

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7



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

**Visor de Historia Clínica Individual del Sistema
Integral para la Atención Primaria**

Autores: Yaisel Botet Cesar

Osmín Pérez Morales

Tutores: Ing. Daichel Bárzaga Rosales

Ing. Yoiler Joaquín Frómeta Moreno

Ciudad de La Habana, julio de 2010

“Año 52 de la Revolución”

DATOS DE CONTACTO

Nombre: Ing. Yoiler Joaquín Frómata Moreno: Graduado de Ingeniería en Ciencias Informáticas en el año 2009, en la Universidad de las Ciencias Informáticas. Presenta la categoría docente de Instructor Recién Graduado, ha impartido la asignatura de Metodología de la Investigación 4to año. Pertenece al Dpto. de Atención Primaria de Salud del Centro de Informática Médica (CESIM). Se ha desempeñado como Jefe de Equipo del Módulo Medicina familiar.
yfrometa@uci.cu

Nombre: Ing. Daichel Bárzaga Rosales: Graduado de Ingeniería en Ciencias Informáticas en el año 2008, en la Universidad de las Ciencias Informáticas. Ha impartido la asignatura Sistema Operativo a tercer año y Seguridad Informática a cuarto año y se desempeña como jefe de asignatura de dichas asignaturas. Pertenece al Dpto. de Sistemas Digitales de la facultad 7 y se encuentra vinculado a la producción en el Dpto. de Atención Primaria de Salud del Centro Especializado en Soluciones de Informática Médica (CESIM). Se ha desempeñado como Jefe de Equipo de Desarrollo del Módulo Registro de Vacunación (anteriores desarrollo), configurando sobre software libre las herramientas y ambientes de trabajo necesarios para tecnología java, actualmente desempeña el Rol de programador en el módulo Historia de Salud familiar.
drosales@uci.cu

RESUMEN

En la Atención Primaria de la Salud se gestionan grandes volúmenes de información relacionada con los datos de los pacientes. Su procesamiento se realiza de forma manual. Los médicos y demás profesionales de la salud, necesitan mantener el contacto directo con dicha información para realizar los estudios, exámenes y labores médicas correspondientes. El objetivo de este trabajo de diploma es desarrollar el Visor de Historias Clínicas para el Sistema Integral de la Atención Primaria de Salud (alas SIAPS), para facilitar la visualización de los datos en esta área de Atención Primaria.

El desarrollo del sistema se basa en tecnologías libres, multiplataforma y sobre una arquitectura en capas, utilizando Java como lenguaje de programación e implementando el patrón de arquitectura Modelo-Vista-Controlador. Como Sistema de Gestión de Bases de Datos se utiliza PostgreSQL y como servidor de aplicaciones el JBoss Server. Son utilizadas las librerías de componentes web JBoss UI y RichFaces.

Entre los beneficios que aporta se encuentra proveer un control estricto de los datos de los pacientes, accediendo a la información clínica del paciente ágilmente, facilitando los procesos de diagnóstico, tratamiento y seguimiento; y podrá realizar investigaciones médicas mediante la revisión de los diagnósticos, entre otras tareas. El uso de esta aplicación permite la visualización de la información referente al paciente, gestionada por lo demás módulos que comprenden alas SIAPS.

Palabras Claves: Atención Primaria de Salud, Visor de Historias Clínicas, alas SIAPS

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Historia Clínica Individual.....	5
1.2 Contenidos de la Historia Clínica.....	5
1.3 Sistemas Informáticos para gestionar información referente a la Atención Primaria de la Salud. 13	
1.4 Análisis crítico de los Antecedentes.....	16
1.5 Tendencias y Tecnologías actuales a considerar y utilizar	16
1.6 Lenguaje de Programación.....	25
1.7 Servidor de Aplicaciones.....	26
1.8 Sistema Gestores de Bases de Datos.....	26
1.9 Metodologías de desarrollo de software.	27
1.10 Herramientas Utilizadas.....	31
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....	32
2.1 Problema y situación problemática.....	32
2.2 Modelo de Dominio	34
2.2.1 Conceptos Fundamentales.....	34
2.3 Propuesta del Sistema	37
2.4 Especificación de los requerimientos de software.....	38
CAPÍTULO 3. DISEÑO DEL SISTEMA.....	41
3.1. Descripción de la arquitectura, fundamentación.....	41
3.2. Patrones de arquitectura.....	41
3.3. Valoración crítica.....	42
3.4. Diseño.....	43
3.5. Modelo del diseño.....	43
3.6. Patrones de diseño.....	44

TABLA DE CONTENIDOS

3.7	Definición de elementos de diseño	45
3.8	Diagramas de clases del diseño	45
CAPÍTULO 4. IMPLEMENTACIÓN		54
4.1.	Modelo de despliegue.	54
4.2	Descripción de Nodos.....	55
4.3	Integración de con otros módulos	55
4.4	Estándares de codificación.....	56
4.5	Seguridad	59
4.6	Tratamiento de excepciones	61
4.7	Descripción de los métodos más importantes.....	62
CONCLUSIONES		64
RECOMENDACIONES		65
REFERENCIAS BIBLIOGRÁFICAS		66
BIBLIOGRAFÍA.....		69
GLOSARIO DE TÉRMINOS		72
ANEXOS		74

INTRODUCCIÓN

"Comprendimos perfectamente que vale, pero millones de veces más la vida de un solo ser humano, que todas las propiedades del hombre más rico de la tierra".

Ernesto Che Guevara

En la búsqueda de una respuesta, que fuera más allá de lo que la experiencia profana pudiera caracterizar como historia clínica (HC), se pudo definir la HC, como el conjunto de documentos surgidos de la relación entre el médico y el paciente y; a partir de la segunda mitad del siglo XX, como el registro de la relación que se establece entre los usuarios y el hospital o la atención primaria. [1]

Los primeros relatos con información clínica se deben a los "médicos" hipocráticos del siglo V (AC), quienes sintieron la necesidad intelectual de consignar por escrito, con precisión y orden, su experiencia profesional ante la enfermedad individual de sus pacientes; algo que, de hecho, les servía de herramienta fundamental, tanto para el tratamiento a otros pacientes con signos y síntomas similares como para enseñar a sus aprendices. [2]

La HC es el elemento principal para el ejercicio profesional del personal sanitario, ya que satisface las necesidades tanto del punto de vista asistencial, al ser el elemento recordatorio para el trabajo clínico con el paciente, como del investigador y docente. Además permite realizar el análisis retrospectivo del quehacer profesional de los trabajadores sanitarios, o sea; no es más que el documento donde se recoge la información de la práctica clínica relativa a un enfermo y los procesos a los que ha sido sometido.

Lo que antiguamente era un cuaderno de notas para el médico que registraba los datos del paciente según su propio criterio, al aparecer la especialización, el trabajo en equipo y la medicina hospitalaria, pasó a ser un registro de un grupo de profesionales; donde es considerada como el único documento válido desde los puntos de vista clínico y legal a todos los niveles de atención en salud. Con frecuencia, en el campo de la atención primaria, la HC se denomina como historia de salud. [3]

Las necesidades de intercambio de información, el crecimiento de los sistemas de información hospitalarios, y la proliferación de tecnologías implementadas sobre Internet, hicieron posible y necesario que los documentos clínicos fueran compartidos entre hospitales, organizaciones y prestadores de salud. Este hecho ha posibilitado la existencia y maduración de métodos y protocolos que establezcan de manera segura el intercambio de información médica.

En la actualidad la HC está llamada a ser uno de los elementos fundamentales en la medicina, lo cual se ha ido dificultando debido a las limitantes que presenta la Historia Clínica convencional o en papel.

En Cuba, con el surgimiento del Equipo Básico de Salud (EBS) conformado por el médico y la enfermera de la familia, aparece en el Sistema Nacional de Salud (SNS) la Historia Clínica Individual (HCI). En sus inicios la HCI se encontraba en manos del paciente lo cual traía consigo que las mismas no estuvieran disponibles en el momento de realizar estudios médicos, se perdía la planificación, control y seguimiento de los pacientes que no acudían periódicamente a los consultorios del médico de la familia; a ello también se le debe agregar la pérdida y deterioro de las mismas.

Con el objetivo de disminuir las deficiencias provocadas por la posesión por parte de los pacientes de las HCI la Dirección del SNS decide en el año 1996 recoger las HCI y mantenerlas en los correspondientes consultorios del médico de la familia; esta acción resolvió algunos de los inconvenientes existentes pero trajo consigo nuevos problemas entre los que destacó la escasez de espacios para almacenar el creciente volumen de información, de igual modo los errores de archivado han provocado demora a la hora de consultar alguno de estos documentos. Con la adopción de esta medida aumentó notablemente el número de pacientes que se personalizan en los hospitales sin la HCI, lo cual provoca que en dichas instituciones médicas no se tenga conocimiento inmediato de la historia de salud del paciente teniendo que esperar por el traslado de la HCI correspondiente al paciente por parte del médico de la familia, aumentándose con ello la posibilidad de deterioro de tan importante documento clínico.

La HCI presenta además una serie de limitantes las cuales han estado presente en la misma, encontrándose tanto en manos de los pacientes, como archivadas en los consultorios del médico de la familia como son:

El desorden y la falta de uniformidad de los documentos traen consigo que dentro de la HCI la información no se encuentre cuando se necesita lo que provoca una baja disponibilidad de la información contenida en esta y por ende que en muchas ocasiones se oriente la repetición de exámenes complementarios con todos los gastos económicos que los mismos demandan. La información ilegible la cual es causa indirecta de más de 7 000 muertes al año y de 1.5 millones de intoxicaciones por tomar el medicamento erróneo, según estudios del Instituto de la Ciencia Médica Estadounidense (IOM). El modelo como tal incurre en considerables gastos económicos tanto por su concepción en forma de libro, como el de tener un formato para niño y otro para adulto.

Dentro de la Historia Clínica la información se encuentra poco estructurada creando problemas a la hora de buscar datos específicos por parte de un profesional de la salud, al tener que revisar la HC completa, empleando tiempo de trabajo de los integrantes del EBS, el cual se pudiera utilizar en otras tareas.

En toda institución de atención de salud, el profesional que radica en esta debe poder consultar la información de un paciente, independientemente de en qué centro se encuentre almacenada, sea del mismo nivel de atención o un nivel adyacente. La información que necesita consultar el médico no es confiable, además de no estar estructurada de forma correcta, empañando su entendimiento y dificultando la toma de las decisiones. La información contenida no es inalterable por lo que las HCI han ido perdiendo el carácter científico ya que muchos médicos han perdido la confianza del contenido de la misma.

Luego de haber analizado la situación existente respecto al proceso de consulta de la HCI y por la importancia en el apoyo al mejoramiento del trabajo de nuestros profesionales de la salud se establece como **problema a resolver**: ¿Cómo facilitar el proceso de consulta de la información contenida en la HCI de la Atención Primaria de Salud (APS)?

Para el desarrollo de la investigación se plantea como **objeto de estudio**: la gestión de la información de la Historia Clínica dentro de la APS. El **campo de acción** se enfoca en: la visualización de la información contenida en la Historia Clínica Individual de la Atención Primaria de la Salud.

Para dar solución al problema antes mencionado se define como **objetivo general**: Desarrollar una aplicación Web que facilite el proceso de consulta de la información contenida en la HCI en la APS.

Se plantea como **idea a defender** que: con la implantación de un sistema que facilite el proceso de consulta de la HCI de la APS, se garantizará una mayor disponibilidad de la información médica de los pacientes, trayendo consigo un mayor dominio de los mismos y por ende mejor aprovechamiento.

Para dar cumplimiento al objetivo general se plantean las siguientes **Tareas de la Investigación**:

- Investigar y analizar los modelos actuales de la HCI.
- Revisar la bibliografía sobre las tendencias actuales en el mundo sobre la HCE
- Utilizar la arquitectura definida por el departamento de Atención Primaria de la Salud.
- Modelar los flujos de trabajo de Modelo del Negocio, Requerimientos, Análisis y diseño, e Implementación.

El presente documento está estructurado en cuatro capítulos como se presenta a continuación, estos contienen todo lo relacionado con el trabajo investigativo realizado.

Capítulo 1. Fundamentación Teórica: Se muestra una explicación de la fundamentación del tema. Se describen los conceptos fundamentales, asociados al dominio del problema, así como las tendencias de las tecnologías actuales a tener en cuenta.

Capítulo 2. Características del Sistema: Se describe la situación problemática existente y el flujo actual de los procesos a través de un Modelo de Dominio, el cual facilita el análisis necesario para determinar las funcionalidades del sistema a desarrollar. Estas se describen detalladamente mediante la especificación de requerimientos, descripción de casos de uso y representación gráfica.

Capítulo 3. Diseño del Sistema: Se describen los aspectos relacionados con el diseño de la solución propuesta, los patrones de diseño a utilizar en el desarrollo de la aplicación, se definen la estructura y los elementos del diseño, se presentan los diagramas de clases de los casos de uso del sistema utilizando estereotipo web. Se presenta el diagrama de clases persistentes y el modelo de datos. Conformándose, finalmente, el Modelo de Diseño, el cual constituye el punto de partida para la implementación del sistema.

Capítulo 4. Implementación: En este capítulo se describe la integración con otros componentes. Se muestra el modelo de implementación con el diagrama de despliegue. Se realiza una descripción detallada de los métodos más complejos en el desarrollo de la aplicación o agentes en el proceso de implementación, así como los estándares de diseño, codificación y tratamientos de errores utilizados.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

El objetivo de este capítulo es abordar los distintos aspectos que se utilizarán como soporte teórico del sistema a desarrollar. En el mismo se analizan los conceptos que se deben conocer para poseer un dominio básico de problema. En el mismo se realiza un análisis detallado del estado del arte en cuanto a sistemas, tecnologías, metodologías y herramientas.

1.1 Historia Clínica Individual

De la interrelación médico paciente se genera una serie de información médica, la cual se registra en diversos documentos clínicos, el conjunto de ellos constituyen la HCI. Desde el punto de vista clínico puede asegurarse que la HCI surge con el primer episodio de la enfermedad o el control de salud, la misma debe ser única y acumulativa de modo que en esta se encuentre registrada toda la información clínica de un paciente durante el periodo de vida del mismo.

La HCI es el documento básico de la atención médica primaria; tiene una función diagnóstica y sirve de base para el planeamiento, ejecución y control en cada caso, de las acciones destinadas al fomento, recuperación y rehabilitación de la salud. [4] Una pregunta interesante sería: ¿en qué medida el actual modelo de la HCI utilizado en la APS de Cuba se ajusta a estos propósitos?

Como se ha venido observando la HCI es utilizada para apoyar la asistencia médica; pero en la actualidad ante últimos avances de la medicina el actual modelo de la HCI se ha mostrado deficiente y carente de valor científico; lejos de apoyar la labor médica la ha entorpecido. [5]

1.2 Contenidos de la Historia Clínica

Al realizar un estudio histórico de las diferentes historias clínicas que a lo largo de la historia han existido se puede ver como el contenido de muchas de estas fueron realizando importantes aportes a los modelos de historia clínica que se conoce hoy en día.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.2.1 Historia Clínica Hipocrática

Estas historias clínicas datan del comienzo de la vida en que se pretendía interpretar las enfermedades. Los chamanes pretendían localizarlas para extraer y expulsar el cuerpo extraño causaba la enfermedad, lo cual implicaba, a la vez, una actitud terapéutica.

En estas historias clínicas lo fundamental era la descripción de los síntomas del paciente de acuerdo a lo que podían percibir. Se basaban en la descripción del problema y se intentaba relacionar con el medio ambiente en que se produjo la enfermedad sin ir más allá, como pudieran ser antecedentes de la enfermedad, ni se establecía una diferencia entre síntomas objetivos y subjetivos.

1.2.2 Los Consillia

Los consillia medievales era una historia clínica doctrinal de un puro estilo galénico, no aportaba al avance del conocimiento pero si contaba con un carácter docente.

Contaba de cuatro partes:

1. Título: En donde se establecía la entidad nosológica de la que se tratara.
2. Primera sección: contenía los síntomas observados por el médico ordenados conceptualmente y con algún que otro comentario doctrinal.
3. Segunda sección: contenía la discusión de las cuestiones relativas a la etiología, a la patogenia y al tratamiento.
4. Formula final: de tipo religioso. [6]

1.2.3 Historia Clínica Renacentista

En este tipo de historia clínica es que se comienza a hablar de órganos o forma normal y alterado. Contiene el primer protocolo de autopsia vinculado a la historia clínica. Esta historia clínica entre sus principales aportes tuvo:

- Recuperación de la ordenación cronológica de los síntomas al hacer las descripciones.
- Finalización de la redacción de la misma con la descripción de las lesiones encontradas en el cadáver al practicarle la autopsia. [7]

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.2.4 Historia Clínica Sydenhamiana

En el siglo XVII se produce la ruptura definitiva con la tradición galénica para dar paso a la revolución científica que marcó esta etapa. Aparece una nueva nosología que originó que la historia clínica comenzara a ser reconocida como el elemento básico de descripción de la enfermedad de un paciente.

Los aportes en esta historia clínica fueron:

- Empirismo clínico: solo describen lo que el observador es capaz de recabar por sus órganos de los sentidos. De esta forma el médico se aparta de tener que dar explicaciones y se aparta de la historia de la enfermedad.
- Especificidad: empieza a describir casos de enfermedad individual pero correlacionándolo con los casos típicos de una determinada especie morbosa.

1.2.5 Historia Clínica del siglo XVIII

Basándose en la Historia Clínica Sydenhamiana, el médico neerlandés Hermann Boerhaave [8] realizó un modelo de historia clínica que es considerado canónico y que con algunas variaciones que se le fueron incorporando con los siglos es el que se utiliza hoy.

1.2.5.1 Historia Clínica Boerhaaviana

Se establece que el examen de los enfermos debía constar de tres tiempos; inspección, anamnesis y exploración.

Con la inspección el médico conoce el sexo del paciente, el biotipo, el estado de la vida del paciente, y además: los hábitos, costumbres, posición social, etc.

Con la anamnesis el sanitario intenta conocer todo lo relacionado a los antecedentes, tanto los familiares como los personales, el comienzo de la enfermedad, el curso de la misma hasta el momento en que asiste al médico.

Con la exploración objetiva se pretende buscar el estado morfológico funcional de las distintas partes del organismo del paciente.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.2.6 Historia Clínica del siglo XIX

Para el estudio de la historia clínica en este periodo es necesario tener en cuenta las tres mentalidades médicas: mentalidad anátomo – clínica, la mentalidad fisiopatológica y la mentalidad etiológica.

1.2.6.1 Historia Clínica anátomo-clínica

Se comienza a hablar de las enfermedades, no por los síntomas sino por las lesiones que encuentran y así se comienza a clasificar las enfermedades por las lesiones.

1.2.6.2 Historias Clínicas fisiopatológicas

Teniendo en cuenta que para esta mentalidad la enfermedad era una alteración de las funciones del organismo provocados por procesos materiales o por procesos energéticos dan lugar a la aparición en la historia clínica de una serie de signos fisiológicos que indicaban el trastorno funcional, por ejemplo la temperatura, y los resultados de nuevas exploraciones, además de la sucesión en el tiempo del curso de las enfermedades.

1.2.6.3 Historia Clínica Etiológica

Para este modo de pensar las causas de la enfermedad eran lo más importante llegando a concebir a esta como el resultado de la lucha del organismo humano por vencer al agente causal. En la historia clínica se remarca la relación entre los antecedentes, tanto familiares como personales, con el estado actual de la enfermedad. Se incluyen nuevas exploraciones: los cultivos de gérmenes, la identificación de las sustancias venenosas, etc. [9]

1.2.7 Historia clínica del siglo XX

Durante este tiempo se comienza a comprender que hay que estudiar al hombre sano o enfermo como un todo. Naciendo así dos nuevas mentalidades que repercutirán en la historia clínica: las mentalidades biopatológicas y la antropatológica.

En cuanto a la Historia Clínica:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1. **Historia Biopatológica:** se desarrolla en función de la constitución del ser humano, uno de los aportes de este modo de pensar a la historia clínica es la descripción del hombre la cual adquiere un rango especial. Se incluyen pruebas funcionales que estudian la adecuación del ser vivo a su nuevo estado.
2. **Historia Antropatológica:** Como la constitución del hombre no resuelve el problema de la personalidad, la historia clínica se convierte en la biografía de la enfermedad de un paciente.

Al transcurrir los años la historia clínica se ha ido enriqueciendo y ganando en composición, llegando a que las mismas se desarrollen entorno a cinco apartados básicos:

- Descripción del Individuo.
- Antecedentes remotos y próximos
- Enfermedad actual
- Curso de la enfermedad
- Inspección del cadáver

Todas estas historias clínicas vistas hasta el momento eran elaboradas por los médicos sin existir un debido control de la elaboración de las mismas, por otra parte no cumplían con los mínimos de calidad exigidos por la Organización Mundial de la Salud (OMS) para este tipo de modelo. Ello trajo consigo que se diera un movimiento en pro de crear un modelo de HCI el cual fuera estándar y facilitara las tareas de elaboración de estas así como las consultas y que además tributase a la docencia e investigación. [10]

Producto de estos nuevos movimientos se crean nuevos modelos de Historia Clínica:

1.2.8 Historia Clínica Cronológica

Este tipo de historia clínica se basa en distribuir los datos en una sucesión correlativa a lo largo del tiempo. Este modelo de historia clínica es muy útil para pacientes con escasos problemas sanitarios o que acuden excepcionalmente a las consultas; pero no suele ser para nada útil a mediano y largo plazo.

Dentro de estas historias clínicas podemos encontrar dos modelos:

- La historia clínica cronológica por secciones (**ver anexo 1**): Cada departamento o servicio asistencial tiene su sección en la cual registran los datos de forma cronológica. Este modelo facilita

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

la recuperación de la información por parte del médico de aquello que le interese, pero hace bastante difícil conocer el estado del paciente pues habría que revisar cada una de las secciones de forma íntegra.

- La historia clínica cronológica integrada: Es la historia clínica cronológica de forma pura, en la que cada uno de los datos se van registrando sucesivamente, con este modelo habitualmente es necesario revisar toda la historia clínica para encontrar una determinada información o conocer el estado del paciente.

1.2.9 Historia clínica centrada en el paciente

Este tipo de historia clínica define o conceptualiza que debe existir un registro por paciente, donde debe aparecer todos los episodios de salud del individuo. Se le suma el objetivo de que un paciente debe tener solo un registro al menos en el mismo nivel de atención.

1.2.10 Historia Clínica Orientada al Nivel de Atención

Existen diferentes modelos de registro médico con características propias del nivel de atención en el cual brindan soporte. De esta forma, una historia clínica de internación es diferente en cuanto a su estructura con respecto a la de la atención ambulatoria o de una central de emergencias. [11]

1.2.11 Historia clínica orientada a las especialidades

En el nivel secundario de atención, las especialidades médicas organizaron la información del registro según el dominio que les correspondía con un ingreso estructurado. [12]

1.2.12 Historia Clínica Orientada a las patologías

Una de las características del avance de la medicina en el campo de la investigación clínica se basa actualmente en estudios sobre temas puntuales, motivo por el cual la orientación del registro a patologías permitió los estudios prospectivos y retrospectivos de las mismas. En este tipo de modelo se almacena la información con un nivel de especificidad adecuado para la patología en cuestión. [13]

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.2.13 Historia Clínica Orientada Hacia las Decisiones

Este modelo de historia clínica tal y como el nombre lo indica tiene que ver con la toma de decisiones. Por ello presenta una orientación hacia el diagnóstico probable de enfermedad y las demás dolencias que hay que excluir en cada caso.

1.2.14 Historia Clínica Orientada a Problemas (HCOP)

Para lograr el entendimiento de este modelo de historia clínica se hace necesario conocer la definición de problema:

“...todo aquello que requiera diagnóstico, manejo posterior o interfiera con la calidad de vida, de acuerdo con la percepción del paciente” [14]

De acuerdo a la definición de la HCOP, esta está constituida por cuatro elementos:

1. **La Base de Datos (ver Anexo 2):** Es la fotografía del estado del paciente. Contiene la información de los problemas vigentes, los antecedentes personales y familiares y el desarrollo de los principales problemas.
2. **La lista de problemas:** Funciona como un índice de contenidos de la HCOP. Suele ubicarse en la primera página y constituye un resumen de todas las necesidades de salud de la persona.

La lista de problemas se encuentra formada por dos listas:

- **Lista Maestra o lista de problemas crónicos:** en esta lista se registran los problemas más firmes, sean estos crónicos o inactivos.
 - **Lista de Problemas Agudos:** Se asemeja a una lista preliminar o lista de trabajo, donde se registran problemas volátiles o de corto periodo de actividad. En ella, los problemas suelen rotularse con una letra y se anotan todas sus fechas de aparición en la consulta, para verificar su evolución.
3. **Plan Inicial:** Para cada problema priorizado en la primera consulta es necesario establecer un plan inicial de manejo. Esta sección despliega una planificación sistematizada del abordaje de cada problema. Para ello se emplea una estructura que ayuda a evitar omisiones y a transparentar la lógica de las acciones del prestador de salud. Sus componentes son:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- **Plan Diagnóstico:** Este plan contempla toda la información a ser recogida en un posterior interrogatorio, exámenes de laboratorio o complementarios, imágenes de diferentes tipos, etc.
 - **Plan Terapéutico:** Se registran las indicaciones orientadas para la solución del problema. Incluye el tratamiento farmacológico, como las recomendaciones dietéticas y de cambios de estilos de vida, terapias recomendadas y otras recomendaciones relevantes.
 - **Plan de seguimiento o Monitoreo:** Contiene diversos indicadores, los cuales deben ser medidos regularmente para controlar la evolución del problema. Incluye elementos propios de la prestación, tales como exámenes auxiliares o el peso, así como los del auto monitoreo y auto cuidado. Lo importante es que sean validos para mostrar la tendencia del problema.
 - **Plan Educativo:** Breve descripción de la información que se le debe dar al paciente sobre su enfermedad.
- 4. Notas de la Evolución:** Solo se evolucionan los problemas priorizados de cada consulta. Estas se dividen en cuatro secciones formando el acrónimo SOAP:
- 1 Subjetivo: Información brindada por el paciente e impresiones del médico o paciente sobre la evolución de los síntomas
 - 2 Objetivo: Es la información objetiva. Recoge los signos del paciente, datos del examen físico y exámenes auxiliares.
 - 3 Apreciación: Interpretación y reevaluación del problema en función de las secciones previas.
 - 4 Plan: Se especifican cambios en los planes iniciales los cuales se van reajustando de acuerdo a los hallazgos y resultados obtenidos hasta el momento.

La Historia Clínica Orientada al Problema es un medio empleado en los sistemas de Atención Primaria de diversas partes del mundo, y se extiende cada vez más como un estándar de trabajo en la informatización de registros médicos. Este éxito indudablemente se origina en sus múltiples bondades como repositorio del conocimiento acumulado sobre el paciente, como medio de comunicación entre los prestadores de

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

salud, como registro estructurado que contribuye a la docencia o investigación, y como medio que facilita la auditoría de acciones prestacionales. [15]

El llevar las anotaciones de acuerdo a la estructura propuesta en la HCOP, es muy útil para obtener notas de atenciones claras, breves y bien organizadas, superando las deficiencias comunes presentadas por las historias convencionales. Estas ventajas, convierten a la HCOP en una potente herramienta universal para las acciones realizadas por el personal de salud de la APS.

1.3 Sistemas Informáticos para gestionar información referente a la Atención Primaria de la Salud.

El mundo está encaminado a la mejora del proceso de gestión de la información de la Historia Clínica, con acciones como la creación de Historias Clínicas Únicas, donde se registre la información del paciente y los actos del personal de salud en un sistema de información. Así como la Interoperabilidad: que es la 'comunicación' entre diferentes tecnologías y diferentes aplicaciones de software, que permite el 'intercambio' de datos en forma precisa, efectiva y consistente, y que permite el 'uso' de la información intercambiada. [16]

Los principales sistemas del mundo están optando por el uso de la codificación de la información médica; lo cual no solo contribuye a la seguridad sino que permite que se puedan relacionar datos que escritos como texto suelto no se le hallen concordancias.

La firma digital es otra de las soluciones que actualmente se pone en práctica, la cual no es más que: un método criptográfico que en la transmisión de un mensaje es quien asocia la identidad de una persona o de un equipo informático al mensaje o documento. Cumple con requisitos como: Identidad, integridad, no repudio y como último aspecto, la confidencialidad.

En la mayoría de los sistemas actuales existentes en el planeta, se utiliza el conjunto de estándares para el intercambio electrónico de información médica **Health Level Seven (HL7)**. La organización mundial que conlleva dicho nombre, cuenta con varios estándares que facilitan el envío de información entre aplicaciones con el objetivo de definir una serie de herramientas de interoperabilidad.

Algunos de los estándares son:

- Mensajería HL7 Versión 2: Estándar de mensajería para el intercambio electrónico de datos de salud.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Mensajería HL7 Versión 3: Estándar de mensajería para el intercambio electrónico de datos de salud basada en el RIM (Reference Information Model).
- CDA HL7: (Clinical Document Architecture) Estándar de arquitectura de documentos clínicos electrónicos.
- SPL HL7: (Structured Product Labeling) Estándar electrónico de etiquetado de medicamentos.
- HL7 Medical Records: Estándar de administración de Registros Médicos.
- GELLO: Estándar para la expresión de reglas de soporte de decisiones clínicas.
- Arden Syntax: Es estándar sintáctico (if then) para compartir reglas de conocimiento clínico.

En varias zonas del mundo como en Zaragoza, España; encaminan los esfuerzos a la conformación de tecnologías y sistemas, donde el sistema sería el encargado de enviarle al facultativo –sin necesidad de pedirlo- la información, prueba o dato relevante de algún paciente. Además de que con un simple clic, el médico dispondrá de todos los datos para hacer el mejor diagnóstico posible. [17]

Actualmente existen sistemas en el mundo que no resuelven totalmente el problema planteado.

Antecedentes Internacionales:

ITALICA:

Es una Historia Clínica Electrónica con el objetivo de integrar los sistemas informáticos preexistentes en el Hospital Italiano de Buenos Aires, está centrada en el paciente y orientada a objetos. Incluye datos demográficos del paciente, listado de alergias, últimas evoluciones (notas clínicas), listado de medicación que consume el paciente, niveles de atención a los que consultó últimamente, fecha de últimos exámenes complementario y listado de problemas activos. Muestra además de los antecedentes del paciente, los antecedentes familiares. En este existe un cuadro de información relevante donde aparecen los programas de seguimiento de enfermedades crónicas en los que se encuentra enrolado el paciente. El sistema posee un componente de servicios efectores donde mediante la mensajería electrónica basada en el estándar HL7 se envían los exámenes complementarios.

Cuenta con un servicio de interconsultas donde un médico solicita la intervención de un profesional de otro servicio como interconsultor. Este sistema posee una interfaz para que el médico elija el servicio y explique el motivo de el pedido, donde al confirmarse esta operación se envía un correo electrónico al grupo médico correspondiente, notificando el pedido, con los datos filiatorios del paciente, su ubicación en el hospital y datos del resumen de historia clínica. [18] De este módulo se puede heredar la operabilidad ya que es muy similar a los procesos de consulta que se desarrollan dentro de la atención primaria.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

La Historia Clínica Electrónica "ITÁLICA" se desarrolló con arquitectura cliente-servidor, y se programó en Power Builder que es uno de los lenguajes desarrollados para el marco de trabajo .NET, por lo que es totalmente imposible el trabajo con este sistema ya que está desarrollado con tecnología propietaria.

Visor de Historia Clínica Electrónica del Servicio Cántabro de Salud

Este visor permite acceder a los datos de un paciente, que se encuentren almacenados en los centros de salud u hospitales. El Visor se nutre de la información depositada en las bases de datos de los Centros de Atención primaria y del hospital. El usuario tendrá acceso a la información a través de la Intranet del Servicio Cántabro de Salud utilizando canales seguros.

Para ello un motor de integración corporativo manda toda la información a Oracle Healthcare Transaction Base (Oracle HTB), que actúa como repositorio central que almacena y gestiona información clínica en formato HL7 versión 3. La integración se va a realizar con el motor de integración JCAPS v 5.1.1 de Sun Microsystems utilizando mensajería HL7 v3 que se está imponiendo como la mensajería sanitaria estándar a nivel internacional. Las conexiones a las bases de datos orígenes se hace con JDBC. Se genera así un repositorio constante de datos normalizados, lo que permite la creación de un historial electrónico único para cada paciente. Los informes integrados en el HTB se realizan con mensajería CDA (Clinical Document Architecture) de HL7. [19]

El proyecto tuvo una inversión de 450 000 euros, se inició con un piloto de desarrollo que sirvió para detectar las limitaciones y necesidades que se plantearon a la hora de ampliar el proyecto.

El sistema no terminado plantea incorporar toda la gestión de la información asociada a los servicios de atención primaria prestados en las instituciones como: vacunaciones, seguimiento a los pacientes crónicos, seguimiento a embarazadas, educación sanitaria.

Para el trabajo con este sistema se requiere del pago de una licencia de 800 dólares por usuario o 40000 dólares por procesador.

Antecedentes Nacionales:

Era de esperar que con el alto nivel en Ciencias Médicas existente en Cuba desde los primeros años de la revolución, se realizaran intentos por informatizar los registros médicos, pero en la mayoría de los casos, están basados en resolver problemas particulares o específicos.

La empresa cubana de software para la salud SOFTEL, ha desarrollado un sistema para el control de las Historias Clínicas denominado: Clínico, la cual ya brinda su tercera versión con la incorporación de nuevos

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

módulos. Este incluye un registro del paciente que satisface los requerimientos del sistema, pero es de poca utilidad para los profesionales, ya que no supe todas las necesidades informativas que necesita el médico, además de sustituir mecánicamente la HC basada en papeles, cuando el objetivo debe ser la creación de un registro electrónico donde toda la información del paciente se encuentre en forma codificada y no sea necesaria su eliminación periódica. [20]

De manera nacional existen factores que dificultan la informatización de la Historia Clínica como los problema de Soporte técnico: Necesidad de redes, necesidad de equipamiento (servidores, escáner, etc.), necesidad de equipamiento especial para entrada directa de los datos desde lugares fuera de la institución de salud.

1.4 Análisis crítico de los Antecedentes

Después de haber analizado las características de diferentes soluciones y tecnologías existentes en el mundo, se ha podido determinar que en su gran mayoría, son sistemas que presentan funcionalidades de relevancia para la visualización de la información contenida en la Historia Clínica Individual. Los sistemas internacionales mencionados utilizan tecnología propietaria, o sea que la licencia para adquirir estas tecnologías es privativa, por lo que sería necesario invertir una gran suma de dinero para hacerse de estos recursos.

1.5 Tendencias y Tecnologías actuales a considerar y utilizar

1.5.1 Internet: TCP/IP

En el mundo con el advenimiento de la Internet se logran conectar ordenadores por todo el planeta a esta inmensa red que tiene un ámbito mundial a través de las distintas líneas de intercambio de datos (telefónicas, fibra óptica, satélite, televisión, ondas radiales, telefonía móvil, red eléctrica, ordenadores portátiles).

En la red global se encuentran conectados los ordenadores de distintos sistema operativo, hardware y configuración, todo posible al protocolo TCP/IP que es el que se encarga de que la comunicación entre todos sea posible.

Cuba planea la incorporación de estas nuevas tendencias dentro de unos años, cambios con los que se podrá alcanzar una gran variedad de novedosos servicios.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.5.2 Sistemas distribuidos. Modelo Cliente servidor.

Un sistema distribuido es una colección de computadoras independientes; es decir autónomas, que aparecen ante los usuarios del sistema como una única computadora, las cuales se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo.

Los sistemas distribuidos presentan importantes características entre las que destacan:

- **Concurrencia.**- Esta característica de los sistemas distribuidos permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que interactúan en la red.
- **Carencia de reloj global.**- Las coordinaciones para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no tienen una temporización general, está más bien distribuida a los componentes.
- **Fallos independientes de los componentes.**- Cada componente del sistema puede fallar independientemente, con lo cual los demás pueden continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continúa trabajando.

Los sistemas distribuidos han ido evolucionando pasando por:

Procesamiento central(Host): Uno de los primeros modelos de ordenadores interconectados, llamados centralizados, donde todo el procesamiento de la organización se llevaba a cabo en una sola computadora, normalmente un Mainframe, y los usuarios empleaban sencillos ordenadores personales.

Grupo de Servidores.- Otro modelo que entró a competir con el anterior, también un tanto centralizado, son un grupo de ordenadores actuando como servidores, normalmente de archivos o de impresión, poco inteligentes para un número de Minicomputadores que hacen el procesamiento conectados a una red de área local.

La Computación Cliente Servidor.- Este modelo, que predomina en la actualidad, permite descentralizar el procesamiento y recursos, sobre todo, de cada uno de los servicios y de la visualización de la Interfaz

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Gráfica de Usuario. Esto hace que ciertos servidores estén dedicados solo a una aplicación determinada y por lo tanto ejecutarla en forma eficiente.

En la arquitectura cliente servidor la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores, estos últimos responden a la demanda del cliente que la produjo.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet.

¿Qué es una arquitectura?

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización.

Debemos señalar que para seleccionar el modelo de una arquitectura, hay que partir del contexto tecnológico y organizativo del momento y, que la arquitectura Cliente/Servidor requiere una determinada especialización de cada uno de los diferentes componentes que la integran.

¿Qué es un cliente?

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

¿Qué es un servidor?

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.5.3 Patrones de arquitectura.

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. Estos capturan la experiencia existente y probada para promover buenas prácticas.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

El Modelo Vista Controlador (MVC): Es un patrón de arquitectura de software el cual es muy utilizado para el diseño de aplicaciones con sofisticadas interfaces. Describe una forma, muy utilizada en la Web, de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Este modelo de arquitectura reporta ventajas como:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Soporta múltiples vistas: Como la vista se encuentra separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.
- Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación y como el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

El diseñar un sistema con este patrón da la posibilidad de construir por separado las piezas o partes del programa y luego unir las en tiempo de ejecución. En caso de que alguna parte se vea afectada, conllevando a que el funcionamiento de un componente no sea el óptimo este puede ser sustituido por otro.

1.5.4 Frameworks, librerías y componentes.

Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Son diseñados con el objetivo de facilitar el desarrollo de software, permitiéndole a los desarrolladores y diseñadores dedicar más tiempo a identificar requerimientos del software ya que disminuyen el tener que tratar con los detalles de bajo nivel que provee un sistema funcional.

Java Server Faces (JSF)

Es un framework de desarrollo basado en el MVC; pretende normalizar y estandarizar el desarrollo de aplicaciones web. Trata a la vista (la interfaz de usuario) de una forma algo diferente a lo que acostumbrado en aplicaciones web, donde la programación del interfaz se hace a través de componentes y basada en eventos, permite la asociación entre los datos de la interfaz y los datos de la aplicación web. JSF es muy flexible; permitiendo crear componentes propios, o crear propios “render” para pintar los componentes según convenga.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Seam UI

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

RichFaces

RichFaces es una librería de componentes visuales para JSF, la cual es de código abierto. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.

Dentro de las características de RichFaces se puede encontrar que:

- se integra perfectamente en el ciclo de vida de JSF,
- incluye funcionalidades Ajax, de modo que nunca se ve el JavaScript y tiene un contenedor Ajax propio,
- contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un número bastante amplio que cubren casi todas nuestras necesidades,
- soporta facelets,
- soporta css themes o skins,
- es un proyecto open source, activo y con una comunidad también activa.

Ajax

Es el acrónimo de **A**synchronous **J**avaScript **A**nd **X**ML (JavaScript asíncrono y XML), es una técnica para el desarrollo web con la cual se pueden crear aplicaciones interactivas. Aplicaciones que se ejecutan en el cliente mientras se mantiene la comunicación asíncrona con el servidor en segundo plano; permitiéndolo realizar cambios sobre las páginas sin la necesidad de recargarlas, lo cual trae consigo un aumento en interactividad, velocidad y usabilidad en las aplicaciones.

Ajax incorpora:

- 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. [21]

Ajax4JSF

Ajax4jsf es una librería open source que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, etc. En definitiva Ajax4jsf permite dotar a nuestra aplicación JSF de contenido mucho más profesional con muy poco esfuerzo. [22]

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.

Dentro de las propiedades más interesantes de facelets se encuentra:

- Trabajo basado en plantillas.
- Fácil composición de componentes.
- Creación de etiquetas lógicas a la medida.
- Funciones para expresiones.
- Desarrollo amigable para el diseñador gráfico.
- Creación de librerías de componentes.

JBoss Seam

JBoss Seam es un poderoso y moderno framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB). Una característica importante es que se pueden hacer validaciones en los POJOs (Plain Object Java). Mientras que en los frameworks tradicionales, todo el estado es

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

administrado básicamente en la sesión HTTP, Seam provee una mayor granularidad de contextos de estado. [23]

Se introduce el concepto de contextos. Cada componente existe dentro de un contexto.

El principal contexto, es el 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* del navegador procesos 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*). [24]

Hibernate

Hibernate es un framework de software libre que provee herramientas de mapeo objeto – relacional permitiendo reducir significativamente el tiempo de desarrollo. El mismo genera sentencias sql a partir de sentencias HQL lo cual le ofrece a nuestras aplicaciones poseer portabilidad entre los motores de bases de datos.

Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible. También ofrece un API para construir las consultas programáticamente (conocido como “*criteria*”).

Deben destacarse ciertas características de las colecciones en Hibernate: [25]

- 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.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

FreeMarker

Es una biblioteca de clases para el lenguaje de programación Java. Su uso es factible en aplicaciones basadas en el modelo vista controlador. Es un motor para plantillas; FreeMarker genera páginas de texto que muestra los datos utilizando plantillas.

Hermes-HL7-CDA-v2

Hermes es una biblioteca de clases desarrollada para la creación de mensajes HL7 v2.3.1, así como su validación. También soporta la creación de documentos clínicos conforme al estándar CDA. Para los mensajes se utilizan las reglas de codificación XML. Las herramientas sobre la cuales se apoya este desarrollo permiten obtener plataformas en un tiempo corto y tiene un potencial de réplica en otras plataformas diferentes de Java. [26]

1.5.5 Tecnologías

CCS: hojas de estilo en cascada (Cascading Style Sheets), representan un avance importante para los diseñadores de páginas web, al darles un mayor rango de posibilidades para mejorar la apariencia de las páginas. Las CCS constituyen un mecanismo simple, que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura.

Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML, separando el contenido de la presentación. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

XHTML: (Lenguaje Extensible de Marcado de Hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web ante sus limitaciones de uso con las cada vez más abundantes herramientas basadas en XML. Es una reformulación de los tres tipos de documento definidos por HTML 4, pero como aplicaciones de XML

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

XML: No es más que un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

No ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

HL7 v3: Estándar de mensajería para el intercambio electrónico de salud basada en el Modelo de Referencia de Información (RIM).

CDA: Documento estándar definido por HL7 que especifica la estructura y la semántica de un documento clínico con el propósito de lograr un correcto intercambio de la información. Objeto de información que puede incluir texto, imágenes, sonido y otros elementos multimedia. Documento XML derivado del modelo de referencia de información (RIM) de HL7.

1.6 Lenguaje de Programación

El lenguaje de programación escogido es Java. Lenguaje de programación orientado a objetos y posee mucha similitud con el lenguaje 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. Fue desarrollado a principio de los años 90 por Sun Microsystems con la idea inicial de usarlo para la creación de páginas web.

La programación en java permite el desarrollo de aplicaciones bajo el esquema cliente – servidor, como de aplicaciones distribuidas.

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

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

general. Java permite programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. [27]

1.7 Servidor de Aplicaciones

El servidor JBoss es un proyecto de código abierto, con el que se consigue un servidor de aplicaciones basado en J2EE, e implementado al 100% en Java; soporta todas las funcionalidades de J2EE 1.4 e incluye servicios adicionales como clustering, caching y persistencia. También soporta Enterprise Java Beans, arquitectura componente del lado del servidor para la plataforma Java, que permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos.

Uno de los rasgos más importantes de JBOSS es su apoyo a la implementación "en caliente". Lo que esto significa que implementar un nuevo EJB es tan simple como copiar el archivo correspondiente en el directorio correspondiente. Si esto se hace mientras el Bean ya está cargado, JBOSS lo descarga automáticamente, y entonces carga la nueva versión.

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 Gestores de Bases de Datos

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad; actúa como interfaz entre la Bases de datos y el usuario. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

El **PostgreSQL**, es un SGBD, que constituye una mejora del sistema de base de datos Postgres, ya que mantiene los puntos fuertes de su antecesor como ser su poderoso modelo de datos y toda su riqueza en cuanto a tipos de datos, y reemplaza el lenguaje de consultas PostQuel con un subconjunto más extenso

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

de SQL. Es un proyecto Open Source mucho más puro, ya que no tiene empresas comerciales detrás como es MySQL o Firebird (en un principio fue Borland). [28]

1.9 Metodologías de desarrollo de software.

El propósito de las metodologías de software es establecer un contrato social entre todos los participantes en un proyecto para conseguir la solución más eficaz con los recursos disponibles.

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

1.9.1 Integración de modelos de madurez de capacidades (CMMI)

Integración de modelos de madurez de capacidades (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”.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

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 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

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

las personas que se encargarán del desarrollo posterior. Para llevar a cabo estas actividades es utilizado el documento IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos realizado por la universidad y cuyo objetivo es definir el proceso de administración de requisitos.

Este documento establece el ciclo de vida a seguir asociado a los proyectos involucrados en el proceso de mejora, el cual consta de 9 fases y se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico del área Administración de Requisitos.

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 necesaria 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).

1.9.2 El Proceso Unificado de Desarrollo de Software (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 finaliza con un hito donde se debe tomar una decisión importante.

Para la descripción de los productos de trabajo resultantes de cada uno de los flujos de trabajo de RUP así como de las fases del ciclo vida definido en el 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.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.9.3 Lenguaje Unificado de Modelado (UML)

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. [29]

1.10 Herramientas Utilizadas

Visual Paradigm for UML 6.1 Enterprise Edition: Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo.

Para el desarrollo de la aplicación web, se tiene al Entorno de desarrollo integrado (IDE) **Eclipse SDK** en su versión 3.3.2, con la colección de plug-ins de JBoos Tools, necesarios para utilizar los frameworks de desarrollo que se proponen.

1.11 Conclusiones

En este capítulo se profundizó en los conocimientos de conceptos necesarios para una correcta comprensión de la investigación. Se realizó todo un estudio sobre los modelos de HCI existentes eligiéndose la HCOP, como la HCI más conveniente para la APS dada sus características acordes a la atención médica brindada por la APS y su factibilidad para la informatización. Además de un análisis y descripción de las tecnologías y herramientas que serán utilizadas a lo largo del desarrollo del sistema propuesto.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En este capítulo se explica detalladamente la situación problemática existente y los conceptos fundamentales que se abordan en el desarrollo del modelo del dominio, modelo que sirve para especificar las condiciones, capacidades y cualidades que el sistema debe tener. Se presentan además los requisitos funcionales y no funcionales.

2.1 Problema y situación problemática

El creciente desarrollo de la informática ha producido un gran impacto social, cada día es mayor el número de instituciones y empresas que se inclinan por la incorporación de aplicaciones capaces de gestionar su información y de este modo, obtener una mayor dinámica y eficiencia en sus respectivos procesos de negocio.

La incorporación de sistemas informáticos en el área de la salud que apoyen la gestión de la información manejada por los profesionales de la salud aporta un notable incremento en la calidad de la información, así como una notable facilidad en la extracción de la misma.

En Cuba se desarrolla el Sistema Integral para la Atención Primaria de la Salud para la informatización de la APS, facilitando la gestión de la información tanto biomédica como administrativa en beneficio del primer nivel de atención de salud. El Sistema de Integral para la Atención Primaria de la Salud se encuentra conformado por diversos módulos, los cuales gestionan por separado, varias secciones de la Historia Clínica. Esta desunión causa problemas a la hora de visualizar los datos, ya que al no existir interoperabilidad entre los módulos se dificulta la consulta de la información, pues la información del paciente no estaría completa.

El Sistema Integral cuenta con 14 módulos a ser implementados según la necesidad para el sistema ligero o robusto:

- ✓ Módulo de Configuración: 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: gestiona los procesos básicos y fundamentales de la APS, es el núcleo del sistema, contiene los datos generales del paciente, Historia Clínica Individual, Historia

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Clínica Familiar, las planificación-seguimiento y control del paciente, los ingresos en el hogar y seguimiento diario, las remisiones, pesquisajes, 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 Clínico Quirúrgico: 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.
- ✓ Módulo de Medios Diagnóstico: gestiona la información de los laboratorios clínico y parasitología, microbiología, imagenología, endoscopia, drenaje biliar, ECG, laboratorio de alergia, optometría y audiometría en la atención primaria de salud.
- ✓ Módulo de Urgencias y Emergencias: gestiona la información de los cuerpos de guardia, consultas de urgencias, curaciones, traumatología, apoyo vital y trombolisis, salas de observación, inyecciones, aerosol y electrocardiograma de urgencias en la atención primaria de salud.
- ✓ Módulo de Salud Materno Infantil: gestiona la información de los hogares o casas de cuidado materno, planificación familiar, consultas de pediatría, consultas de obstetricia, consultas de ginecología, programas maternos-infantiles, salón de partos, regulación menstrual, tratamiento de pareja infértil y prevención de cáncer cérvico uterino en la atención primaria de salud.
- ✓ Módulo de Enfermería: 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.
- ✓ Módulo de Asistencia Social y Adulto Mayor: gestiona la información asociada a los trabajadores sociales relacionados con la salud, la información relacionada con las hogares o casas de cuidados de abuelos y los equipos multidisciplinarios de atención gerontológica en la atención primaria de salud.
- ✓ Módulo de Estomatología: gestiona la información de las clínicas estomatológicas de la atención primaria de salud, prótesis, ortodoncia, parodoncia, máxilo facial, rayos x, consejería bucal, laboratorio de prótesis, esterilización de estomatología y registros médicos y admisión de estomatología.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

- ✓ Módulo de Salud Mental: gestiona la información relacionada con la Psicometría, Psicología, Psiquiatría General e Infantil, defectología y consejería en la atención primaria de salud.
- ✓ Módulo de Rehabilitación Integral: gestiona la información de centros de rehabilitación y recuperación de la atención primaria de salud, medicina natural y tradicional, consulta de deshabituación tabáquica, podología, terapia ocupacional, gimnasio terapéutico, psicoprofiaxis del parto, calor IR y Masaje, Electroterapia, consulta de fisioterapia y médico MNT, consulta de defectología y logofoniatría, consulta de orientación nutricional y técnico de logofoniatría, peloidoterapia y parafina, hidroterapia, rehabilitación domiciliaria, consulta de calidad de vida y kinesioterapia.
- ✓ Módulo de Higiene y Epidemiología: gestiona la información relacionada con salud ambiental, escolar y del trabajador, consejería, vigilancia epidemiológica y farmacológica, promoción de salud y programas de prevención y control de enfermedades transmisibles, no transmisibles, escolares y ocupacionales.
- ✓ Módulo de Discapacitados Físicos: gestiona la información relacionada con el tratamiento, seguimiento y recuperación de los problemas de salud de los discapacitados físicos.
- ✓ Módulo de Servicios Generales y de Aseguramiento: gestiona la información relacionada con electromedicina, almacenes, lavandería, pantry-comedor, transporte y mantenimiento en la atención primaria de salud.

2.2 Modelo de Dominio

Es la representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades, el modelo del dominio podría considerarse como un diccionario visual de las abstracciones relevantes, vocabulario del dominio e información 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. [30]

2.2.1 Conceptos Fundamentales

Paciente: Término sanitario el cual se refiere a las personas que recibe un determinado servicio de un profesional de la salud.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Grupos Dispensariales: Clasificación del paciente según su problema de salud. Para ello existen cuatro grupos:

- Grupo I: son las personas aparentemente sanas
- Grupo II: conformado por las personas en riesgo o con factores de riesgo
- Grupo III: compuesto por las personas enfermas
- Grupo IV: que son aquellas personas presentan alguna discapacidad

Además de estas clasificaciones definidas en el sistema cubano, los técnicos pueden configurar nuevas clasificaciones para los grupos dispensariales.

Historia Clínica Individual: Es el documento clínico que registra en si misma toda la historia de salud de un paciente, tiene una función diagnóstica y es la base para la planificación, ejecución y control de las acciones de salud las cuales están encaminadas principalmente al fomento, recuperación y recuperación de la salud.

Factores de Riesgo: Es la tenencia por parte de un paciente de una característica o circunstancia, o conjunto de ellas, las cuales están asociadas a la posibilidad de tener o padecer un determinado problema de salud. Existen factores de riesgo ambientales, de comportamiento, biológicos, económicos, socio-culturales entre otros, pudiendo estar presente en una misma persona más de uno aumentando de este modo la posibilidad de tener o padecer un problema de salud.

Antecedentes Patológicos Familiares: Son el registro de enfermedades o padecimientos de un familiar, los cuales son necesarios tener en cuenta durante la vida del paciente pues pueden afectar o no, a corto, mediano o largo plazo la salud del mismo.

Tarjeta de vacunación: Es la información en forma de tabla sobre las vacunas administradas a un paciente, en la cual se puede apreciar en qué momento y que dosis le fue suministrada.

Problema: "...todo aquello que requiera diagnóstico, manejo posterior o interfiera con la calidad de vida, de acuerdo con la percepción del paciente" [31]

Los problemas pueden ser clasificados en dependencia de su duración o vigencia.

Según su duración:

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

- **Crónicos:** Son problemas de un largo periodo de duración, normalmente requieren de una mayor continuidad en la atención médica. En ellos también es importante considerar factores de riesgo que requieran de una especial vigilancia por su alto por ciento de afectar la salud del paciente.
- **Agudos:** Son problemas que presenta un corto periodo de duración y que normalmente no causan grandes daños a la salud del paciente y pueden ser tratados mediante tratamientos convencionales.

Según su Vigencia:

- **Activos:** Son los problemas que demandan de acciones de salud inmediatamente o en el futuro.
- **Inactivos:** Son problemas que no requieren de una acción médica pero es necesario mantener un cuidado sobre ellos pues pueden activarse en un momento determinado como pueden ser entre otros las alergias.

Lista de Problemas Personales: Contiene el registro de todos los problemas de salud, los sociales y demográficos, los psicológicos. Dada las diferentes clasificaciones de los problemas, la lista de problemas se divide en dos listas:

- **Lista de problemas agudos:** Es como una lista de trabajo la cual contiene problemas volátiles o de corta duración.
- **Lista de problemas crónicos/maestra:** Contiene la lista de los problemas más firmes, los cuales son de larga duración.

Además de estas clasificaciones definidas en el sistema cubano, los técnicos pueden configurar nuevas clasificaciones para los problemas.

Plan Inicial: Se establece para cada problema priorizado de un paciente con el propósito de establecer un plan de acción inicial con el objetivo de restablecer la salud.

Notas de la Evolución: Es el registro de la evolución de cada problema. Para ello se apoya en el uso del llamado acrónimo SOAP:

2.2.2 Diagrama del Modelo de Dominio

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

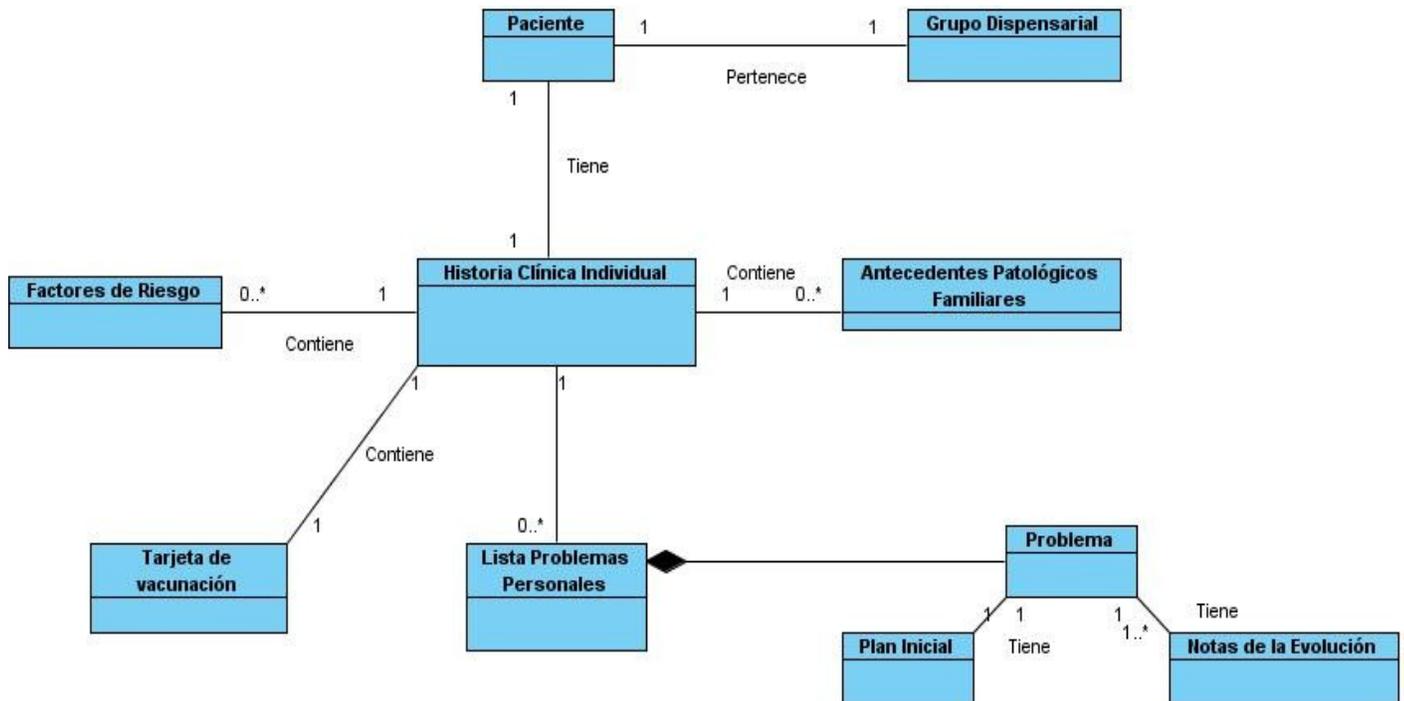


Figura 2.1 Modelo del dominio

2.3 Propuesta del Sistema

Para resolver el problema planteado se dispone crear un visor de Historia Clínica para el Sistema Integral de Atención Primaria de la Salud, el cual facilitará la gestión de consulta de la información y garantizará la interoperabilidad entre los sistemas que se encuentren en uso dentro del Sistema Integral.

La concepción del modelo de datos del SIAPS responde a la utilización de la HCOP permitiendo la visualización de las partes de la misma:

- BD Paciente
- Lista de Problemas
- Notas de la Evolución
- Plan inicial

El mismo enriquece la HCOP con elementos del modelo de HCI utilizado actualmente añadiendole:

- Tarjeta de vacunación
- Factores de riesgo
- Antecedentes patológicos familiares

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

-Índice de masa corporal

2.4 Especificación de los requerimientos de software

Un requerimiento de software es la capacidad o condición que tiene que lograr un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Estos se dividen en requerimientos funcionales y requerimientos no funcionales.

2.4.1 Requerimientos funcionales.

A continuación se muestra un listado de los objetivos que el sistema debe cumplir:

- RF1 Buscar Paciente
- RF2 Ver datos generales paciente.
- RF3 Configurar íconos.
- RF4 Antecedentes patológicos familiares
- RF5 Antecedentes patológicos personales
- RF6 Hoja de flujo medicación.
- RF7 Exámenes complementarios.
- RF8 Índice de Masa Corporal (IMC).
- RF9 Lista de problemas agudos.
- RF10 Problemas del paciente.
- RF11 Tarjeta de vacunación.

2.4.2 Requerimientos no funcionales.

Los requisitos no funcionales se refieren a las propiedades o cualidades del sistema. Estos responden a cualidades que el producto debe tener y las características para que este sea atractivo, confiable, usable y seguro.

2.4.2.1. Seguridad

- 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.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

- 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.
- El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.
- Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros policlínicos por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.

2.4.2.2. Restricciones de diseño

La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio. La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

2.4.2.3. Interfaz de usuario

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

2.4.2.4. Requerimientos de hardware

Estaciones de trabajo

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En la solución se incluyen estaciones de trabajo para las consultas del Sistema para la Atención Primaria alas SIAPS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web. Por lo que se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz.

Servidores

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de Base de datos: Procesador Dual - Core 4GB de memoria y 2x72GB de disco. Servidores de Aplicaciones: Procesador Dual - Core 4GB de memoria y 2x72GB de disco.

2.4.2.5. Requerimientos de 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.

2.5. Conclusiones

En este capítulo, se explica necesidad de la creación de un visor de historias clínicas existente así como los conceptos fundamentales que se abordaron para la creación del modelo de dominio. Quedando vigente que es la solución para lograr la relación entre los módulos existentes en el Sistema Integral para la Atención Primaria de la Salud. La propuesta del sistema permitió definir la utilización de la Historia Clínica Orientada al Problema enriquecida con elementos de la Historia Clínica Cronológica por Secciones. Se establecieron los requerimientos funcionales y no funcionales, estableciendo la propuesta del sistema.

CAPÍTULO 3. DISEÑO DEL SISTEMA

En el presente capítulo se expone el diseño propuesto para la solución de la aplicación, modelándose los artefactos necesarios que contribuyen a la implementación del sistema. Se muestran los principales diagramas de clases y secuencia del modelo diseño. Además de describir la arquitectura del sistema y su aplicación en el Visor de Historia Clínica.

3.1. Descripción de la arquitectura, fundamentación.

Un sistema software es una única entidad, pero al arquitecto y a los desarrolladores les resulta útil presentar el sistema desde diferentes perspectivas para comprender mejor el diseño. Estas perspectivas son vistas del modelo del sistema. Todas las vistas juntas representan la arquitectura.

3.2. Patrones de arquitectura

Para el desarrollo de este sistema el patrón más utilizado es el Modelo-Vista-Controlador, que ayuda a la hora de la organización del código, ya que se encontraría separada la lógica de control, los datos de la aplicación y la interfaz del usuario en tres componentes distintos. Al proponerse como base de la arquitectura la plataforma de desarrollo Java, posibilita que se utilicen librerías y especificaciones que evidencian el uso del patrón.

El modelo que se encarga del almacenamiento de los datos, su estructura y las relaciones entre los mismos. Para el acceso a datos se utiliza el framework Hibernate que proporciona una gran independencia del gestor de Base de datos, controla la accesibilidad a la información almacenada, así como la realización de consultas rápidas; específicamente se usa la implementación de Java Persistence Api (JPA) de Hibernate 3.3, minimizando por un lado las configuraciones en XML débilmente definidas por un tipo y por otro lado usando los servicios del contenedor de Enterprise Java Beans en su versión 3.0 (EJB3), se elimina gran parte del código "infraestructural" en cuanto a transacciones, la transmisión del contexto de persistencia, etc..

El controlador que engloba toda la lógica del negocio, está representado por las clases controladoras y usa como framework SEAM, el cual unifica las tecnologías estándares JSF, EJB3, JPA garantizando crear una capa consistente que impide todo tipo de errores. Seam provee una mayor granularidad de contextos

CAPÍTULO 3. DISEÑO DEL SISTEMA

de estado, el principal quizá es el contexto conversacional, con esto se logra un uso más eficiente de la memoria evitando memory-leaks. Gracias al contexto de persistencia extendido implementado por Seam y basado en Hibernate, se puede dar uso del concepto de conversación que permite definir funcionalidades en las que se involucren varios request donde los cambios hechos a las entidades solo son persistidos al final de la conversación.

El framework Jboss Seam integra los diferentes elementos existentes en la aplicación ya que se encarga de abstraer al desarrollador del funcionamiento interno, permitiendo la utilización de anotaciones con las que es posible acceder al modelo de datos directamente desde la vista, característica que permite un mejor funcionamiento del sistema.

En la vista residen todas las clases interfaces con las que interactúa directamente el usuario, estas clases están construidas mayormente con componentes Java Server Faces (JSF) y usan la librería de componentes Richfaces, además del framework de presentación Facelets y encontrarse presentes controles como Ajax4Jsf. Estos componentes hacen peticiones al controlador a través de eventos que lanzan; y son controlados por un *servlet*, encargado del ciclo de vida de estos y que a su vez, transporta los datos provenientes de las controladoras, actualizando de forma eficiente la interfaz de usuario.

Otro de los lenguajes utilizados en la capa de presentación es el Java Script, gracias a su característica de ser un lenguaje de programación interpretado, es decir, que no requiere compilación; lo cual permite ejecutar código en el lado del cliente contribuyendo con la rapidez en la conexión que se establece entre el cliente y el servidor. [32]

3.3. Valoración crítica

Lo anteriormente detallado plantea grandes ventajas del uso de la arquitectura definida, pero existen inconvenientes con su aplicación para desarrollar el sistema en proposición.

Durante el proceso de desarrollo de software, existen cambios en la base de datos del sistema que se encuentra implementando. Normalmente los cambios en la base de datos de un sistema crean problemas o alteraciones en el proceso de concepción del mismo, mas, la problemática debe manejarse de forma tal que los cambios provoquen un impacto mínimo en la implementación del sistema.

CAPÍTULO 3. DISEÑO DEL SISTEMA

La solución propuesta posee una alta dependencia a nivel de base de datos. El intercambio de información entre los módulos del sistema se realiza mediante consultas directas a los esquemas de la base de datos de cada uno de los mismos. Esta relación tan estrecha provoca que cualquier cambio en los esquemas que son utilizados por otros módulos del sistema, perjudique el desarrollo de los módulos correspondientes.

La capa de acceso a datos encapsula toda la lógica de almacenamiento, independizando al resto del sistema del mecanismo de persistencia. Esta capa es la encargada de persistir las entidades que se manejan en negocio, el acceso a los datos almacenados, la actualización, etc. Esto permite que los módulos del cliente sean creados con un de alto nivel de abstracción.

En la arquitectura que se utilizará para el desarrollo de la solución propuesta, la capa de acceso a datos solo se limita a representar, a través del ORM Hibernate las tablas de la base de datos mediante las clases entity. Las consultas que se realizan para obtener los datos, ya sea al esquema perteneciente al módulo o a otro esquema del cual se necesite información, se hacen directamente desde las clases controladoras, por lo que el programador debe preocuparse por la forma en que están representados de los datos.

Una posible solución al problema es la creación, de una clase “fachada” donde existan métodos que permitan acceder a los datos manejados en cada módulo; de manera tal que si cambiara en su estructura el modelo de alguno de estos módulos, no se comprometa el negocio del resto de los módulos que lo utilizan. Estas clases formarían parte de la capa de negocio.

3.4. Diseño

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe como implementar el sistema. Debe ser suficientemente específico para que el sistema pueda ser implementado sin ambigüedades.

3.5. Modelo del diseño

Es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. [33]

CAPÍTULO 3. DISEÑO DEL SISTEMA

Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas. Es utilizado como una entrada fundamental para las subsiguientes actividades de implementación. Entre los artefactos del modelo de diseño se encuentran: descripción de la arquitectura y realización de casos de uso.

3.6 Patrones de diseño

Un patrón es una solución a un problema que aparece con frecuencia. Se presentan como plantillas donde se les asigna un nombre y un resumen de los problemas, las fuerzas que lo hacen surgir, y las ventajas y desventajas que provee su utilización.

Entre los que más se utilizan se encuentran los patrones GRASP que son los patrones para la asignación de responsabilidades, donde se destacan por su uso en el diseño:

- **Experto:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Alta cohesión:** La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Este patrón expresa que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase.
- **Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí posible. De tal forma que en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre clases.
- **Creador:** Este patrón es el encargado de asignarle a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos:
 - ✓ B agrega los objetos A.
 - ✓ B contiene los objetos A.
 - ✓ B registra las instancias de los objetos A o
 - ✓ B utiliza especialmente los objetos A.
 - ✓ B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A). B es un creador de los objetos A.

Si existe más de una opción, prefiera la clase B que agregue o contenga la clase A. [34]

CAPÍTULO 3. DISEÑO DEL SISTEMA

- Controlador: Asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

3.7 Definición de elementos de diseño

En la realización de los diagramas de diseño se utilizan tres tipos de estereotipos de clases fundamentales: Client page (páginas servidoras) que son las que interactúan con el usuario, encargándose de mostrar la información, Server page (página servidora) que se encarga de construir o generar las páginas clientes y los Form (formularios) que son los que se encargan de enviar los datos a la página servidora para que sean procesados los pedidos.

3.8 Diagramas de clases del diseño

3.8.1 Mostrar Datos Generales del Paciente

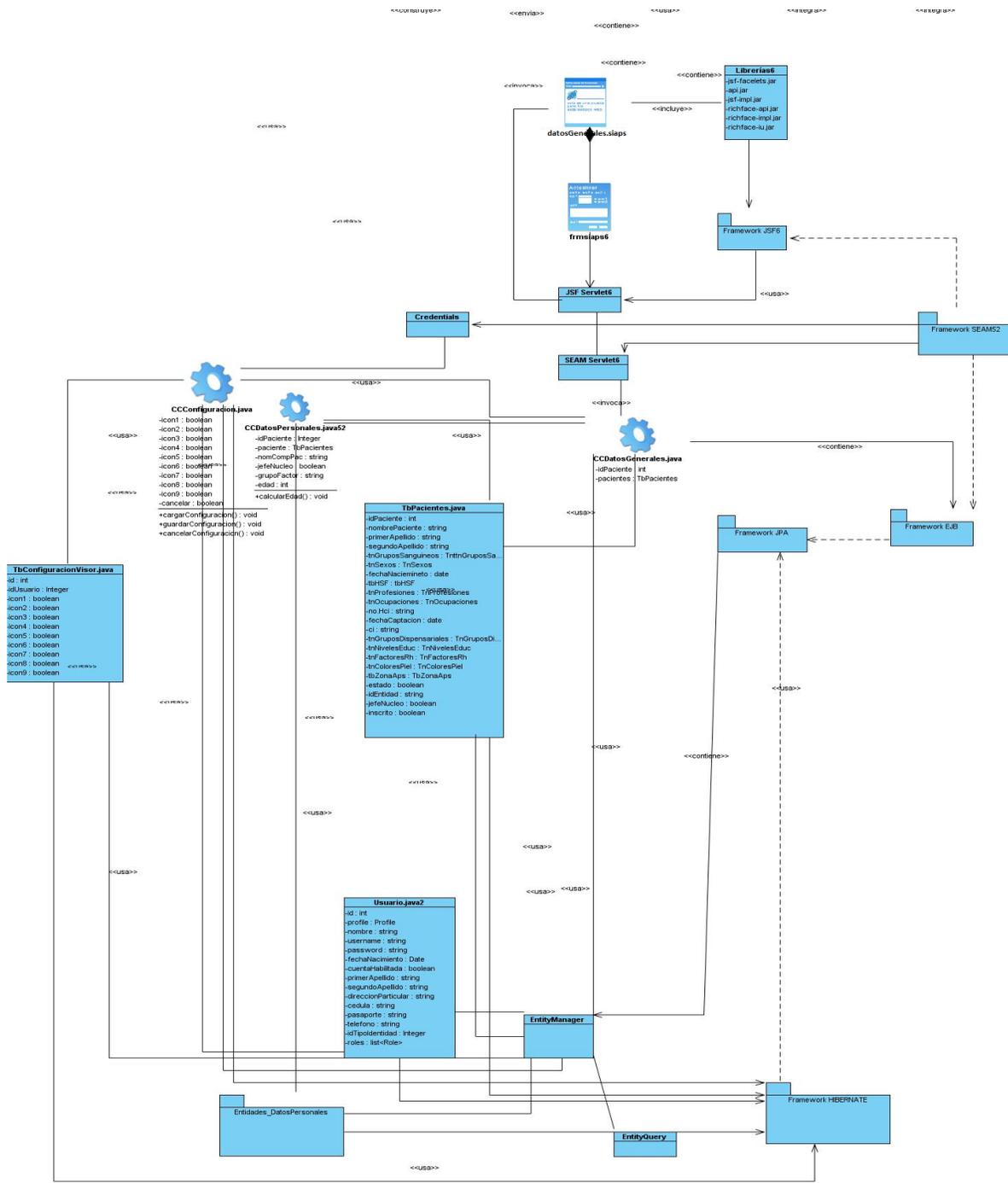


Figura 3.1 Diagrama de clases del diseño- Mostrar antecedentes patológicos familiares

CAPÍTULO 3. DISEÑO DEL SISTEMA

3.8.3 Descripción de las clases del diseño

A continuación se describen algunas de las clases que han sido identificadas para la futura implementación del sistema, con el objetivo de lograr una mayor comprensión del sistema.

En un inicio se describen las clases comunes a los diversos diagramas de diseño definidos.

Nombre: CCdatosPersonales.java
Tipo de clase: controladora
Descripción: La clase CCdatosPersonales.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. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Tabla 3.1 Descripción de la vista<CCdatosGeneralesPac.java>

Nombre: CCConfiguracion.java
Tipo de clase: controladora
Descripción: La clase CCConfiguracion.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. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Tabla 3.2 Descripción de la vista<CCConfiguracion.java>

3.8.3.1 Datos Generales del paciente.

Nombre: DatosGenerales.siaps
Tipo de clase: interfaz

CAPÍTULO 3. DISEÑO DEL SISTEMA

Descripción: Página web que se ejecuta del lado del cliente, le permite al profesional de la salud la visualización de todos los datos generales del paciente.

Tabla 3.10 Descripción de la vista< DatosGenerales.siaps>

Nombre: CCDatosGenerales.java
Tipo de clase: controladora
Descripción: La clase CCDatosGenerales.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. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Tabla 3.11 Descripción de la vista<CCDatosGenerales.siaps>

Nombre: TbPacientes.java
Propósito: Proveer el mapeo con la base de datos.
Tipo de clase: entidad
La clase TbPacientes.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente DatosgeneralesPaciente.siaps Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.12 Descripción de la vista<TbPacientes.siaps>

Nombre: Usuario.java
Propósito: Proveer el mapeo con la base de datos.
Tipo de clase: entidad

CAPÍTULO 3. DISEÑO DEL SISTEMA

La clase Usuario.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente DatosgeneralesPaciente.siaps Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.15 Descripción de la vista< Usuario.siaps>

3.8.3.2 Antecedentes patológicos familiares.

Nombre: patologicosFamiliares.siaps

Tipo de clase: interfaz

Descripción: Página web que se ejecuta del lado del cliente, le permite al profesional de la salud la visualización de los antecedentes patológicos de un paciente.

Tabla 3.3 Descripción de la vista< patologicosFamiliares.siaps>

Nombre: CCAntecedentesPatologicosFamiliares.java

Tipo de clase: controladora

Descripción: La clase CCAntecedentesPatologicosFamiliares .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. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.

Tabla 3.4 Descripción de la vista<CCAntecedentesPatologicosFamiliares.java>

Nombre: TbPaciente.java

Propósito: Proveer el mapeo con la base de datos.

Tipo de clase: entidad

CAPÍTULO 3. DISEÑO DEL SISTEMA

La clase TbPaciente.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente patologicosFamiliares.siaps. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.5 Descripción de la vista<TbPaciente.java>

Nombre: TbConsultaPaciente.java

Propósito: Proveer el mapeo con la base de datos.

Tipo de clase: entidad

La clase TbConsultaPaciente.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente patologicosFamiliares.siaps. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.6 Descripción de la vista< TnParentescos.java>

Nombre: TrConsultaPacProblema .java

Propósito: Proveer el mapeo con la base de datos.

Tipo de clase: entidad

La clase TrConsultaPacProblema.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente patologicosFamiliares.siaps. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.7 Descripción de la vista<TbProblemaSIdPaciente.java>

CAPÍTULO 3. DISEÑO DEL SISTEMA

Nombre: TrParentescosPaciente .java
Propósito: Proveer el mapeo con la base de datos.
Tipo de clase: entidad
La clase TrParentescosPaciente .java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente patologicosFamiliares.siaps. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.8 Descripción de la vista< TrParentescosPaciente.java>

Nombre: TbProblemasSldPaciente.java
Propósito: Proveer el mapeo con la base de datos.
Tipo de clase: entidad
La clase TnCpapsEnfermedad.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente patologicosFamiliares.siaps. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.9 Descripción de la vista< TnCpapsEnfermedad.java>

Nombre: Problema.java
Propósito: Proveer el mapeo con la base de datos.
Tipo de clase: entidad
La clase Problema.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente patologicosFamiliares.siaps. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

CAPÍTULO 3. DISEÑO DEL SISTEMA

Tabla 3.9 Descripción de la vista< Problema.java>

Nombre: Usuario.java
Propósito: Proveer el mapeo con la base de datos.
Tipo de clase: entidad
La clase Usuario.java es una página que se ejecuta del lado del servidor en la Capa de Presentación. Recibe y valida los datos que se envían desde la página cliente patologicosFamiliares.siaps. Codifica la información y construye las estructuras que serán enviadas a la Capa de Negocio. Se invoca al método del negocio para el registro de los nuevos datos, una vez concluida la ejecución de sus responsabilidades.

Tabla 3.9 Descripción de la vista< Usuario.java>

3.9 Conclusiones

En este capítulo se describió la arquitectura utilizada para implementar el sistema propuesto. Se justificaron los patrones utilizados, tanto los de diseño como los de arquitectura. La asimilación de la arquitectura permitió realizar una valoración crítica de la arquitectura definida, concluyendo que la misma posee problemas de flexibilidad para el desarrollo del sistema. Los requerimientos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a desarrollar.

CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo se describe la integración con otros componentes. Se muestra el modelo de implementación con el diagrama de despliegue. Se realiza una descripción detallada de los métodos más complejos en el desarrollo de la aplicación o agentes en el proceso de implementación, así como los estándares de diseño, codificación y tratamientos de errores utilizados.

4.1. Modelo de despliegue.

El modelo de despliegue no es más que la representación física de los nodos, es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. [35]

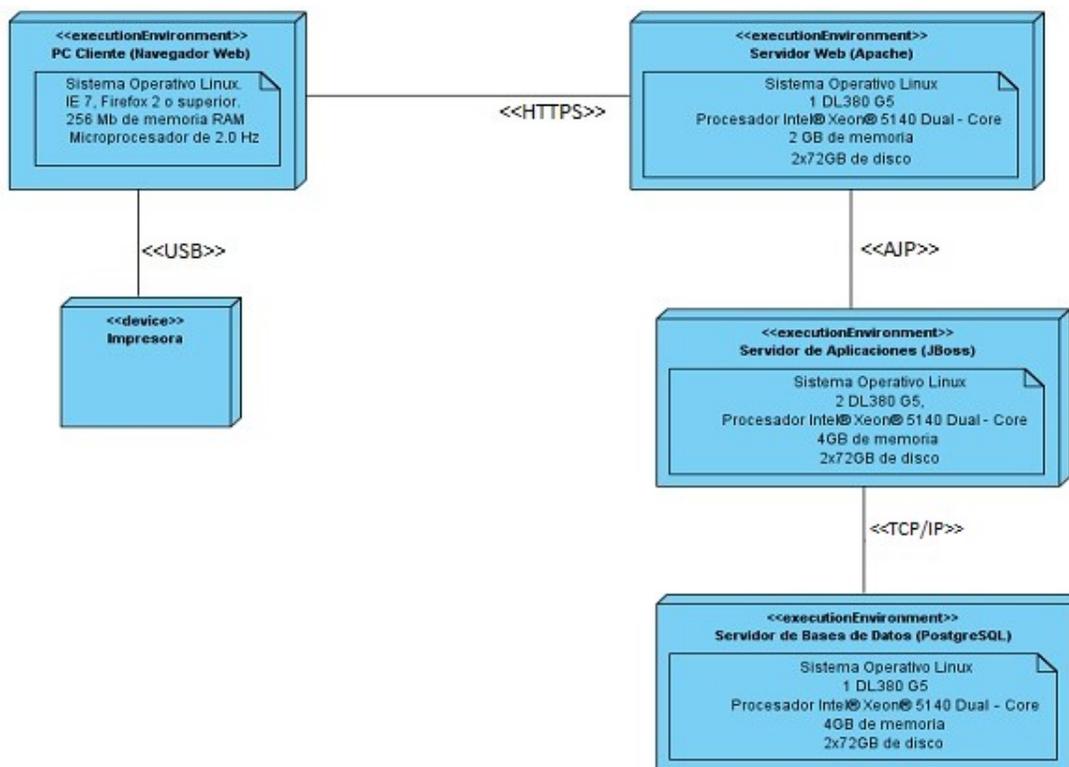


Figura 4.1 Diagrama de despliegue

CAPÍTULO 4. IMPLEMENTACIÓN

4.2 Descripción de Nodos

Estaciones Clientes (Navegador Web): Serán las estaciones de los usuarios, las cuales servirán para acceder al sistema web o integrado mediante el navegador.

Servidor Web (Apache): Servidor Web Apache que provee el servicio de interfaz al usuario final mediante un portal convencional y otro WAP para dispositivos móviles, pues es el ordenador que estará como fachada a Internet, al mismo tiempo será el puente o proxy para entrar al clúster de servidor (res) de aplicaciones que proporciona Jboss. En escenarios híbridos brindará la ejecución y actualización de la solución local mediante la tecnología Java Web Start.

Servidor de Aplicaciones (JBoss): Servidor de Aplicaciones Jboss certificado por SUN para el estándar JEE5, hospedará la solución integrada, proveerá de un clúster para balancear la carga de peticiones hechas por los usuarios garantizando de esta manera disponibilidad de la información mostrada.

Servidor de Bases de Datos (PostgreSQL): Servidor de Datos PostgreSQL, en el cual residirá toda la información operacional, que será alimento del repositorio central de historias clínicas electrónicas.

Dispositivo Impresora: Dispositivo conectado a las estaciones clientes de los usuarios que permitirá realizar la impresión de los reportes del sistema.

4.3 Integración de con otros módulos

Los sistemas por necesidad están llamados de una manera u otra a la integración con otros en funcionamiento o en desarrollo, incluso con sistemas que aun no estén concebidos. El módulo del Visor de Historia Clínica no está excluido de esta realidad, al contrario, el mismo mantiene en la actualidad estrecha relación con los cinco módulos en desarrollo en las Siaps. La solución propuesta se nutre de la información de los restantes módulos permitiendo mostrar toda la información conformando una Historia Clínica configurable que sirva de complemento al profesional de la salud en todas sus labores asistenciales, investigativas y docentes. El Visor como objetivo plantea la relación con los módulos que aún se encuentran en desarrollo, de los cuales tomaría, de manera similar a la relación que posee con los módulos en existencia, información para mostrar.

Queda manifestada la integración al Sistema Integral para la Atención Primaria de la Salud, la cual se realiza a nivel de Base de Datos.

CAPÍTULO 4. IMPLEMENTACIÓN

4.4 Estándares de codificación.

Un estándar de codificación no es más que “Un conjunto de reglas de notación y nomenclatura, específicas de cada lenguaje de programación, que se usan y se siguen durante la fase de implementación (codificación) de una aplicación y reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores que no son detectados por los compiladores, reduciendo el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismos.” [36]

Algunas de las convenciones más importantes de este estándar son las siguientes:

Declaraciones de Clases e Interfaces.

Número	Partes de la Declaración de una Clase o Interface	Descripción
1	Comentario de documentación de la clase o interface. (<i>/** ... */</i>)	
2	Sentencia class o interface.	
3	Comentario de implementación de la clase o interface si fuera necesario. (<i>/* ... */</i>)	Este comentario debe contener cualquier información aplicable a toda la clase o interface que no era apropiada para estar en los comentarios de documentación de la clase o interface.
4	Variables de clase (static).	Primero las variables de clase public, después las protected, después las de nivel de paquete (sin modificador de acceso), y después las private.
5	Variables de instancia.	Primero las public, después las protected, después las de nivel de paquete (sin modificador de acceso), y después las

CAPÍTULO 4. IMPLEMENTACIÓN

		private.
6	Constructores.	
7	Métodos.	Estos métodos se deben agrupar por función más que por visión o accesibilidad. Por ejemplo, un método de clase privado puede estar entre los métodos públicos de instancia. El objetivo es hacer el código más legible y comprensible.

Tabla 4.1 Declaraciones de Clases e Interfaces.

Rompiendo Líneas.

Cuando una expresión no entre en una línea, romperla de acuerdo con estos principios:

- Romper después de una coma.
- Romper antes de un operador.
- Preferir roturas de alto nivel (más a la derecha que el "padre") que de bajo nivel (más a la izquierda que el "padre").
- Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.
- Si las reglas anteriores llevan a código confuso o a código que se aglutina en el margen derecho, indentar justo 8 espacios en su lugar.
- Ejemplos de cómo romper la llamada a un método:

```
unMetodo(expresionLarga1, expresionLarga2, expresionLarga3,  
         expresionLarga4, expresionLarga5);
```

```
var = unMetodo1(expresionLarga1,  
               unMetodo2(expresionLarga2,  
                         expresionLarga3));
```

Comentarios de Bloque.

Los comentarios de bloque se usan para dar descripciones de ficheros, métodos, estructuras de datos y algoritmos. Los comentarios de bloque se podrán usar al comienzo de cada fichero o antes de cada método. También se pueden usar en otros lugares, tales como el interior de los métodos. Los comentarios

CAPÍTULO 4. IMPLEMENTACIÓN

de bloque en el interior de una función o método deben ser indentados al mismo nivel que el código que describen.

Un comentario de bloque debe ir precedido por una línea en blanco que lo separe del resto del código.

```
/*  
 * Aquí hay un comentario de bloque.  
 */
```

Los comentarios de bloque pueden comenzar con /*, que es reconocido por indent (1) como el comienzo de un comentario de bloque que no debe ser reformateado. Ejemplo:

```
/*_  
 * Aquí tenemos un comentario de bloque con cierto  
 * formato especial que quiero que ignore indent(1).  
 *  
 * uno  
 *   dos  
 *     tres  
 */
```

Declaraciones de Clases e Interfaces.

Al codificar clases e interfaces de Java, se siguen las siguientes reglas de formato:

- Ningún espacio en blanco entre el nombre de un método y el paréntesis "(" que abre su lista de parámetros.
- La llave de apertura "{" aparece al final de la misma línea de la sentencia declaración.
- La llave de cierre "}" empieza una nueva línea indentada para ajustarse a su sentencia de apertura correspondiente, excepto cuando no existen sentencias entre ambas, que debe aparecer inmediatamente después de la de apertura "{".

```
class Ejemplo extends Object {
    int ivar1;
    int ivar2;

    Ejemplo(int i, int j) {
        ivar1 = i;
        ivar2 = j;
    }

    int metodoVacio() {}

    ...
}
```

- Los métodos se separan con una línea en blanco.

Referencias a Variables y Métodos de Clase.

Evitar usar un objeto para acceder a una variable o método de clase (static). Usar el nombre de la clase en su lugar. Por ejemplo:

```
metodoDeClase();           //OK
UnaClase.metodoDeClase();  //OK
unObjeto.metodoDeClase();  //EVITAR!
```

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.

4.5 Seguridad

Para que cualquier sistema informático sea calificado como seguro debe contar con un correcto equilibrio entre las siguientes características:

CAPÍTULO 4. IMPLEMENTACIÓN

Integridad: La información sólo puede ser modificada por quien está autorizado y de manera controlada.

Confidencialidad: La información sólo debe ser legible para los autorizados.

Disponibilidad: La información debe estar disponible cuando se necesita.

Irrefutabilidad (No repudio): El uso y/o modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción.

En correspondencia con dichas características se plantean a continuación un conjunto de acciones que, llevadas a cabo, permitirá a los usuarios finales disfrutar de un software seguro:

- Se mantendrá seguridad y control a nivel de usuario (Ver figura 4.7), 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.
- 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.
- Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude. Los mecanismos serán capaces de informar al personal autorizado sobre posibles irregularidades que den indicios sobre la introducción de información falseada.
- El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el subsistema, este elemento ya no exista.
- Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros policlínicos por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.



Figura 4.7 Acceso al Sistema.

4.6 Tratamiento de excepciones

Las excepciones son errores o eventos que interrumpen el proceso de ejecución de un sistema, impidiendo el correcto funcionamiento de la aplicación. El tratamiento de estas excepciones, es lo que permite que el programa continúe funcionando aun con error, ya que se transfiere el control al manipulador adecuado donde va a ser tratado.

En el sistema propuesto, el tratamiento de excepciones se realiza mediante la sentencia try-catch-finally, try para generar la excepción en caso de que alguna sentencia que se encuentra dentro del bloque lo requiera y catch para atraparla.

El tratamiento de errores está presente en todos los fragmentos de código donde pueda existir un error, fundamentalmente en aquellas donde exista selección de datos de una base de datos, con el objetivo de evitar que los datos manipulados no estén en el formato correcto.

El sistema cuenta con un archivo XML (page.xml) que posee según el tipo de excepción, la configuración de todos los mensajes que se deben mostrar. El mismo archivo además posee la pagina a la que se redirecciona en caso de aparecer un error inesperado. También se usa el componente SeamFacesMessages, del framework Seam, el que se encarga de capturar los mensajes de error o notificación que aparecen en el código de las clases controladoras, y permite mostrarlos en vistas o interfaces de usuario.

A continuación se muestra la arquitectura que JPA tiene para el control de las excepciones:

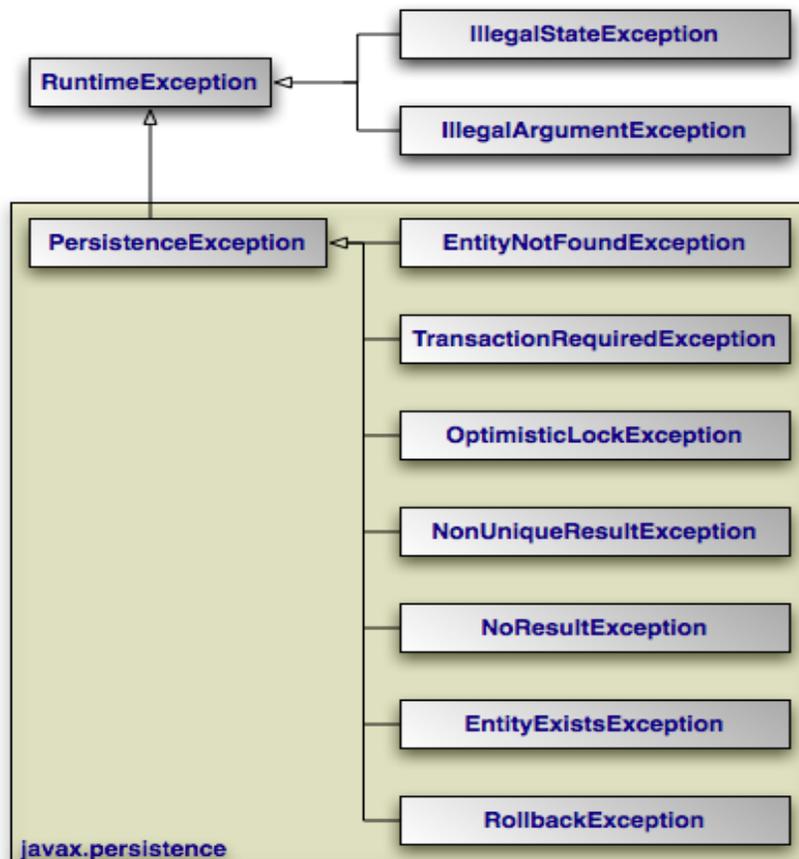


Figura 4.8

4.7 Descripción de los métodos más importantes

Antecedentes patológicos familiares

- `nombreParentesco`

El método devuelve una lista de arreglos string donde en una primera posición se devuelve la descripción del parentesco del paciente (made, padre, hermanos,.....) y en una segunda posición el identificador de cada familiar respectivamente. Está creado con la idea de mostrar dentro de la página

de Antecedentes Patológicos Familiares, una tabla que se confeccione dinámicamente con la descripción de los familiares de un paciente.

- **obtenerProblemas**

Este método devuelve una matriz con los problemas crónicos asociados a cada familiar de un paciente. Se complementa con el método **nombreParentesco**, donde llenaría los datos de los problemas crónicos de los familiares asociados a la descripción de que trae el método anteriormente expuesto, conformando una tabla dinámica donde quedarían reflejados los familiares del paciente y sus respectivos problemas crónicos o los denominados antecedentes patológicos personales del paciente.

Buscar Pacientes

- **reiniciarBusqueda**

Este método anula los criterios de búsqueda, posibilitando listar todos los pacientes que se encuentren en la base de datos a la hora de realizar la búsqueda.

- **cambiar**

El mismo no devuelve valores; su objetivo es posibilitar dentro de la página Buscar Pacientes la visualización de la búsqueda avanzada y la búsqueda básica. Recibe como entrada una variable booleana que es la que maneja la configuración de tal tarea, si la variable es verdadera muestra la búsqueda avanzada, de lo contrario anula los campos de la búsqueda avanzada y mostraría la búsqueda básica.

- **recibirdir**

El método recibe un identificador de dirección y a través del mismo realiza la búsqueda del número de historia de salud familiar el cual se almacena en el campo de búsqueda No. HSF.

4.8. Conclusiones

En este capítulo quedó reflejada la integración de la solución propuesta con otros componentes, donde el mismo se relaciona con todos los módulos del sistema alas SIAPS. Se presentó el modelo de implementación con el diagrama de despliegue, así como los estándares de codificación definidos para la implementación.

CONCLUSIONES

Al terminar la presente investigación, se han podido cumplir los objetivos y tareas propuestas, determinando como conclusiones:

- El estudio de los modelos actuales de la Historia Clínica Individual permitió definir la utilización de la Historia Clínica Orientada al Problema y la Historia Clínica Cronológica por Secciones.
- Se realizó el diseño de la solución en correspondencia con la arquitectura propuesta, obteniéndose un sistema robusto y flexible capaz de gestionar la información relacionada con la visualización de los datos del paciente dentro de la Atención Primaria de la Salud.
- Se desarrolló el módulo de Visor de Historia Clínica, que facilita la consulta de información en la Atención Primaria de la Salud.

RECOMENDACIONES

Para el buen desempeño y puesta en marcha de la aplicación se recomienda:

- Enriquecer el sistema con gráficos que permitan un mejor entendimiento y visualización de las funcionalidades.
- Proponer su utilización y generalización en Cuba y en los países que lo requieran.
- Incluirle al visor la opción de visualizar CDA, con el objetivo de lograr la interoperabilidad semántica dentro del Sistema Nacional de Salud. Así se da paso a la utilización estándares que posibilitan la interoperabilidad sintáctica, logrando con ello que toda la información médica pertinente a un paciente sea enviada un repositorio de CDA en cual se conformaría la HCE única del paciente.
- Utilizar la solución propuesta para la arquitectura, detallada en la valoración crítica, con el objetivo de eliminar la alta dependencia existente a nivel de base de datos.

REFERENCIAS BIBLIOGRÁFICAS

1. Historia clínica: Enciclopedia Libre Universal en Español [sitio en Internet]. Disponible en: http://enciclopedia.us.es/index.php/Historia_cl%C3%ADnica [Consultado: 6 de mayo del 2005].
2. Torralba V. Estandarización de la historia clínica electrónica [sitio en Internet]. 2004. Disponible en: <http://greco.dit.upm.es/~tomas/cursos/isi/trabajos/2003/vtorralba.pdf> [Consultado: 12 de mayo del 2005].
3. Ídem a la 1.
4. Llanio Navarro, R. Propedéutica Clínica y Fisiopatología. La Habana: Ed. Pueblo y Educación, 1983:29-30.
5. Da Costa Carballo, Carlos Manuel. Otros documentos: la historia clínica.
6. Ídem a la 5.
7. Ídem a la 5.
8. Ídem a la 5.
9. Lain Entralgo, Pedro: El empirismo Clínico. En: Lain Entralgo. Pedro: Historia de la Medicina, Barcelona: Salvat (Colección: Biblioteca Médica tic Bolsillo. No 21). 1986 [309-320]
10. Uno de los tres grandes sistemáticos del setecientos junto con los alemanes Georg Ernst Stahl (1659-1734) y Friedrich Hotfmann (1660-1742)
11. Curso Universitario Sistemas de información en los sistemas de salud. AMIA 10x10. Unidad 6
12. Ídem a la 7.
13. Ídem a la 7.
14. Savage P. A book that changed my practice. Problem oriented medical records. BMJ. 2001; 322 (2): 275.
15. Starfield B. Primary care: concept, evaluation and policy. New York: Oxford University Press; 1992. Pag.404.
16. The National Alliance for Health Information Technology. What is Interoperability? 2005 [cited 2006; Disponible en: <http://www.nahit.org/alliance2/images/b4f854c1e68bf997c.pdf>.
17. Alonso Lanza, J.J. Ing. La historia clínica electrónica: ideas, experiencias y reflexiones. Artículos

REFERENCIAS BIBLIOGRÁFICAS

18. Sánchez Mansolo, Athos Alejandro. Historia Clínica Electrónica: Una herramienta para la asistencia, docencia e investigación en medicina, 1998.
19. Laya, M. Oracle Press Release: El Servicio Cántabro de Salud implementa Oracle HTB en su Visor de Historia Clínica Electrónica
http://www.oracle.com/global/es/corporate/press/np_serv_cantabro_salud_implementa_oracle.html
20. Ídem a la 14
21. Amarino, Mariano. Ajax un nuevo acercamiento a aplicaciones Web. 2005. Disponible en:
<http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>
22. JBoss Community. (2007). Recuperado el 28 de Diciembre de 2008, de JBoss Ajax4jsf. Introducción.
23. Framework JBoss Seam. Febrero 2008. Disponible en:
<http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>
24. Medisist. Enlace del médico moderno. [En línea] [Citado el: 3 de enero de 2009.]
<http://www.medisist.com.mx/enlace-del-medico-moderno.htm>.
25. Suárez González, Hector. Manual Hibernate. 2003. Disponible en:
<http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf>
26. Hermes. Mensajería HL7 V2.X y CDA. Disponible en:
http://www.fgr.cu/index.php?option=com_content&view=article&id=560:hermes-mensajeria-HL7-v2x-y-CDA&catid=64:tic&Itemid=89
27. 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.
28. Caraballo, Daniel; Madera, Mario; Odin, Marcelo. Estudio de GNU/Linux y el software libre como solución integral para empresas. 2004. p74. Disponible en:
<http://www.fing.edu.uy/~asabique/prgrado/2004eofgl/contenido/archivos/Anexo-II.pdf>
29. Jacobson, I. Booch, G. Rumbaugh, J. *El Proceso Unificado de Desarrollo de software*. s.l.: Addison-Wesley, 2000.
30. Marrero Nieblas, Ariuska y Suarez Rivero, Yoel Odelso. Modulo Banco de sangre del sistema de información hospitalaria alas HIS, 2009.
31. Idem a la 10.
32. web estilo. [En línea] <http://www.webestilo.com/javascript/>

REFERENCIAS BIBLIOGRÁFICAS

33. Raquel Ochoa Ornelas. Facultad de ingeniería mecánica y eléctrica. Universidad de Colima, 2003.
[Disponible en: http://digeset.ucol.mx/tesis_posgrado/Pdf/Raquel%20Ochoa%20Ornelas.pdf]
34. Conferencia 2 de Arquitectura 2010. Arquitectura y Patrones de diseño. Universidad de las Ciencias Informaticas. Disponible en : <http://eva.uci.cu/mod/resource/view.php?id=14075>
35. Ídem a la 29.
36. Ídem a la 30.

BIBLIOGRAFÍA

Alcantar Hernández, Fernando. 2008. GestioPolis. *Mapeo entre clases persistentes utilizando el software Hibernate My Eclipse para conectar a una base de datos con XML.*

Alonso Lanza, J.J. Ing. La historia clínica electrónica: ideas, experiencias y reflexiones. Artículos

Amarino, Mariano. Ajax un nuevo acercamiento a aplicaciones Web. 2005

Cantale, Carlos R. Dr. Historia Clínica Orientada a Problemas. 2007.

Caraballo, Daniel; Madera, Mario; Odin, Marcelo. Estudio de GNU/Linux y el software libre como solución integral para empresas. 2004. p74.

Colectivo de Autores. *Burocracia en las consultas de Atención Primaria e informatización: ¿hemos mejorado?*

Contreras, José L. *La Importancia De Las Tic en Aps.*

Franky, María Consuelo. *La persistencia en los nuevos frameworks.* Bogotá : s.n., 2007.

Frómata Moreno, Yoiler, Brocard Delfino, Yusniel y Suárez Corrales, Maikel. Diseño y Servicios Web para el Registro de Población de la Atención Primaria del Sistema de Información para la Salud. UCI. La Habana : s.n., 2007. Tesis.

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.

Gervás JJ, Pérez Fernández M, García Sagredo P. Normalización de la información en atención primaria: presente y futuro. Gaceta Sanitaria 1988; 2: 203-207.

Hermes. Mensajería HL7 V2.X y CDA.

Hookom, Jacob. *Inside Facelets Part 1: An Introduction.* [En línea].

Jacobson, I. Booch, G. Rumbaugh, J. *El Proceso Unificado de Desarrollo de software.* S.I.: Addison-Wesley, 2000.

Lain Entralgo, Pedro: El empirismo Clínico. En: Lain Entralgo. Pedro: Historia de la Medicina, Barcelona: Salvat (Colección: Biblioteca Médica tic Bolsillo. No 21). 1986 [309-320]

Laya, M. Oracle Press Release: El Servicio Cántabro de Salud implementa Oracle HTB en su Visor de Historia Clínica Electrónica.

Llanio Navarro, R. Propedéutica Clínica y Fisiopatología. La Habana: Ed. Pueblo y Educación, 1983:29-30.

León Garcés, Johander. Estándares de Codificación para SIAPS. 2009.

Marrero Nieblas, Ariuska y Suarez Rivero, Yoel Odelso. Modulo Banco de sangre del sistema de información hospitalaria alas HIS, 2009.

Ramos, Juan Alonso. Adictos al trabajo. [En línea].

Raquel Ochoa Ornelas. Facultad de ingeniería mecánica y eléctrica. Universidad de Colima, 2003.

Roa Ruben. Historia Clínica Orientada a Problemas, 2008

Sánchez Mansolo, Athos Alejandro. Historia Clínica Electrónica: Una herramienta para la asistencia, docencia e investigación en medicina, 1998.

Savage, P. A book that changed my practice. Problem oriented medical records. BMJ. 2001; 322 (2): 275.

Suárez Bustamante, Miguel Dr. Historia Clínica Orientada al Problema. Parte I. Fundamentos. RAMPA, 2006.

Suárez González, Héctor. 2003. javaHispano. *Manual Hibernate*. [En línea] 21 de marzo de 2003.

Starfield B. Primary care: concept, evaluation and policy. New York: Oxford University Press; 1992. Pag.404.

Tarafa Guzmán, Alejandro y Martínez Díaz, Yosmel. Módulo Citas del Sistema de Información Hospitalaria alas HIS. UCI. La Habana: s.n. 2009. Tesis.

Torralba V. Estandarización de la historia clínica electrónica [sitio en Internet]. 2004.

GLOSARIO DE TÉRMINOS

Aplicación o Sistema Informático: Programas con los cuales el usuario final interactúa a través de una interfaz y que realizan tareas útiles para éste.

Cliente Servidor: Modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

Concurrencia: Ejecución simultánea de dos o más actividades durante el mismo intervalo de tiempo.

Componente: Parte física y reemplazable de un sistema que se ajusta a, y proporciona la realización de, un conjunto de interfaces.

Deficiencia: Es toda pérdida o anomalía de una estructura o función psicológica, fisiológica o anatómica.

Dependencia: Relación semántica entre dos elementos, en la cual un cambio en uno puede afectar al otro.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología comprendidos por los practicantes de ese dominio.

Equipo Básico de Salud: Binomio conformado por el médico y enfermera de la familia, que atiende una población geográficamente determinada, que puede estar ubicado en la comunidad, centros laborales o educativos.

Interoperabilidad: Condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible. Entre las iniciativas recientes más destacadas para dotar a la Web de interoperabilidad se encuentran los servicios Web y la Web semántica.

GLOSARIO DE TÉRMINOS

Internet: Método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red (lógica) única.

Informática: Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

Memory-leak: (Fuga de memoria), es un error de software que ocurre cuando un bloque de memoria reservada no es liberado en un programa de computación. Comúnmente ocurre porque se pierden todas las referencias a esa área de memoria antes de haberse liberado.

Paquete: Mecanismo de propósito general para organizar elementos en grupos.

Policlínico: Unidad de salud donde se brindan servicios médicos a una población geográficamente determinada perteneciente al nivel asistencial de APS.

Servicio: Unidad de software que encapsula alguna funcionalidad de negocio y proporciona estas a otros servicios a través de interfaces públicas bien definidas.

Software: Conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Subsistema: Agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por los elementos contenidos.

ANEXOS

Anexo 1.

Ministerio de Salud Pública Historia Clínica	<div style="border: 1px solid black; padding: 5px; background-color: #cccccc;"> Historia Clínica Individual Atención Primaria de Salud </div>	FAMILIA _____ _____ _____
---	--	---------------------------------

DATOS GENERALES DEL PACIENTE				
1er. Apellido	2do. Apellido	Nombre (s)	Sexo F <input type="checkbox"/> M <input type="checkbox"/>	No.C. de Identidad
Color de la Piel B N M A	Educatividad Primaria <input type="checkbox"/> Secundaria Básica <input type="checkbox"/> Técnico Medio <input type="checkbox"/> Pre-Universitaria <input type="checkbox"/> Universitaria <input type="checkbox"/> Ninguno <input type="checkbox"/>			Grupo Sang. _____
				Factor RH _____

ALERGIA:

DOMICILIO

Núm. HC del Paciente		Área de Salud	FECHA		Dirección del domicilio
Consultorio	Fecha Nuc.		Alta	Baja	

ESQUEMA DE VACUNACIÓN

I.C.G.	Vacunas/dosis		Edad	Fecha	Vacunas/dosis		Edad	Fecha
	1ra.	Adm. Maternidad			1ra.	2da.		
H.B.V*	1ra.	Entre 12-24 hrs. nacido			PR5 Gdo.	React	12 meses	
	2da.	1 mes			DT 1er Gdo.	React	5-6 años	
	3ra.				AT 5to. Gdo.	1ra.	9-10 años	
	React	12 meses			AT 8to. Gdo.	2da.	9-10 años	
H.B.V**	1ra.	Entre 12-24 hrs. nacido			TT 9to Gdo.	React	13-14 años	
	2da.	1 mes			AT 11to Gdo.	React	15-16 años	
	3ra.	6 meses						
D.P.T	1ra.	2 meses			ANTI FOLIO Campaña			
	2da.	4 meses						
	3ra.	6 meses						
	React	15 meses			TT			
H.I.B	1ra.	2 meses				React		
	2da.	4 meses				React		
	3ra.	6 meses				React		
	React	15 meses				React		
A.M.B.C	1ra.	3 meses						
	2da.	5 meses						

GRUPO DISPENSARIAL:

Anexo 2.

BASE DE DATOS - Interrogatorio * N normal AN anormal					
Sexo	varón		Edad	35 años	
Estudios	universitarios		Ocupación	abogado	
Estado Civil	casado		Medicación	no	
Residencia	San Isidro		Procedencia	Nació en Junín	
Antecedentes patológicos			Antecedentes familiares		
<i>Clinicos</i>	Paperas, varicela		Cardiopatía isquémica	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
<i>Quirúrgicos</i>	apendicitis		ACV	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
<i>Traumatológicos</i>	Fractura de tobillo		DBT	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
<i>Internaciones</i>	Si, apendicitis		HTA	SI <input checked="" type="checkbox"/>	NO <input type="checkbox"/>
<i>Gineco-Obstétricos</i>	No corresponde		Abuso de alcohol / drogas	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
			Ca Mama	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
			Ca Colon	SI <input checked="" type="checkbox"/>	NO <input type="checkbox"/>
			Otros	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
Tabaco	SI <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	Protección solar	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>
Alcohol	SI <input type="checkbox"/>	NO <input type="checkbox"/>	Catarsis	N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Dieta	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>	Sueño	N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Actividad física	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>	Diuresis	N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Conduce	SI <input type="checkbox"/>	NO <input type="checkbox"/>	Actividad sexual	N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Alergias a medicamentos	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>	F. de riesgo de Sida	SI <input type="checkbox"/>	NO <input checked="" type="checkbox"/>

BASE DE DATOS - Examen físico * N normal AN anormal					
Peso	70 Kg	Talla	1,70 m	IMC	24
Ta			120		80
F. cardíaca			80		
F. respiratoria			17		
Piel		N <input checked="" type="checkbox"/>		AN <input type="checkbox"/>	
Boca		N <input checked="" type="checkbox"/>		AN <input type="checkbox"/>	
Vision		N <input type="checkbox"/>		AN <input checked="" type="checkbox"/>	
Audición		N <input checked="" type="checkbox"/>		AN <input type="checkbox"/>	
Cardiovascular				N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Respiratorio				N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Abdomen				N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
O.M. Articular				N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Neurológico				N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Genital M				N <input checked="" type="checkbox"/>	AN <input type="checkbox"/>
Genital F				N <input type="checkbox"/>	AN <input type="checkbox"/>

BASE DE DATOS - Exámenes complementarios			
Laboratorio	Radiología	Electrocardiograma	Otros
Colesterol LDL 198	Ecografía Litiasis renal	HVI	Biopsia piel, vasculitis