diagramas de estructura estática que se encargan de describir el comportamiento del sistema, mostrando las clases, los atributos y las relaciones entre ellos. Son utilizados durante el proceso de diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará, los componentes que se encargaran del funcionamiento y la relación entre ellos.

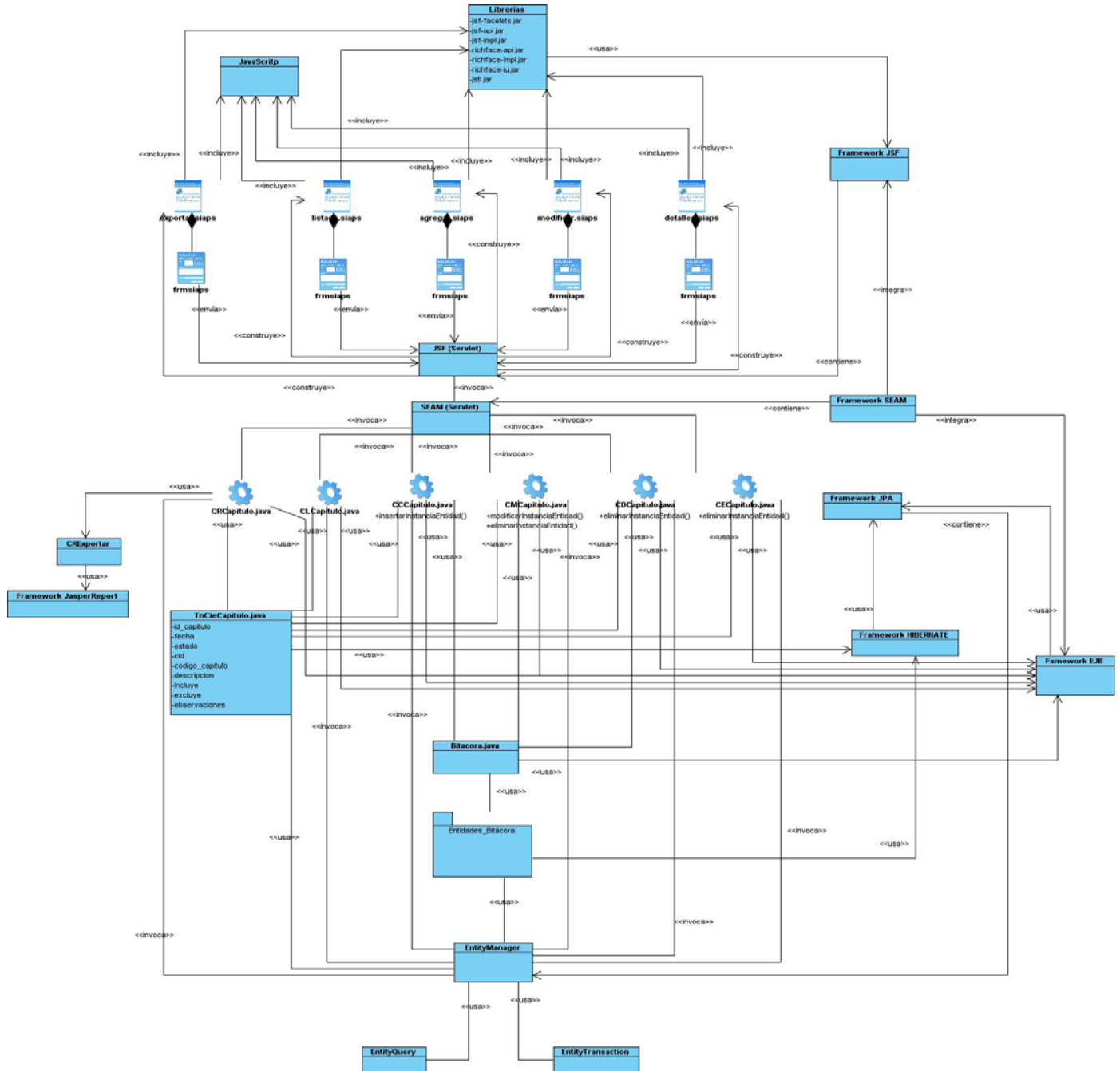
A continuación se muestran algunos diagramas de clases del diseño del sistema propuesto:

3.2.2.1. Gestionar Datos Dirección.



CAPÍTULO 3. DISEÑO DEL SISTEMA

3.2.2.2. Gestionar Datos Capítulo.



3.2.3 Descripción de las clases del diseño.

En el caso de los codificadores es similar la estructura general de la capa de presentación, por lo que se cuenta con una clase cliente para cada una de las acciones (agregar, modificar, listar, detalles y exportar). Igual sucede con las páginas servidoras que responden a la lógica del negocio.

Al describir las clases del diseño se tendrá una mayor comprensión del sistema a desarrollar. A continuación se describen algunas de ellas:

3.2.3.1 Gestionar Datos Dirección

Clase agregar.siaps

Propósito

Proveer la interacción con el usuario.

Descripción

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

Clase CRDireccion.java

Propósito

Proveer una respuesta a las peticiones realizadas en la vista.

Descripción

La clase CRDireccion.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 exportar solicitada sobre la página cliente correspondiente. Hace uno del

CAPÍTULO 3. DISEÑO DEL SISTEMA

Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Clase CCDireccion.java

Propósito

Proveer una respuesta a las peticiones realizadas en la vista.

Descripción

La clase CCDireccion.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 solicitada sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Clase TbDireccion.java

Propósito

Proveer el mapeo con la base de datos.

Descripción

La clase TbDireccion.java es una clase que se ejecuta del lado del servidor. Representa una tabla en el modelo de datos relacional que contiene los datos de las direcciones 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.

CAPÍTULO 3. DISEÑO DEL SISTEMA

3.2.4 Modelo de Datos.

El Modelo-Entidad-Relación (MER) también conocido como modelo de datos es un tipo de diagrama para modelado de bases de datos. Su objetivo es representar relaciones que existen en la vida real entendiendo su semántica. Los cuatro elementos fundamentales de un MER son: las entidades, atributos, interrelaciones y dominio, los diagramas se encuentran en el expediente de proyecto.

Conclusiones

En este capítulo se definieron los patrones a utilizar como son: GRASP y MVC, incorporando los beneficios desde el punto de vista arquitectónico y tecnológico que proveen. Con la obtención de los diagramas de clases del diseño, se obtuvo una representación gráfica de la estructura de las clases para el desarrollo del sistema, de fácil entendimiento para el equipo de desarrollo y como punto de partida para la fase de implementación.

Capítulo 4. Implementación

Este capítulo constituye la continuidad del Modelo de Diseño, implementando en términos de componentes las clases y subsistemas obtenidos en la disciplina anterior. Fundamenta la necesidad de integración del Módulo de Configuración con otros sistemas. Se presenta la descripción de algunos de los principales métodos y el Diagrama de Despliegue de la solución propuesta así como un estudio de los estándares de diseño, codificación y tratamiento de errores utilizados en la aplicación.

4.1 Relación con otros sistemas.

En la actualidad casi ninguna aplicación se concibe como un sistema aislado, y el Módulo de Configuración es un ejemplo primordial, pues es la razón de su existencia. Se contempla su integración al alas SIAPS como plataforma única para la gestión, procesamiento y transmisión de la información clínica en el Sistema Nacional de Salud. Este módulo brinda soporte a todos los que componen la primera iteración.

4.1.1 Módulo de Clínico Quirúrgico del SIAPS

Este módulo utiliza la información relacionada con las especialidades y los médicos, ubicados en una Entidad además de la información relacionada con los Nomencladores Asistenciales.

4.1.2 Módulo de Enfermería del SIAPS

Este módulo utiliza la información relacionada con los departamentos ubicados en una Entidad.

4.1.3 Módulo de Medios de Diagnostico del SIAPS

Este módulo utiliza la información relacionada con los departamentos y servicios asociados a una Entidad.

4.1.4 Módulo de Medicina Familiar del SIAPS

CAPÍTULO 4. IMPLEMENTACIÓN

Este módulo utiliza la información relacionada con los Nomencladores Asistenciales y las Áreas de Salud.

4.2 Implementación.

Para comenzar a desarrollar el flujo de trabajo de Implementación se comienza con el resultado del diseño y se define cómo se organizan las clases y objetos en términos de componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación. Uno de los principales propósitos de este flujo de trabajo consiste en desarrollar la arquitectura y el sistema como un todo.

4.2.1. Descripción de algunos métodos

Seguidamente se muestran algunas descripciones de algunos de los métodos del módulo, en la Capa de Negocio.

Método insertarAreadeSalud.

Implementa la acción de insertar en la BD la información referente a una nueva Área de Salud recibida como parámetros de entrada. Contiene un conjunto de validaciones que garantizan el buen formato de los datos así como la no existencia en la BD de la información a insertar. Maneja los errores que se pueden emitir en estas validaciones para cada caso. Utiliza la interface Entity Manager que permite el manejo de las entidades y contiene un grupo de funcionalidades que posibilitan la persistencia de los datos.

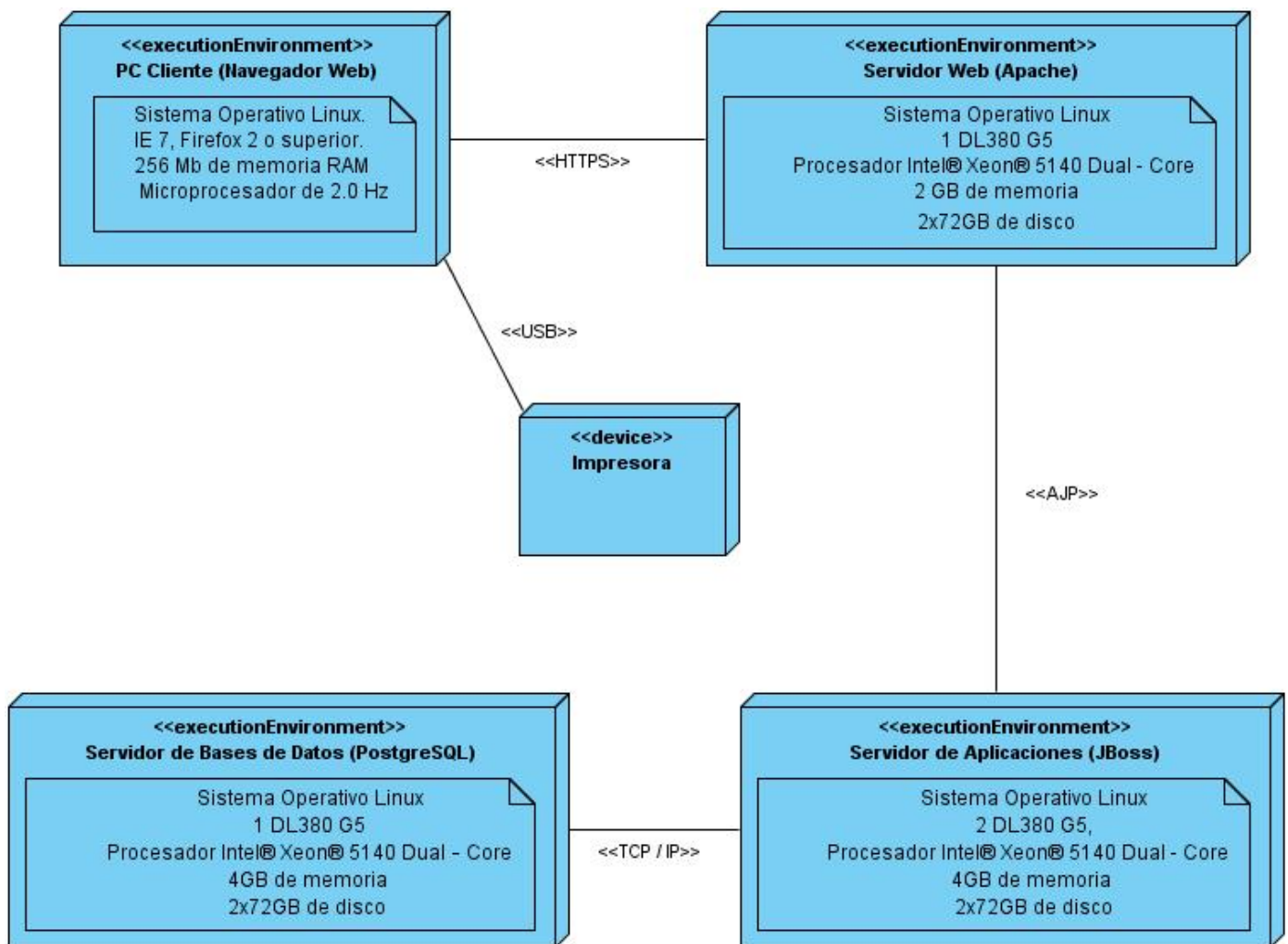
Método modificarCapitulo.

Implementa la acción de modificar en la BD la información referente a un Capítulo recibida como parámetros de entrada, estos se obtienen luego de la modificación realizada, sobre los datos cargados después de haber seleccionado el Capítulo a modificar en la página listar, esta envía el identificador del Capítulo y mediante una de las funciones del Entity Manager se extraen los datos modificables del elemento. Contiene un conjunto de validaciones que garantizan el buen formato de los datos y maneja los errores que se pueden presentar.

4.2.3. Diagrama de Despliegue.

El diagrama de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos. Los nodos representan recursos de cómputos: procesadores o dispositivos de hardware y entre ellos existen relaciones que representan medios de comunicación entre ellos.

La Vista de Despliegue del Módulo de Configuración quedaría de la siguiente manera:



4.3 Estándares de diseño, codificación y tratamiento de excepciones.

Con el propósito de que exista homogeneidad entre las aplicaciones que se encuentran integradas al Sistema Integral para la Atención Primaria de Salud alas SIAPS se han definido una serie de estándares tanto para el diseño como para la codificación, además cómo se le dará el tratamiento de errores al módulo para que los códigos fuentes de las aplicaciones y los mensajes que se emitan mantengan una uniformidad.

4.3.1 Estándares de diseño.

Estándares de codificación.

Las convenciones de código o estándares de codificación son importantes para los programadores por un gran número de razones:

- ✓ El 80% del coste del código de un programa va a su mantenimiento.
- ✓ Casi ningún software lo mantiene toda su vida el autor original.
- ✓ Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.
- ✓ Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.

Se definen una serie de pautas entre las que se encuentran:

Idioma.

- ✓ Se debe utilizar como idioma el español, las palabras no se acentuarán.

Comentarios, líneas y espacios en blanco.

CAPÍTULO 4. IMPLEMENTACIÓN

- ✓ Todos los ficheros fuente deben comenzar con un comentario en el que se lista el nombre de la clase, información de la versión, fecha, y copyright.
- ✓ Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas.
- ✓ Se deben usar siempre dos líneas en blanco en las siguientes circunstancias:
 - Entre las secciones de un fichero fuente.
 - Entre las definiciones de clases e interfaces.
- ✓ Se debe usar siempre una línea en blanco en las siguientes circunstancias:
 - Entre métodos
 - Entre las variables locales de un método y su primera sentencia
 - Antes de un comentario de bloque o de un comentario de una línea.
 - Entre las distintas secciones lógicas de un método para facilitar la lectura.
- ✓ Se deben usar espacios en blanco en las siguientes circunstancias:
- ✓ Una palabra clave del lenguaje seguida por un paréntesis debe separarse por un espacio.

Indentación y longitud de la línea.

- ✓ Se deben emplear cuatro espacios como unidad de indentación. La construcción exacta de la indentación (espacios en blanco contra tabuladores) no se especifica. Los tabuladores deben ser exactamente cada 8 espacios.
- ✓ Evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

Variables, constantes, clases y métodos.

- ✓ Todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres subguión "_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje.
- ✓ Los nombres de las variables deben ser cortos pero con significado. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador casual su función. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.
- ✓ Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("_"). (Las constantes ANSI se deben evitar, para facilitar su depuración.)
- ✓ Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas.
- ✓ Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.

4.3.2 Tratamiento de excepciones.

Durante el tiempo de ejecución de un sistema pueden fracasar diferentes rutinas, es a esto a lo que comúnmente se le llama excepción. Las excepciones son situaciones anómalas que requieren un tratamiento especial. No tienen por qué ser errores. Si se consigue dominar su programación, la calidad de las aplicaciones que se desarrollen aumentará considerablemente, con ellas se obtiene un sistema más robusto y fiable.

CAPÍTULO 4. IMPLEMENTACIÓN

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.

Una posible situación sería la introducción de caracteres extraños (¡"·\$%&/, etc.) estos no son válidos para ninguna de las operaciones de configuración, las excepciones son generadas desde el controlador mediante el uso de expresiones regulares y macheadas con un controlador de mensajes que identifica el tipo de excepción y muestra un mensaje con la información relacionada al usuario.

En la aplicación se manejan además las excepciones que se lanzan a partir de las operaciones sobre la base de datos, como la inserción de elementos con campos iguales que poseen la restricción de ser únicos, entre otros.

CAPÍTULO 4. IMPLEMENTACIÓN

Conclusiones

En este capítulo se mostró el diagrama de despliegue, también se presentaron ejemplos de los estándares de codificación y de diseño, además se trataron algunas de las posibles excepciones que el sistema pudiera presentar, como resultado se concluyó que:

- Los módulos desarrollados tendrán acceso a información crítica, estandarizada y configurada por el Módulo de Configuración.
- Los métodos, clases y pantallas se diseñaron se implementaron siguiendo las pautas establecidas por el proyecto, garantizado la legibilidad y estandarización del sistema.
- Los mensajes globales para el tratamiento de excepciones se corresponden con las pautas establecidas por el proyecto.
- El diagrama de despliegue cumple con los principios de la arquitectura cliente servidor definida.
- El Módulo Configuración del Subsistema Web para el Sistema Integral para la Atención Primaria de Salud alas SIAPS responde a las funcionalidades previstas para la primera versión.

Conclusiones

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

- ✓ El estudio del estado del arte determinó la reutilización de algunas de las funcionalidades del alas HIS, ahorrando tiempo de desarrollo de la aplicación y la necesidad de un nuevo módulo (Módulo Configuración del Subsistema Web para el Sistema Integral para la Atención Primaria de Salud alas SIAPS).
- ✓ La arquitectura propuesta demostró ser robusta y actualizada, permitiendo un desarrollo sólido de los componentes que conforman el módulo.
- ✓ La aplicación de los estándares y patrones internacionales analizados, así como las pautas establecidas por el proyecto, resultó en un esquema sencillo y robusto de fácil entendimiento por el equipo de desarrollo. Como resultado se generó un código legible y de rápida asimilación para futuros desarrolladores con dominio de las pautas.
- ✓ El Módulo Configuración del Subsistema Web para el Sistema Integral para la Atención Primaria de Salud alas SIAPS responde a las funcionalidades previstas para la primera versión, garantizando su correcto funcionamiento.

RECOMENDACIONES

Recomendaciones

Las recomendaciones de la investigación están dirigidas a sugerir acciones para complementar el producto obtenido. Por lo que para el buen desempeño y puesta en marcha de la aplicación se hacen las siguientes recomendaciones:

- Continuar con la investigación para el desarrollo del módulo teniendo en cuenta las nuevas aplicaciones que deben incorporarse al producto alas SIAPS.
- Conformar y aplicar un plan de adiestramiento para el personal involucrado en el trabajo con la aplicación.

GLOSARIO DE TÉRMINOS

Glosario de Términos

Aplicación (Sistema): Sistema que ofrece a un usuario final un conjunto coherente de casos de uso.

Aplicación Web: Es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de un navegador o browser. Estas son muy populares debido a la habilidad para actualizar y mantener la información manipulada sin distribuir e instalar el software en miles de potenciales clientes.

Arquitectura: Conjunto de decisiones significativas acerca de la organización de un sistema de software, la selección de los elementos estructurales a partir de los cuales se componen el sistema. La misma se interesa no solo por la estructura y el comportamiento, sino también por las restricciones y compromisos de uso, funcionalidad, funcionamiento, flexibilidad al cambio, reutilización, comprensión, economía y tecnología, así como por aspectos estéticos.

Arquitectura Java EE: Anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4, es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por The Java Community Process / JCP

Historia de Salud Familiar HSF: Es un documento oficial cuyo objetivo fundamental consiste en registrar la información relacionada con las características socioeconómicas, biológicas e higiénicas de los integrantes de la familia.

Historia de Salud Individual HCI: Documento oficial donde se encuentra reflejada la trayectoria sanitaria de un individuo desde su nacimiento hasta su muerte.

GLOSARIO DE TÉRMINOS

Nomencladores Asistenciales: Los nomencladores asistenciales están conformados por los 3 clasificadores internacionales referentes a las enfermedades, el Clasificador Internacional de Enfermedades CIE, el Clasificador Internacional de Enfermedades para la Atención Primaria de Salud CIAP y el Registro de Enfermedades de Declaración Obligatoria EDO.

Sistema Nacional de Salud: Un sistema de salud es la suma de todas las organizaciones, instituciones y recursos cuyo objetivo principal consiste en proteger y mejorar el estado de bienestar de la población, tiene que proporcionar servicios que respondan a las necesidades de la población y sean justos desde el punto de vista financiero. El correcto manejo de la información y el conocimiento, derivados de los datos procesados por los profesionales de la salud, es uno de sus mayores desafíos.

Tecnologías de la Informática y las Comunicaciones: Agrupan los elementos y las técnicas utilizadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, Internet y telecomunicaciones.

REFERENCIAS BIBLIOGRÁFICAS

Referencias Bibliográficas

[1] Ramírez Márquez, Abelardo Dr., Dr. Castell-Florit Serrate, Pastor, Dr. Mesa, Guillermo. 2003. INFOMED, Portal de Salud de Cuba. Escuela Nacional de Salud Pública (ENSAP), El Sistema Nacional de Salud de Cuba. [En línea] 2003. [Citado el: 20 de abril de 2008.] http://www.sld.cu/galerias/doc/sitios/infodir/09_el_sistema_nacional_de_salud.doc.

[2] Ídem a la referencia 1.

[3] Ídem a la referencia 1.

[4]CYMDIST (SOM)
<http://www.cyme.com/es/software/cymdistsom/>

[5] <http://www.sistemaisis.com/erp-modulos.htm>

[6]Software para Consultorios Médicos
<http://www.grandiyasociados.com/modulo.asp?Pais=ES&IDSoftware=8&m=49>

[7] Wikipedia, la enciclopedia libre. Internet. [En línea] [Citado el: 25 de Marzo de 2008.] http://es.wikipedia.org/wiki/Acceso_a_Internet.

[8] GNU Proyecto, Free Software Foundation. [En línea] [Citado el: 25 de Marzo de 2008.] <http://www.gnu.org/philosophy/free-sw.es.html>.

[9] Ídem a la referencia 8.

[10] Cabello Albino, Clifton. 2007. Sitio Web: El mundo Linux. Definición de Linux. [En línea] 2007. [Citado el: 20 de Marzo de 2008.] <http://www.elmundolinux.com/definicionlinux.php>.

[11] World Wide Web Consortium (W3C). Servicio Web. [En línea] [Citado el: 20 de Marzo de 2008.] <http://www.w3.org/>.

[12] Web Services - XML-RPC, SOAP, sobre PHP, Perl, y otros conceptos. [En línea] [Citado el: 30 de Abril de 2008.] <http://www.webnova.com.ar/articulo.php?recurso=426>.

[13] Ídem a la referencia 12.

[14] Ídem a la referencia 12.

[15] Arquitectura Cliente/Servidor. [En línea] [Citado el: 20 de Marzo de 2008.] <http://www.csi.map.es/csi/silice/Global71.html>.

REFERENCIAS BIBLIOGRÁFICAS

- [16] Aplicación en capas. [En línea] [Citado el: 2008 de Marzo de 2008.] <http://oness.sourceforge.net/proyecto/html/ch03s02.html>.
- [17] Ídem a la referencia 16.
- [18] Ídem a la referencia 16.
- [19] Casal Terrenos, Julio. Blogs: Julio Casal. Desarrollo de Software Basado en Componentes. [En línea] [Citado el: 15 de Marzo de 2008.] http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3985/default.aspx.
- [20] Ídem a la referencia 19.
- [21] Ídem a la referencia 19
- [22] Microsoft Corporation. Whitepaper: La arquitectura SOA de Microsoft® aplicada al mundo real. [En línea] Diciembre de 2006. [Citado el: 15 de Marzo de 2008.] www.microsoft.com/soa.
- [23] Ídem a la referencia 22
- [24] Martín M, Angela 2008. Sitio Web: ChanelPlanet. Software: Beneficios de la Arquitectura Orientada a Servicios. [En línea] 2008. [Citado el: 20 de Abril de 2008.] <http://www.desarrolloweb.com/articulos/392.php>.
- [25] <http://www.osalt.com/es/jboss>
- [26] http://www.postgresql-es.org/sobre_postgresql
- [27] Amarino, Mariano. Ajax un nuevo acercamiento a aplicaciones Web. 2005.
a. Disponible en: <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>
- [28] Ídem a la referencia 27
- [29] JBoss Community. (2007). Recuperado el 28 de Diciembre de 2008, de JBoss Ajax4jsf. Introducción. Disponible en:
a. <http://www.jboss.org/jbossajax4jsf/docs/devguide/en/html/Introduction.html>
- [30] <http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>
- [31] Suárez González, Héctor. Manual Hibernate. 2003. Disponible en:
a. <http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf>
- [32] http://eva.uci.cu/file.php/452/Bibliografia_Basica/introduccion_css.pdf

REFERENCIAS BIBLIOGRÁFICAS

[33] http://eva.uci.cu/file.php/452/Bibliografia_Basica/HTMLGuiaCompleta.pdf

[34] <http://www.sidar.org/recur/desdi/traduc/es/xhtmll/xhtmll11.htm#xhtml>

[35] <http://www.masadelante.com/faqs/xml>

[36] García de Jalón Javier, Rodríguez Iñigo Mingo José Ignacio, Alfonso Brazález Aitor Imaz, Larzabal Alberto, Calleja Jesús, García Jon. Aprende Java como si estuviera en primero. Escuela Superior de Ingenieros Industriales. Universidad de Navarra. España. Enero 2000.

[37] APSI_SW_DI_017_alasSIAP_Metodología_Enfocado_Procesos V1.0.doc

[38] Ídem a la referencia 37

[39] Ídem a la referencia 37

[40] Ídem a la referencia 37

[41] Ídem a la referencia 37

[42] Ídem a la referencia 37

[43] Ídem a la referencia 37

[44] Ídem a la referencia 37

[45] Ídem a la referencia 37

[46] Ídem a la referencia 37

Bibliografía

1. Agustín Froufe. Java 2. Manual de usuario y tutorial. Editorial Ra-Ma, 1999.
2. Ajax: Un nuevo acercamiento a las aplicaciones web Maestros del Web
<http://www.maestrosdelweb.com/editorial/ajax/>
3. Alin Software. Programación aplicada a la vida.
<http://www.alinsoft.com/neoendo.php>
4. Arquitectura Basada en Componentes
<http://www.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>
5. Bruce Eckel. Thinking in Java. 4th Edition. Prentice Hall. 2006.
6. CIAP-2 - Spanish
http://www.spapex.es/pdf/ciap_2_esp.pdf
7. Desarrollo de Software basado en Componentes
<http://www.lambdasi.com.ar/textocomp.asp?id=9>
8. Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Patrones de Diseño. Addison-Wesley, 2002.
9. Guía Breve de XHTML
<http://www.w3c.es/Divulgacion/guiasbreves/XHTML>
10. Introducción a Ajax4jsf. Juan Alonso Ramos
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>
11. Jacobson, Booch, Rumbaugh. El Lenguaje Unificado de Modelado. El Proceso Unificado de Desarrollo de Software. Editorial Addison Wesley, 1999
12. Luis Fernando Sánchez Maldonado. Business Process Management (BPM): articulando estrategia, procesos y tecnología.
<http://www.degerencia.com/articulos.php?artid=611>
13. M.C.Paulk, C.V.Weber, B.Curtis y M.B.Chrisiss. The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley.
14. Meyer. Prentice-Hall. Construcción de Software Orientado a Objetos. Bertrand 1998.
15. Patrón "Modelo-Vista-Controlador"

<http://www.proactiva-calidad.com/java/patrones/mvc.html>

16. Programación Java

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

17. Psico Activa

<http://www.psicoactiva.com/cie10/cie1.htm>

18. Rodríguez de la Fuente, Pérez, Carretero. Programación de Aplicaciones Web. Editorial Thomson, 2003.

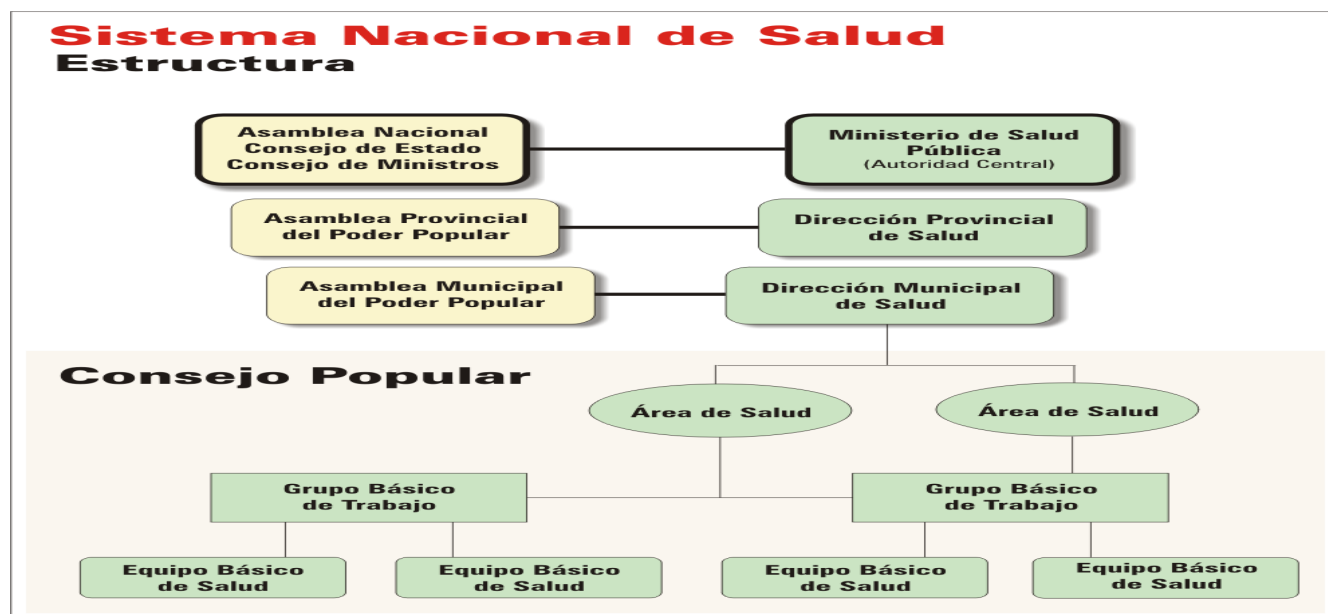
19. Roger S. Pressman. MacGraw-Hill. Ingeniería del Software. Un enfoque práctico. 2001. 5ª Edición.

20. Scott W. Ambler, Tyler Jewell. Mastering Enterprise Java Beans. Second Edition. Ed Roman. Wiley 2002.

21. Watts S. Humphrey. Addison-Wesley. Introducción al Proceso Software Personal (PSP). 2001.

Anexos

Anexo 1: Estructura del Sistema Nacional de Salud



Anexo 2: Arquitectura en tres capas.

