

# Universidad de las Ciencias Informáticas

## Facultad 7



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

Componentes web Atención Domiciliaria,  
Comunitaria y Planificación de Intervenciones de Enfermería  
para el Sistema Integral de Atención Primaria de Salud

**Autor:** José Luis Díaz Ríos

**Tutores:** Ing. Yoelvis Osés Sosa  
Ing. Yurien López Hernández

Ciudad de La Habana, Junio del 2011  
“Año del 53 de la Revolución”

## DATOS DE CONTACTO

Ing. Yoelvis Osés Sosa, Profesor Instructor, graduado en el año 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor de la Facultad # 7. Ha impartido las asignaturas Historia de la Informática, Software Libre, Seguridad Informática y Estándares Internacionales para la Gestión de Información en la Salud. Se desempeña como Asesor de Mercadotecnia del Centro de Informática Médica (CESIM). Ha presentado trabajos en UCIencia 2008, 2010 y en las Convenciones y Feria Internacional Informática 2009 y 2011, posee publicaciones en las memorias de dichos eventos. Ha sido tutor de varios trabajos de diploma.

*e-mail:* [yoses@uci.cu](mailto:yoses@uci.cu)

Ing. Yurien López Hernández - (ylhernandez@uci.cu): Graduado de Ingeniero en Ciencias Informáticas en el año 2010 con Diploma de Oro. Actualmente labora en la Universidad de Ciencias Informáticas (UCI), desempeñándose como profesor de la Facultad No. 7 vinculado a la producción. Actualmente es parte del proyecto Sistema Integral para la Atención Primaria de Salud (SIAPS) donde se desarrolla como Jefe del módulo Enfermería. Presentó un trabajo en la XIV Convención y Feria Internacional Informática 2011.

*e-mail:* [ylhernandez@uci.cu](mailto:ylhernandez@uci.cu)

### RESUMEN

Dentro de la Atención Primaria de Salud, el personal de enfermería es responsable de garantizar una atención eficiente a la salud de la población; dos de sus principales funciones las conforman las atenciones domiciliarias y comunitarias. A pesar de la importancia de la información que se genera en estas actividades, actualmente estos procesos no se gestionan de una forma correcta. Procesar toda la información generada en dicha área consume mucho tiempo y esfuerzo. Por estas razones se propone el desarrollo de los Componentes web: Atención Domiciliaria y Comunitaria, para el Sistema Integral para la Atención Primaria de Salud; que faciliten la gestión de la información en estas actividades.

El sistema está basado en tecnologías libres y multiplataforma. Se utiliza Java como lenguaje de programación y se implementa el patrón de arquitectura Modelo Vista Controlador. Se emplea la Plataforma Java Enterprise Edition 5.0, PostgreSQL como Sistema Gestor de Bases de Datos, Hibernate como herramienta para la persistencia de los datos y el framework integrador Seam.

Con los Componentes web: Atención Domiciliaria y Comunitaria, del Módulo de Enfermería del SIAPS se espera aumentar la calidad de la atención brindada a los pacientes. Además debe contribuir a mejorar los procesos de atención a la comunidad y a los pacientes hospitalizados en los hogares; permitiendo que la información que estos generan, sea actualizada y accesible en tiempo real.

**PALABRAS CLAVE:** Atención Comunitaria, Atención Domiciliaria, Atención Primaria de Salud, Componente web, Enfermería.

# TABLA DE CONTENIDO

Introducción.....	1
Capítulo 1.Fundamentación Teórica.....	6
1.1 Conceptos básicos relacionados con el dominio del problema.....	6
1.2 Análisis de las soluciones existentes.....	8
1.3 Tendencias y tecnologías actuales.....	10
1.4 Herramientas Informáticas.....	17
Capítulo 2. Características del Sistema.....	19
2.1 Modelo del Negocio.....	19
2.2 Diagramas de procesos de negocio.....	19
2.3 Propuesta del Sistema.....	25
Capítulo 3. Diseño del Sistema.....	30
3.1 Objetivos del Flujo de Trabajo Diseño.....	30
3.2 Patrones de diseño.....	31
3.3 Diagramas de Clases del Diseño.....	32
3.4 Descripciones comunes.....	35
Capítulo 4. Implementación.....	52
4.1 Patrones de arquitectura.....	52
4.2 Propuesta de integración entre módulos.....	54
4.3 Diagrama de Despliegue.....	55
4.4 Tratamiento de excepciones.....	58
4.5 Estrategias de codificación. Estándares y estilos a utilizar.....	58
Conclusiones.....	63
Recomendaciones.....	64

# TABLA DE CONTENIDO

Referencias Bibliográficas.....	65
Bibliografía.....	67
Glosario de Términos.....	70
Anexos.....	71

## Introducción

El desarrollo vertiginoso de las Tecnologías de la Información y las Comunicaciones (TIC) en estos últimos años ha tenido una alta incidencia en la modernización y eficiencia de todos los sectores de la sociedad. El constante progreso y la gran aceptación de estas tecnologías, se debe a los múltiples y convenientes beneficios que estas brindan. La automatización de tareas es un ejemplo fehaciente de ello, esta permite agilizar y mejorar considerablemente los procesos y actividades que se realizan en una institución. Permitiendo reducir costos, así como incrementar en gran medida su eficiencia.

La incorporación de las TIC en el ámbito institucional puede resultar un elemento clave para mejorar la competitividad e impulsar el desarrollo económico. Cuba no ha estado alejada de estos avances, ha convertido estas tecnologías en un integrante fundamental del concepto moderno de la sociedad.

La salud ha sido una de las esferas donde el inevitable auge de las TIC ha estado presente en cada uno de sus actuales procesos. La necesidad de mantener y mejorar los servicios sanitarios generales y una atención médica integral han hecho de estas tecnologías un escalón fundamental en el desarrollo mundial de esta rama. El sector de la salud ha transitado por un importante proceso de desarrollo durante estos 53 años de Revolución, los servicios antes del triunfo revolucionario eran prácticamente inalcanzables para los trabajadores y campesinos; o sea, solo los magnates políticos y adinerados tenían a su alcance una buena atención médica.

Desde los primeros años de la Revolución se comenzaron a ver los cambios y el desarrollo en esta esfera. El cual se pudiera ejemplificar en cinco etapas: [1]

1. De integración del Sistema Nacional de Salud Único, que comprende 1959 y años de la década de 1960.
2. De consolidación del Sistema Nacional de Salud Único, años de la década de 1970.
3. De incorporación y desarrollo de alta tecnología médica, impulso a las investigaciones y dispensarización de acciones de salud a toda la población del país a través del médico de la familia, años de la década de 1980.
4. De lucha por la preservación de las conquistas logradas en el campo de la salud pública y por su desarrollo en condiciones de crisis económica profunda, años de la década de 1990.

## 5. De desarrollo de la automatización en la salud, años de la década del 2000.

El desarrollo que ha experimentado la Salud Pública en Cuba es expresión del desarrollo económico social alcanzado y de la alta prioridad que la Revolución, el pueblo, le han otorgado a la salud y el bienestar del hombre.

“Se han ido acumulando logros en el mejoramiento del estado de salud de la población, comparable al de países de alto nivel de desarrollo económico social, con una distribución justa y equitativa tendente a borrar las diferencias, entre regiones y grupos humanos, heredadas del capitalismo.” [2]

El Sistema Nacional de Salud se encuentra estructurado en 3 niveles de atención médica: atención primaria, atención secundaria y atención terciaria. De las cuales la atención primaria es la más importante pues da solución al 80 % de los problemas de salud dejando solo un 20% para los otros dos niveles.

La Atención Primaria de Salud (APS) constituye la asistencia sanitaria esencial accesible a todos los individuos y familias de la comunidad a través de medios aceptables para ellos, con su plena participación y a un costo asequible para la comunidad y el país. Es el núcleo del sistema de salud y forma parte integral del desarrollo socioeconómico general de la comunidad.

Precisamente dentro de la Atención Primaria un eslabón fundamental es la labor de las enfermeras, estas tienen un papel relevante en el equipo de atención médica a este nivel, con la puesta en marcha de iniciativas, experiencias novedosas en la prevención y en la promoción del auto cuidado, además de las actividades habituales en el centro, el domicilio y la comunidad. [3]

Las enfermeras son el grupo principal del personal de salud que dispensa la APS. Fomentan y mantienen los vínculos entre las personas, las familias, las comunidades y el resto del sistema de atención de salud, trabajando de manera autónoma y en colaboración para prevenir las enfermedades; además promover, mejorar, mantener y restaurar la salud. Su trabajo abarca la salud de la población, la promoción de la salud, la prevención de las enfermedades y los cuidados de bienestar además de ser el primer punto de contacto para la gestión de la enfermedad. [4]

Una de las actividades más importantes que realizan las enfermeras es la asistencia domiciliaria, esta es una alternativa a la hospitalización convencional en pacientes seleccionados, con objetivos primarios como: disminuir el tiempo de hospitalización, eliminar riesgos inherentes al ingreso en un hospital, mejorar la intimidad y el sentimiento de seguridad del paciente, favorecer la personalización y humanización de los servicios, evitar desplazamientos a la familia, agilizar la inserción en la vida activa y favorecer la educación

sanitaria, su actuación se orienta a pacientes con patología aguda o crónica agudizada que sin este recurso asistencial tendrían que permanecer ingresados en un centro sanitario.

Otra de las actividades más importantes que realizan las enfermeras la constituye la Atención Comunitaria, es aquella que tiene como objetivo la atención a una población, de una comunidad definida y que busca resultados en salud. Requiere de la intervención sobre los factores determinantes de la salud y la enfermedad, para lo que es necesario identificar las necesidades en salud en la población y los factores que las determinan.

La Atención Domiciliaria y Comunitaria de las enfermeras en la APS constituye un elemento de vital importancia para los pacientes y para la comunidad en general ya que garantiza que los servicios médicos lleguen hasta todos los lugares de la comunidad. Las enfermeras se convierten en el centro de la prestación de cuidados en la APS, garantizando un mejor acceso a estos, con lo que se logra una mayor eficiencia en los costos y una mejor aceptación de los servicios de enfermería por parte del paciente.

Según estadísticas del Consejo Internacional de Enfermería (CIE), entre el 60 y el 80 % de la APS prestada tradicionalmente por médicos, puede ser dispensada por enfermeras a un costo menor y resultados similares. [5]

Teniendo en cuenta lo anteriormente expuesto se identifica la siguiente **situación problemática**:

En la actualidad los servicios brindados por las enfermeras se encuentran en todas las áreas de la APS, atienden cada día un número muy elevado de pacientes lo que significa que cada día aumenta considerablemente el flujo de información que ellas manipulan. Dicha información es recopilada engorrosamente de forma manual y en papel, demandando como es lógico una porción muy significativa de tiempo, lo que implica que las enfermeras empleen menos tiempo en la atención directa a los pacientes. Esta situación incide además en la calidad de la información recopilada, dificultando su efectividad en el seguimiento a los pacientes.

La programación improvisada, inestable y manual de las intervenciones de enfermería tales como: promoción del autocuidado, actividades preventivas, seguimiento y control de problemas de afrontamiento, adaptación, los procedimientos técnicos básicos, actividades sociales para la comunidad. Así como la planificación de visitas a hospitalizados en el hogar; de vital importancia para la prevención, promoción y restablecimiento de la salud en los pacientes y la sociedad, incide de forma negativa en la eficiencia de la atención brindada por las enfermeras.

Como en muchas ocasiones la información no es almacenada en lugares seguros y además por ser guardada en formato duro, muchas veces se pierden datos importantes sobre actividades comunitarias como charlas educativas y talleres preventivos, las cuales deben seguir un sentido lógico y constante para un mejor resultado en la comunidad. Además dicha información, se utiliza en otros procesos como la realización de seguimientos y evaluaciones del desempeño y el impacto de las acciones de educación de salud en la comunidad.

Por no existir una correcta estandarización de los datos y además no contar con una eficiente Planificación de Intervenciones, informaciones de vital importancia como las que se recopilan en el seguimiento de tratamientos médicos y en la continua atención a los ingresados en los hogares, no logran una total veracidad. Además muchas informaciones son archivadas con muy poca claridad, lo que constituye un importante obstáculo para un mejor resultado en la Atención Domiciliaria.

Estas constantes limitaciones en la gestión de información a las que se enfrentan las enfermeras diariamente dificultan el desempeño de sus responsabilidades y ponen en riesgo la atención eficiente a la salud del paciente y de la localidad.

Después de analizar la situación antes expuesta, se plantea como **problema a resolver** la siguiente interrogante: ¿Cómo viabilizar la gestión de la información en las actividades de Atención Domiciliaria y Comunitaria, así como la Planificación de Intervenciones de Enfermería en la Atención Primaria de Salud?

El **objeto de estudio** del presente trabajo se refiere al proceso de gestión de información en las actividades de enfermería en la Atención Primaria de Salud, donde el **campo de acción** se centra en el proceso de gestión de la información de las actividades de Atención Domiciliaria y Comunitaria, así como la Planificación de Intervenciones de Enfermería en la Atención Primaria de Salud.

Para dar solución al problema planteado se propuso como **objetivo general**: Desarrollar los Componentes web Atención Domiciliaria, Comunitaria, y Planificación de Intervenciones de Enfermería, que facilite la gestión de la información de estos servicios en la Atención Primaria de Salud.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de la investigación**:

- I. Realizar un análisis de los procesos relacionados con la Atención Domiciliaria y Atención Comunitaria en la APS.

- II. Realizar un estudio bibliográfico de las tendencias actuales nacionales e internacionales relacionadas con los sistemas de gestión de la atención de enfermería en la APS.
- III. Aplicar la arquitectura definida en el Centro de Informática Médica (CESIM) para aplicaciones web.
- IV. Elaborar mediante la guía de desarrollo propuesta por el proceso de mejora los artefactos correspondientes a las fases: “Modelado del Negocio”, “Gestión de Requerimientos”, “Diseño e Implementación” de los procesos de Atención Domiciliaria, Atención Comunitaria y Planificación de Intervenciones de Enfermería en la APS.
- V. Implementar los procesos de Atención Domiciliaria, Comunitaria y Planificación de Intervenciones de Enfermería en la APS.

El documento se encuentra estructurado por cuatro capítulos, los cuales abordan todo lo relacionado con el presente trabajo investigativo.

**CAPÍTULO 1. Fundamentación teórica:** contempla los conceptos relacionados con el dominio de la investigación, además el resultado del estudio detallado del estado del arte en cuanto a tecnologías, metodologías y herramientas utilizadas para el desarrollo, y el análisis de sistemas informáticos existentes en el ámbito nacional e internacional que gestionan información relacionada con los procesos de atención de Enfermería en la APS .

**CAPÍTULO 2. Características del sistema:** en este capítulo se describen detalladamente los procesos, además se aborda todo lo referente al modelado de negocio y se especifican los requisitos funcionales y no funcionales de la aplicación a desarrollar.

**CAPÍTULO 3. Diseño del sistema:** en este capítulo se realiza el diseño de la solución propuesta, se describe la concepción arquitectónica del Sistema Integral para la Atención Primaria de Salud (las SIAPS) y su aplicación en los componentes Atención Domiciliaria, Comunitaria y en el de Planificación de Intervenciones de las enfermeras del Módulo de Enfermería. Además se definen los distintos patrones a utilizar en el diseño del sistema.

**CAPÍTULO 4. Implementación:** en este capítulo se tratarán los principales aspectos relacionados con la implementación del sistema. Se fundamenta la necesidad de integración del Módulo de Enfermería Web con otros sistemas del SIAPS. Se definen también elementos clave para el tratamiento de errores y los estilos de código empleados en el desarrollo de la aplicación.

## Capítulo 1. Fundamentación Teórica

Contempla los principales conceptos relacionados con el dominio de la investigación. Se realiza un estudio detallado del estado del arte actual en cuanto a todo tipo de tecnologías, metodologías y herramientas utilizadas para el desarrollo, así como una panorámica general de sistemas existentes.

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

#### Salud y Sistema Nacional de Salud (SNS).

La Organización Mundial de la Salud (OMS) define a la salud como “el estado de completo bienestar físico, mental y social. Es decir, el concepto de salud trasciende a la ausencia de enfermedades y afecciones en un organismo.” [6]

Se conoce como un Sistema Nacional de Salud a todas aquellas fórmulas y métodos utilizados por un país para organizar y garantizar la atención médica básica necesaria, conceptualizado según la Organización Mundial de la Salud (OMS) como: “*un sistema que engloba todas las organizaciones, instituciones y recursos cuyo principal objetivo es llevar a cabo actividades encaminadas a mejorar la salud del país.*” [7]

Dentro de los objetivos principales del SNS se encuentran:

- Mejorar la salud de la población.
- Prestar servicios de salud con calidad, eficiencia y seguridad.
- Evitar el empobrecimiento de la población por motivos de salud.
- Acceso equitativo.
- Entornos saludables.

De acuerdo a la complejidad de las acciones curativas, preventivas y de rehabilitación, así como la especialidad de cada servicio, la atención de salud está organizada en tres niveles en estrecha relación. Atención Terciara de Salud (ATS), Atención Secundaria de Salud (ASS) y el eslabón más importante el nivel de Atención Primaria de la Salud.

## **Atención Primaria de Salud (APS).**

La APS es un enfoque que constituye la base y determina la manera de trabajar de los restantes niveles del sistema sanitario. Da solución aproximadamente al 80 % de los problemas de salud de la población y que correspondan con las acciones de promoción y protección de la salud.

La OMS la conceptualizó como: “la asistencia sanitaria esencial basada en métodos y tecnologías prácticos, científicamente fundados y socialmente aceptables, puesta al alcance de todos los individuos y familias de la comunidad mediante su plena participación y a un costo que la comunidad y el país puedan soportar, en todas y cada una de las etapas de su desarrollo con un espíritu de autorresponsabilidad y autodeterminación. La Atención Primaria forma parte integrante tanto del sistema nacional de salud, del que constituye la función central y el núcleo principal, como del desarrollo social y económico global de la comunidad y el país.” [8]

Algunos elementos conceptuales de la Atención Primaria de la Salud son los siguientes:

- Integral.
- Integrada.
- Continuada y permanente.
- Activa.
- Accesible.
- Basada en el trabajo en equipo.
- Comunitaria y participativa.
- Programada y evaluable.
- Docente e investigadora.

## **Enfermería.**

La Enfermería es la ciencia que estudia los cuidados, autónomos y en colaboración, que se prestan a las personas de todas las edades, familias, grupos y comunidades, enfermos o sanos, en todos los contextos, e incluye la promoción de la salud, la prevención de la enfermedad, y los cuidados de los enfermos y discapacitados.

Las funciones esenciales de la enfermería son la defensa, el fomento de un entorno seguro, la investigación, la participación en la política de salud y en la gestión de los pacientes y los sistemas de salud.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

El personal de enfermería es de gran utilidad para el servicio de salud general, por ser la persona que entra en contacto más directo con el paciente; sus observaciones y permanente cuidado del mismo le permite dar un informe completo y exhaustivo al médico tratante. Las enfermeras por lo general se encargan de cuestiones relacionadas con el confort del paciente pero también con sus datos y análisis más directos como por ejemplo presión en sangre, nivel de azúcar, pulsaciones, y una evolución general de la persona desde la última vez que la vio el médico. [9]

## **Atención Comunitaria.**

Constituye el conjunto de estrategias, métodos y actividades, cuyo objetivo principal es mantener y mejorar el estado de salud de una población en una comunidad. Esta requiere la intervención sobre todos los factores que puedan influir determinantemente en la salud y en cualquier tipo de enfermedad. Siendo conocido que muchos de los problemas de salud tienen un origen multicausal, una de las características imprescindibles de la Atención Comunitaria es la necesaria intervención multisectorial y no sólo desde los servicios sanitarios, así como un continuo y minucioso seguimiento de los problemas específicos de cada comunidad. [10]

## **Atención Domiciliaria.**

Conjunto de actividades de carácter social y sanitario que se prestan en el domicilio. Esta atención permite detectar, valorar, apoyar y controlar los problemas de salud del individuo y las familias potenciando la autonomía y mejorando la calidad de vida de las personas. Su relevancia es cada vez mayor, debido al aumento progresivo de población envejecida, así como la alta supervivencia de gran número de enfermedades incapacitantes. [11]

## **1.2 Análisis de las soluciones existentes.**

En la actualidad las nuevas tecnologías han revolucionado el mundo y a su vez de una forma u otra todos los sectores la sociedad. Las TIC tienen la capacidad de catalizar los procesos de negocio, haciendo que estos sean cada vez más rápidos y precisos. Las estrategias tecnológicas digitales son aplicables a todas, o casi todas, las instituciones públicas, incluyendo como es lógico todas las especialidades de la atención médica moderna; como consecuencia de esto se ha desatado un gran mercado competitivo de aplicaciones informáticas. A continuación se analizan algunos sistemas enfocados a la APS y dentro de esta específicamente a la enfermería, cada uno de estos dirigidos a las necesidades particulares de la institución para la cual se hayan implementado.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Existe una solución española llamada **LogicPAE** que cuenta con un generador de inteligencia artificial incorporado que permite gestionar completamente el proceso de atención de enfermería de las distintas unidades o servicios que disponga un centro de salud.

LogicPAE es un buen sistema monoplataforma, que solo soporta el sistema operativo Windows y en las versiones XP o superiores, es un sistema escritorio desarrollado sobre la plataforma .NET, que cuenta con tres versiones (Estudiantes, Profesional y Empresarial).

Este sistema no es de fácil adquisición pues para poder obtener un paquete con todas las funcionalidades de más importancia es necesario pagar varias tarifas. Además no presenta ningún tipo de soporte y al igual que para obtenerlo para una buena preparación y capacitación de usuarios, es necesario realizar un pago relativamente alto que puede variar en dependencia del tiempo y la lejanía el lugar donde se realice.

Existe también otra solución española desarrollada en Sevilla, nombrada **MOVICARE**, una aplicación que lleva un periodo de funcionamiento de dos años en varios hospitales andaluces, entre los que se encuentra el Hospital Universitario de Puerto Real y los hospitales de la orden de San Juan de Dios.

Desarrollado por la compañía ICX Sistemas S.A., dedicada al desarrollo de software de aplicación e integración de sistemas en la sanidad, es un producto enfocado a la gestión de la enfermería hospitalaria, presentado como principal ventaja que aborda la enfermería en doble perspectiva por un lado, enfocado a paciente, todo el proceso enfermero, pero por otro lado este enfoque cohabita con el clásico reparto de tareas por turno.

El sistema convive con las estaciones de trabajo tradicionales, ordenadores de mesa, con estaciones móviles, PDA'S para la recogida de datos a pie de cama como pueden ser constantes, anotaciones de evolución o incidencias. [13]

En el ámbito nacional han sido pocos los antecedentes, como aplicación informática enfocada en la enfermería solamente encontramos el software de Proceso de Atención de Enfermería (PAE), implementado por un equipo de desarrollo del CIREN (Centro Internacional de Restauración Neurológica), el cual es utilizado en algunas versiones en varias instituciones del país, como por ejemplo en el Hermanos Ameijeiras y además se utiliza en el centro donde fue desarrollado, en este último en su versión completa.

Otros antecedentes nacionales no han llegado a un producto final, aunque se ha trabajado con bastante intensidad en una aplicación de Esterilización y en otra de Gerencia, desarrollados en Delphi, utilizando

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

SQL Server como gestor de base de datos. Además en la nación se utilizó en la década del 2000 una aplicación desarrollada por el Grupo Nacional de Informática de Enfermería (GINENF) en el procesador Excel, la cual era enviada por correo electrónico diariamente lo que facilitó la estandarización de los datos, así como el procesamiento y almacenamiento de informaciones importantes de este sector.

En Cuba los sistemas antes mencionados no cubren las necesidades de todo el personal de enfermería debido a que la información que gestionan no está acorde al flujo actual de datos que se procesan en los departamentos de enfermería en la APS. Además, estos fueron implementados para una institución específica lo que le imposibilita extenderse a todos los servicios de enfermería en la APS.

En el ámbito internacional los dos sistemas constituyen muy buenas soluciones que poseen potentes funcionalidades, aunque presentan importantes desventajas como que son aplicaciones de escritorio, es decir se complica el proceso de despliegue ya que se torna un poco engorrosa la instalación del software en cada estación de trabajo, y por ser productos privativos hay que pagar por cada estación de trabajo desplegada. Además de que son soluciones monoplataformas que solo pueden ser instaladas sobre un sistema operativo específico. Vale destacar también la falta de compatibilidad e intercambio de información con otros sistemas, lo que impide que el cúmulo de información que manipulan tribute a otros niveles de atención médica.

Ninguno de estos productos cumple con los requisitos de integración plasmados en la estrategia de informatización de la salud en Cuba. Además, les faltan algunas funcionalidades de suma importancia en este sector, como la gestión de las atenciones comunitarias, domiciliarias y el proceso de planificación de horarios para las enfermeras; pues estas soluciones son implementadas para centro y sistemas específicos.

## **1.3 Tendencias y tecnologías actuales.**

El mundo de la informática desde sus inicios se caracterizó por un constante y muy rápido desarrollo, desde aquellos grandes ordenadores que permitían realizar algunas tareas en forma algo limitada y de un uso exclusivo de organizaciones muy selectas, hasta los de la actualidad, los que tienen mayores capacidades y propiedades que los primeros, además de estar cada vez más introducidos en el quehacer cotidiano y en todo los sectores de la sociedad moderna. Esta evolución de los sistemas informáticos ha dado lugar a la creación de numerosas herramientas que ayudan a organizar, diseñar y construir una aplicación informática, permitiendo minimizar el esfuerzo del equipo de desarrollo. Por lo que para la

realización de un producto informático actualmente se hace necesario llevar a cabo un estudio detallado de todas las tecnologías y herramientas que serán utilizadas así como sus ventajas y desventajas de sus servicios.

## 1.3.1 Arquitectura cliente-servidor.

Consiste básicamente en uno o varios clientes que realizan peticiones a un servidor y este le da respuesta. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

Ventajas de esta arquitectura:

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

## 1.3.2 Arquitectura en capas.

El objetivo principal de esta arquitectura es la separación de la lógica de negocios de la lógica de diseño. Su ventaja principal radica en que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo

de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API (de las siglas Application Program Interface) que existe entre niveles. [14]

El diseño más utilizado actualmente es el diseño en tres capas, consiste en:

**Capa de presentación:** es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario.

**Capa de negocio:** es donde residen los programas que se ejecutan. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

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

### 1.3.3 Tecnologías utilizadas en el proceso de desarrollo.

#### Java.

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. [15]

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación a código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible. [16]

#### Frameworks.

Los Frameworks se pueden considerar como soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de

ejecución). Aceleran el proceso de desarrollo, permiten reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

## **Java Server Faces (JSF).**

Es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. [17]

JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Dos librerías de etiquetas personalizadas para Java Server Pages que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Beans administrados.

## **RichFaces.**

Es una biblioteca de componentes para JSF y un avanzado framework para la integración de AJAX con facilidad en la capacidad de desarrollo de aplicaciones de negocio. Los componentes RichFaces vienen listos para su uso sin tener que hacer muchas configuraciones, por lo que los desarrolladores pueden ahorrar tiempo de inmediato para aprovechar las características de los componentes para crear aplicaciones web que proporcionan mejoras en gran medida en la experiencia del usuario más fiable. También incluye un fuerte apoyo para el soporte de temas de aplicaciones JSF y aprovecha al máximo los beneficios de JSF framework incluyendo, la validación y conversión junto con la gestión de estática y dinámica los recursos. [18]

## **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

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

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. [19]

## **Facelets.**

JavaServer Facelets es un framework para plantillas centrado en la tecnología JSF (JavaServer Faces), por lo cual se integran de manera muy fácil. Facelets llena el vacío entre JSP y JSF, siendo una tecnología centrada en crear árboles de componentes y estar relacionado con el complejo ciclo de vida JSF. [20]

## **Jboss Seam.**

Es un framework desarrollado por JBoss, una división de Red Hat. Combina a dos frameworks: Enterprise JavaBeans (EJB) y JavaServerFaces (JSF). Gracias a él puede accederse a cualquier componente EJB desde la capa de presentación refiriéndose a él mediante su nombre de componente seam.

Seam introduce el concepto de contextos. Cada componente de Seam existe dentro de un contexto. El contexto conversacional por ejemplo captura todas las acciones del usuario hasta que éste sale del sistema o cierra el navegador - inclusive puede llevar un control de múltiples pestañas y mantiene un comportamiento consistente cuando se usa el botón de regresar del navegador. [21]

## **Hibernate.**

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

Busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen.

Con esta información Hibernate le permite a la aplicación manipular los datos de la base de datos operando sobre objetos, con todas las características de la programación orientada a objetos (POO). Convierte los datos entre los tipos utilizados por Java y los definidos por SQL (structured query language) ,

genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución. 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.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente. [22]

## **Metodología de Desarrollo de Software.**

Una metodología de software tiene como objetivo reducir costos, retrasos de proyectos e indiscutiblemente mejorar la calidad del producto. La metodología de desarrollo cobra gran importancia en todos los proyectos en especial en los empresariales, pues al no utilizarla adecuadamente se puede desembocar en la frustración del equipo de desarrollo, así como en la insatisfacción de los clientes por lo que su uso es necesario para mejorar y controlar el ciclo de vida de un proyecto.

## **Lenguaje Unificado de Modelado (UML).**

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.

### **1.3.4 Patrón de arquitectura.**

#### **Modelo Vista Controlador.**

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones web. Los elementos de este patrón son:

**Modelo:** esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

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

**Controlador:** este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

## 1.3.5 Servidor de Aplicaciones (JBoss Application Server).

Es el servidor de aplicaciones de código abierto más desarrollado del mercado. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia. Puede ser instalado sobre varios ambientes, tales como Windows o GNU/Linux.

Las características destacadas de JBoss incluyen:

- Producto de licencia de código abierto sin coste adicional.
- Cumple los estándares.
- Confiable a nivel de empresa
- Incrustable, orientado a arquitectura de servicios.
- Flexibilidad consistente.
- Servicios del middleware para cualquier objeto de Java.

## 1.3.6 Sistema Gestor de Base de Datos.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Berkeley Software Distribution) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. [23]

## 1.3.7 Metodologías de Desarrollo de Software.

**CMMI** (Capability Maturity Model Integration) es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software. Con su instauración se espera alcanzar beneficios como:

- Calendarios y presupuestos predecibles en los proyectos.

- Mejora del ciclo de vida dentro del desarrollo de software.
- Mayor productividad.
- Mayor calidad de los productos y servicios que ofrece la universidad a sus clientes y por ende la satisfacción de los mismos.
- Mejorar la moral del personal que labora en el centro.

## 1.3.8 Proceso Unificado de Desarrollo de Software (RUP).

Es un Proceso Unificado de Desarrollo y junto con el Lenguaje de Modelado Unificado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Sus principales características se centran en: implementar las mejores prácticas en ingeniería de software, disciplinar la forma de asignar tareas y responsabilidades, quién hace qué, cuándo y cómo, administrar requisitos, usar arquitectura basada en componentes y controlar cambios y modelado visual del software.

RUP (Rational Unified Process) posee tres características fundamentales: su desarrollo es iterativo e incremental por lo que divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo; es guiado por los casos de uso, un caso de uso describe un fragmento de las funcionalidades del sistema que proporciona al usuario un resultado importante, los casos de uso guían el diseño construcción y prueba del sistema, esto significa que guían el proceso de desarrollo; RUP está centrada en la arquitectura, lo que le permite a los desarrolladores una mayor visibilidad del sistema, pues la arquitectura es una vista del diseño completo del software con las características más importantes resaltadas, dejando a un lado los detalles. Utiliza como lenguaje de modelado UML.

## 1.4 Herramientas Informáticas.

Se utilizó el Visual Paradigm como herramienta CASE (Computer Aided Software Engineering) basada en UML que permite crear los diagramas y artefactos que se generen como parte de la documentación; este soporta los principales estándares de la industria tales como UML, SysML, BPMN, XMI, además ofrece un conjunto de herramientas a los equipos de desarrollo de software, necesarias para los requisitos de la captura, software de planificación, la planificación de controles, el modelado de clases y el modelado de datos entre otros.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Para la programación y desarrollo del sistema, el entorno de desarrollo integrado Eclipse que cuenta con facilidades de refactorización, completamiento de código y es una herramienta multiplataforma que permite la integración con Hibernate. Además provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, así como también posee una atractiva interfaz que lo hace fácil y agradable de usar.

Como cliente para el gestor de base de datos el PostgreSQL, maestro por las bondades visuales que presenta y la facilidad de su manejo. Además, se utiliza el pgAdmin III para la administración de la base de datos, que es una herramienta de código abierto que tiene una interfase gráfica que soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD y Mac OSX.

En este capítulo se ha realizado un estudio de los principales sistemas existentes en el área de Enfermería de la Atención Primaria de la Salud arribando a la conclusión que los sistemas estudiados no satisfacen las necesidades que existen en esta área. Además se realizó una valoración de las principales tecnologías, metodologías y herramientas que son empleadas en el desarrollo de la aplicación necesaria para una mejor comprensión de la presente investigación.

### Capítulo 2. Características del Sistema.

En este capítulo se describe detalladamente el flujo de trabajo del negocio. Se especifican los requisitos funcionales y no funcionales de la aplicación a desarrollar, así como se detalla ampliamente todos los procesos involucrados.

#### 2.1 Modelo del Negocio.

El modelo de negocio de una empresa es una representación simplificada de la lógica de negocio. Describe lo que un negocio ofrece a sus clientes, cómo llega a ellos, y cómo se relaciona con ellos.

Comprende en general el conjunto de las siguientes cuestiones:

- Cómo seleccionará sus clientes.
- Cómo define y diferencia sus ofertas de producto.
- Cómo crea utilidad para sus clientes.
- Cómo consigue y conserva a los clientes.
- Cómo sale al mercado (estrategia de publicidad y distribución).
- Cómo define las tareas que deben llevarse a cabo.
- Cómo configura sus recursos.
- Cómo consigue el beneficio.

#### 2.2 Diagramas de procesos de negocio.

##### 2.2.1 Realizar Seguimiento a Ingresos Domiciliarios.

Una de las actividades de la enfermera que trabaja junto al médico en los consultorios de familia, es el chequeo y evolución de los pacientes ingresados en su hogar. Para hacerlo satisfactoriamente se siguen los siguientes pasos:

La enfermera procede a visitar los ingresos en el hogar planificados en el registro Planificación de Intervenciones de Enfermería. Una vez tomada la conducta del paciente procede a cumplir con el tratamiento médico. Luego deja por escrito la evolución del mismo y actualiza su HCI. La enfermera realiza el seguimiento de los tratamientos, tomando la conducta del paciente en cada visita al hogar. Ver

*Fig. 1.*

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

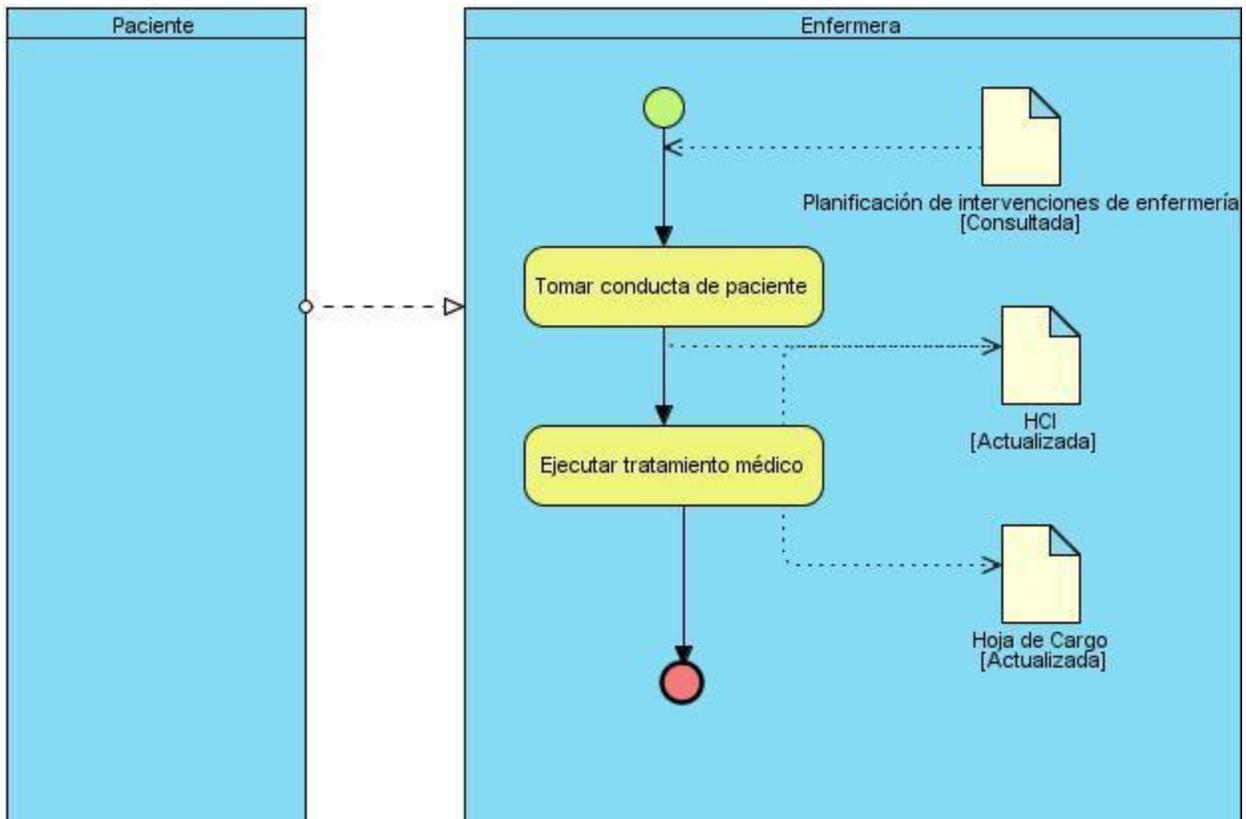


Fig 1: Diagrama de Realizar Seguimiento a Ingresos Domiciliarios.

A continuación se presenta una descripción del proceso Realizar Seguimiento a Ingresos Domiciliarios. Ver tabla 1.

Tabla 1: Descripción del proceso Realizar Seguimiento a Ingresos Domiciliarios.

Nombre:	Realizar seguimiento a ingresos domiciliarios.
Objetivos:	Evolucionar a los pacientes ingresados en su hogar.
Evento(s) que lo generan:	Ingresos en el hogar cuyos pacientes están incapacitados de caminar o estén en reposo absoluto.
Precondiciones:	Planificación de terrenos.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Poscondiciones:	Actualización de la HCI. Actualización de la Hoja de Cargo.
Reglas de Negocio:	Regla de negocio (Regla Textual # 8) Ver documento APS_SIAPS_0116_RNeg_Enf_W.doc. Regla de negocio (Reglas de Relación # 5, 8) Ver documento APS_SIAPS_0116_RNeg_Enf_W.doc.
Responsables:	Enfermera.
Clientes internos:	Enfermera.
Clientes externos:	Paciente.
Entradas:	Planificación de Intervenciones de Enfermería. HCI.
Salidas:	Hoja de Cargo. HCI.
Actividades:	Realizar el interrogatorio. Tomar conducta al paciente. Cumplir tratamiento médico.

### 2.2.2 Planificar Horario y Afectaciones de Enfermería.

Los médicos de consultorios de tipo 1 o las enfermeras jefas de los consultorios de tipo 2, son las que deben realizar la planificación de las afectaciones de trabajos y luego las enfermeras teniendo en cuenta estas afectaciones se planifican las intervenciones que tienen que realizar en un periodo determinado. Ver Fig. 2.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

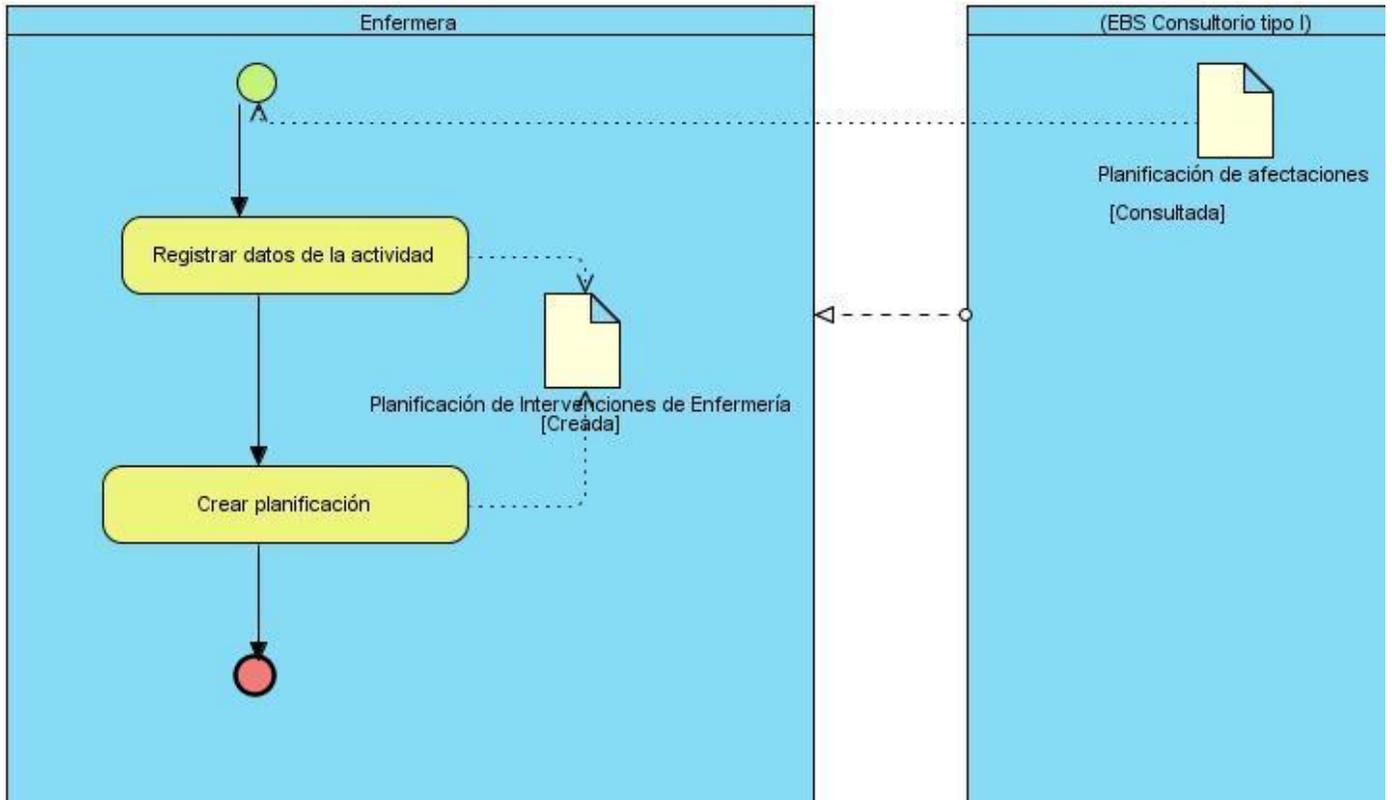


Fig 2: Diagrama de Planificar Horario y Afectaciones de Enfermería.

A continuación se presenta una descripción del proceso Planificar Horario y Afectaciones de Enfermería. Ver tabla 2.

Tabla 2: Descripción del proceso Planificar Horario y Afectaciones de Enfermería.

Nombre:	Gestionar Horario y Afectaciones de Enfermería.
Objetivos:	Realizar las planificaciones de todas las actividades de enfermería.
Evento(s) que lo generan:	Planificación de terrenos.
Precondiciones:	El médico que labora con la enfermera planifica los terrenos que ambos deben hacer.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Poscondiciones:	Registro de todas las actividades a realizar por la enfermera en un rango de tiempo determinado.
Reglas de Negocio:	Regla de negocio (Reglas textuales # 1, 2, 8, 9, 10, 11) Ver documento APS_SIAPS_0116_RNeg_Enf_W.doc.  Regla de negocio (Reglas de Relación #4, 5,6, 7, 8, 9, 10, 13) Ver documento APS_SIAPS_0116_RNeg_Enf_W.doc.
Responsables:	Enfermera
Clientes internos:	Paciente
Clientes externos:	CESIM
Entradas:	Planificación de afectaciones
Salidas:	Planificación de Intervenciones de Enfermería
Actividades:	<ol style="list-style-type: none"><li>1. Registrar datos de la actividad</li><li>2. Crear planificación</li></ol>

### 2.2.3 Realizar Actividad Comunitaria.

El personal de enfermería debe entre otras responsabilidades comunitarias, realizar actividades educativas en la comunidad para corroborar con la prevención de enfermedades e infecciones. Para ello incluyen en su planificación las Actividades Comunitarias. La realización de las mismas tiene lugar en la comunidad o en el consultorio médico conjuntamente con el grupo de pacientes que la misma cite. Ver Fig. 3.

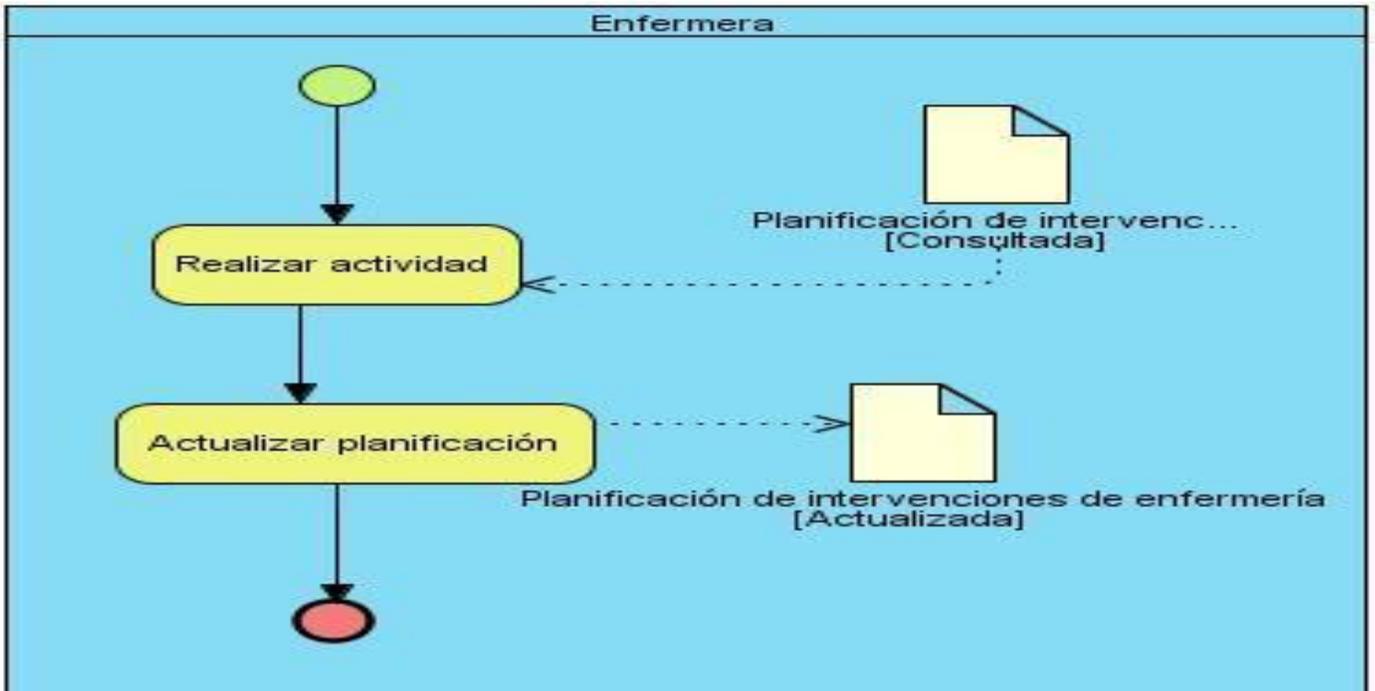


Fig 3: Diagrama de Realizar Actividad Comunitaria.

A continuación se presenta una descripción del proceso Realizar Actividad Comunitaria. Ver tabla 3.

Tabla 3: Descripción del proceso Realizar Actividad Comunitaria.

Nombre:	Realizar Actividad Comunitaria
Objetivos:	Realizar actividades en la comunidad.
Evento(s) que lo generan:	Planificación de Intervenciones de Enfermería.
Precondiciones:	
Poscondiciones:	Registro de todas las actividades a realizar por la enfermera en un rango de tiempo determinado.
Reglas de Negocio:	Regla de negocio (Reglas textuales # 1, 2, 8, 9, 10, 11) Ver documento APS_SIAPS_0116_RNeg_Enf_W.doc.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

	Regla de negocio (Reglas de Relación #4, 5,6, 7, 8, 9, 10, 13) Ver documento APS_SIAPS_0116_RNeg_Enf_W.doc.
Responsables:	Enfermera
Clientes internos:	Paciente
Clientes externos:	CESIM
Entradas:	Planificación de afectaciones
Salidas:	Planificación de Intervenciones de Enfermería
Actividades:	<ol style="list-style-type: none"><li>1. Realizar actividad</li><li>2. Actualizar planificación</li></ol>

### 2.3 Propuesta del Sistema

#### 2.3.1 Requerimientos de Software

El objetivo del flujo de trabajo requerimientos es establecer y mantener un acuerdo con los clientes acerca de lo que el software debe hacer, los requerimientos son condiciones o capacidades que debe alcanzar o poseer un sistema o componente para satisfacer un contrato, estándar u otro documento impuesto formalmente. Los requerimientos deben estar correctamente estructurados, completos y deben ser fáciles de aplicar. [24]

Los requisitos se pueden clasificar en: funcionales y no funcionales.

##### Requisitos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, los mismos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

Después de analizados los procesos del negocio, se han definido los siguientes requisitos funcionales:

##### **RF 1: Gestionar Actividades Comunitarias.**

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

RF 1.1 Buscar Actividades Comunitarias.

RF 1.2 Listar Actividades Comunitarias.

RF 1.3 Agregar Actividades Comunitarias.

RF 1.4 Modificar Actividades Comunitarias.

RF 1.5 Eliminar Actividades Comunitarias.

RF 1.6 Visualizar detalles de las Actividades Comunitarias.

### **RF 2: Realizar Valoración de Enfermería.**

RF 2.1 Registrar Patrón Percepción y Manejo de la Salud.

RF 2.1.1 Agregar Patrón Percepción y Manejo de la Salud.

RF 2.1.2 Modificar Patrón Percepción y Manejo de la Salud.

RF 2.2 Registrar Patrón Nutrición.

RF 2.2.1 Agregar Patrón Nutrición.

RF 2.2.2 Modificar Patrón Nutrición.

RF 2.3 Registrar Patrón Eliminación.

RF 2.3.1 Agregar Patrón Eliminación.

RF 2.3.2 Modificar Patrón Eliminación.

RF 2.4 Registrar Patrón Actividad y Ejercicio.

RF 2.4.1 Agregar Patrón Actividad y Ejercicio.

RF 2.4.2 Modificar Patrón Actividad y Ejercicio.

RF 2.5 Registrar Patrón Sueño y Descanso.

RF 2.5.1 Agregar Patrón Sueño y Descanso.

RF 2.5.2 Modificar Patrón Sueño y Descanso.

RF 2.6 Registrar Patrón Cognitivo y Perceptivo.

RF 2.6.1 Agregar Patrón Cognitivo y Perceptivo.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

RF 2.6.2 Modificar Patrón Cognitivo y Perceptivo.

RF 2.7 Registrar Patrón Autopercepción.

RF 2.7.1 Agregar Patrón Autopercepción.

RF 2.7.2 Modificar Patrón Autopercepción.

RF 2.8 Registrar Patrón Rol y Relaciones.

RF 2.8.1 Agregar Patrón Rol y Relaciones.

RF 2.8.2 Modificar Patrón Rol y Relaciones.

RF 2.9 Registrar Patrón Sexualidad.

RF 2.9.1 Agregar Patrón Sexualidad.

RF 2.9.2 Modificar Patrón Sexualidad.

RF 2.10 Registrar Patrón Adaptación y Tolerancia al estrés.

RF 2.10.1 Agregar Patrón Adaptación y Tolerancia al estrés.

RF 2.10.2 Modificar Patrón Adaptación y Tolerancia al estrés.

RF 2.11 Registrar Patrón Valores y Creencias.

RF 2.11.1 Agregar Patrón Valores y Creencias.

RF 2.11.2 Modificar Patrón Valores y Creencias.

**RF 3: Visualizar Histórico de Valoraciones de Enfermería**

**RF 4: Gestionar Horarios y Afectaciones.**

RF 4.1 Visualizar personal de enfermería por servicios.

RF 4.2 Visualizar acciones de salud de enfermería.

RF 4.3 Registrar horario de enfermería

**Requisitos no Funcionales**

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

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 más seguro, más usable, más rápido, puede ser que haga que más usuarios deseen comprarlo y por ende se convierta en un éxito.

### **Seguridad de acceso y administración de usuarios.**

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

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

Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude.

### **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.

### **Interfaz.**

#### **Interfaces de usuario.**

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

#### **Requerimientos de hardware.**

#### **Estaciones de trabajo.**

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

En la solución se incluyen estaciones de trabajo para las consultas del sistema alas SIAPS, las que necesitan una capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web, se recomienda IE 7, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo de 256 MG de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo Linux.

### **Servidores.**

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

### **Requerimientos de software.**

El sistema debe correr en sistemas operativos Windows, 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.

En este capítulo se modelaron y especificaron los procesos de negocio obteniéndose para cada uno de ellos el Diagrama de Procesos del Negocio. Se obtuvo una perspectiva más amplia del sistema a desarrollar, identificándose los requisitos funcionales y no funcionales que deberán ser cumplidos. Además fueron explicadas detalladamente las acciones a ejecutar con el objetivo de esclarecer los procesos y dar cumplimiento a las necesidades de los usuarios finales del sistema.

### Capítulo 3. Diseño del Sistema.

En este capítulo se realiza el diseño de la solución propuesta con anterioridad, se describe la concepción arquitectónica del sistema alas SIAPS y su aplicación en los componentes Atención Domiciliaria y Comunitaria en el Módulo de Enfermería, con el objetivo de entender la estructura y dinámica de la organización de los mismos.

Se describe el diseño de la solución propuesta con el objetivo de transformar los requisitos funcionales en un diseño de clases, teniéndose en cuenta los aspectos relacionados con los requisitos no funcionales. Además, se definen los distintos patrones a utilizar en el diseño del sistema.

#### 3.1 Objetivos del Flujo de Trabajo Diseño.

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

Sus objetivos fundamentales son:

- Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones.
- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes.
- Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Ser capaces de visualizar y reflexionar sobre el diseño utilizando una notación común.

Crear una abstracción sin costuras de la implementación del sistema, en el sentido de que la implementación es un refinamiento directo del diseño que rellena lo existente sin cambiar la estructura. Esto permite la utilización de tecnologías como la generación de código y la ingeniería de ida y vuelta entre el diseño y la implementación.

### 3.2 Patrones de diseño.

Entre los patrones de diseño que más se utilizan están los GRASP (General Responsibility Assignment Software Patterns) que son los patrones generales para la asignación de responsabilidades. De todos ellos se destacan principalmente por su utilización en el diseño los patrones:

- Experto: es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo.
- Creador: el patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que: tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase.
- Alta cohesión: expresa que la información que almacena una clase debe de ser coherente y 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 que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

En la realización del presente modelo de diseño se pusieron de manifiesto todos estos patrones anteriormente mencionados, utilizados para mejorar la calidad de los diagramas de esta fase.

A continuación algunos ejemplos donde se ponen en práctica estos patrones:

- Experto:

Las Clases CLRegistrar\_Act\_Comunitaria y CMRegistrar\_Act\_Comunitaria, son las responsables de listar y modificar respectivamente las actividades comunitarias ya que las mismas cuentan con la información necesaria para realizar estas actividades.

- Creador:

La clase CPatron\_Funcional es la encargada de la creación de los 11 patrones funcionales que conforman la valoración de enfermería, esta cuenta con las instancias de las clases de cada uno de los 11 tipos de objetos y además cuenta con toda la información necesaria para su creación.

- Bajo acoplamiento

Se utilizan varias clases responsables de agrupar datos, como tipos de objetos e instancias de atributos de otras clases, las cuales permiten que cualquier modificación que se realice en ellas no repercuta en las otras que la contemplan para su funcionamiento.

### 3.3.1 Patrones en la capa de datos

En la capa de datos, y para lograr una mayor flexibilidad y por tanto asimilar mejor los cambios futuros que se produzcan en la base de datos se utilizan los siguientes patrones:

- Active record: consiste en que un objeto envuelve una tupla de una estructura de datos de un recurso externo, como una fila en una base de datos, y adiciona alguna lógica del dominio a dicho objeto.
- Identity field: la idea fundamental de este patrón es que guarda el campo id de la base de datos en un objeto para mantener la relación entre el objeto cargado en memoria y la fila correspondiente en la base de datos.
- Foreign Key Mapping: consiste en mapear una asociación entre objetos por cada relación entre tablas por claves foráneas.
- Query Object: mantener un objeto que permita hacer consultas a la base de datos.

### 3.3 Diagramas de Clases del Diseño.

#### Gestionar Actividades Comunitarias.

A continuación se presenta el diagrama de clases del diseño del requisito Gestionar Actividades Comunitarias, el mismo fue dividido en tres paquetes para una mejor visualización quedando estructurado de la siguiente manera: vista (Fig. 4), control (Fig. 5) y modelo (Fig. 6); para observar el diagrama en el formato integrado remitirse al Anexo 1.



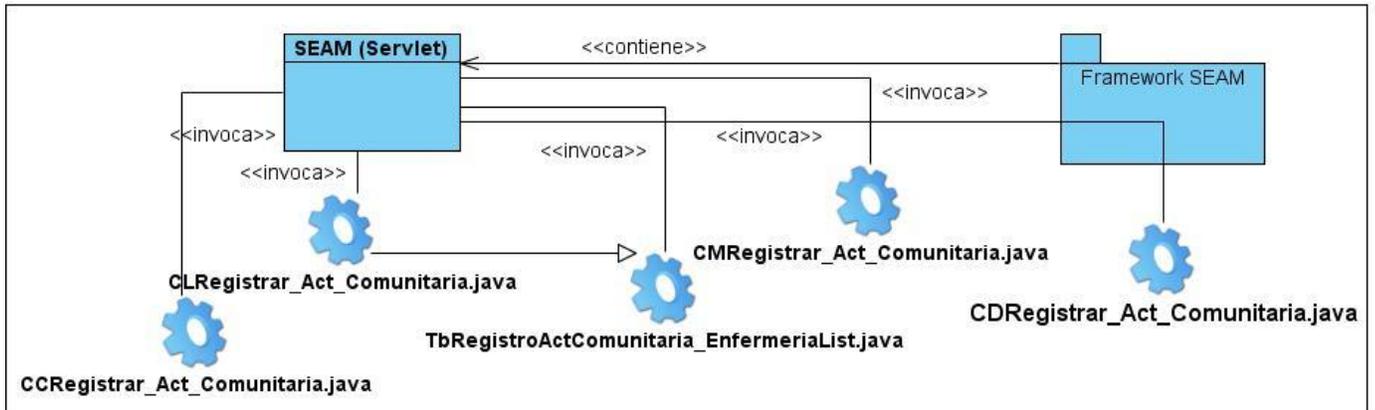


Fig 5: Gestionar Actividades Comunitarias. Control

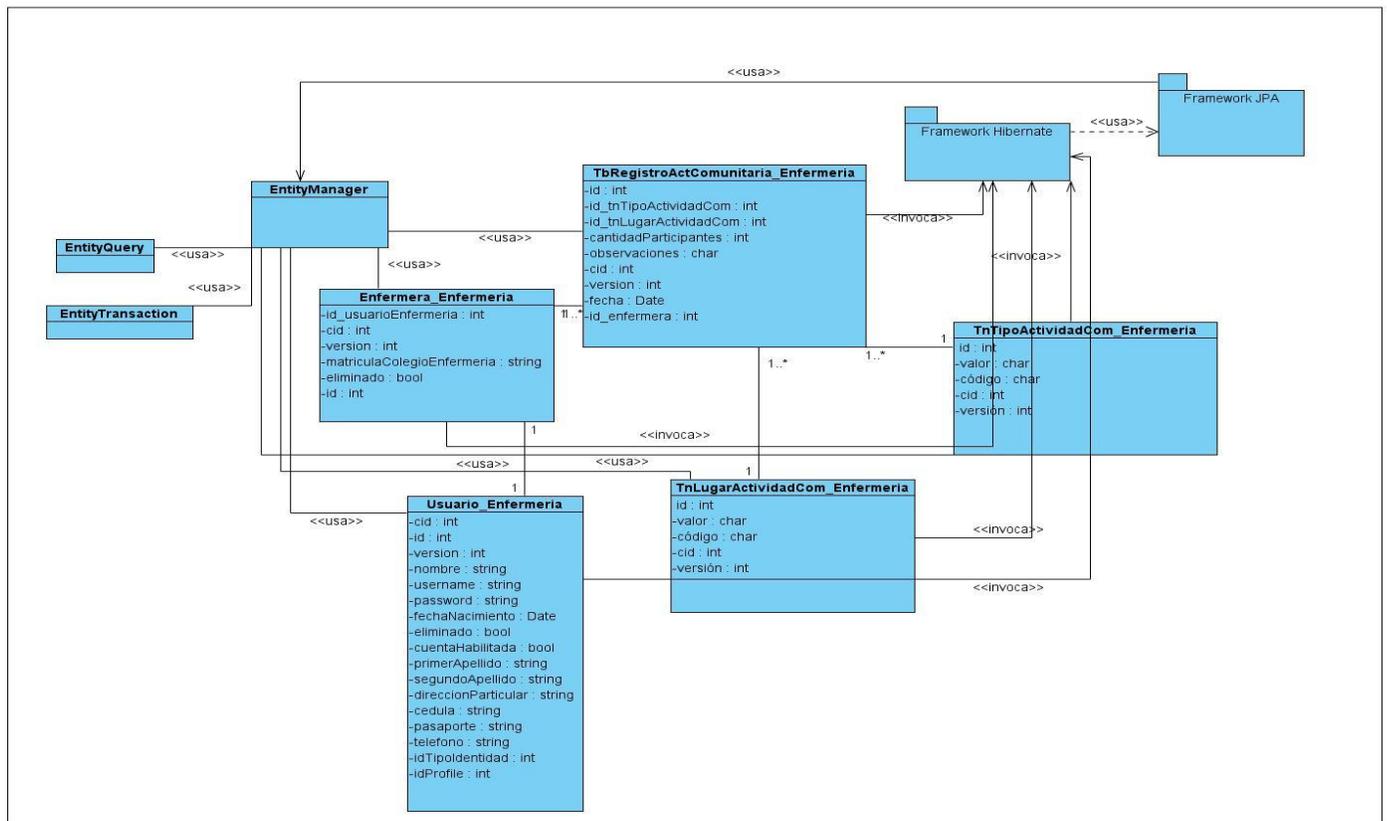
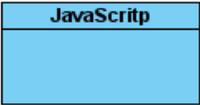
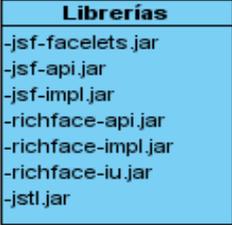


Fig 6: Gestionar Actividades Comunitarias. Modelo.

### 3.4 Descripciones comunes.

A continuación se presenta una descripción de las clases del diseño que son comunes en todos los requisitos funcionales. Ver tabla 5.

Tabla 4: Descripción de las clases comunes.

<b>Capa de Presentación</b>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="text-align: center;">  </div> <p style="text-align: center;">Figura 6. Clase JavaScript</p>	<p>Proveer el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.</p>
<b>Descripción:</b>	
<p>Permite realizar las validaciones del lado del cliente y construir páginas más dinámicas integradas a un navegador web</p>	
<b>Nombre:</b>	<b>Propósito:</b>
<div style="text-align: center;">  </div> <p style="text-align: center;">Figura 7. Clase Librerías</p>	<p>Permite la creación de interfaces.</p>
<b>Descripción:</b>	
<p>Contiene un conjunto de librerías que permiten construir una interfaz de usuario con componentes reutilizables y extensibles. Reducen significativamente la carga de construir y mantener aplicaciones web</p>	

con componentes de interfaz del lado del servidor.

**Nombre:**

**Propósito:**



Figura 8. Clase JSF Servlet

Interceptar las peticiones a las páginas JSF.

**Descripción:**

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

**Nombre:**

**Propósito:**

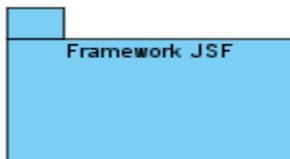


Figura 9. Paquete Framework JSF

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

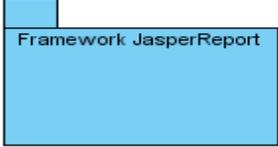
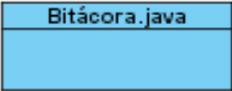
**Descripción:**

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

### Capa de Negocio

**Nombre:**

**Propósito:**

 <p>Figura 10. Clase CRExportar</p>	<p>Proveer las funcionalidades para exportar a diferentes formatos.</p>
<p><b>Descripción:</b></p>	
<p>Permite realizar las funciones para exportar a diferentes tipos de archivos mediante el uso de las librerías del Framework JasperReports.</p>	
 <p>Figura 11. Paquete Framework JasperReports</p>	<p><b>Propósito:</b></p> <p>Añadir características de generación de reportes a aplicaciones java.</p>
<p><b>Descripción:</b></p>	
<p>Es una librería de clases de Java de código abierto desarrollada para facilitar el agregar capacidades de reporte a las aplicaciones Java. Permite realizar reportes de código abierto que tiene como función el llevar documentos ricos en contenido a la pantalla, a la impresora, o a archivos PDF, HTML, XLS, CSV y XML.</p>	
 <p>Figura 12. Clase Bitácora.java</p>	<p><b>Propósito:</b></p> <p>Proveer las funcionalidades para realizar las auditorias del sistema.</p>
<p><b>Descripción:</b></p>	

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

<b>Nombre:</b>	<b>Propósito:</b>
----------------	-------------------



Figura 13. Paquete Framework EJB

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

**Observaciones:**

El Framework EJB está incluido en las capas de Negocio y Persistencia.

**Descripción:**

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

### Capa de Datos

<b>Nombre:</b>	<b>Propósito:</b>
----------------	-------------------



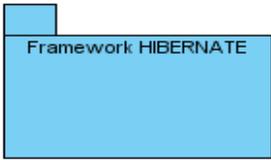
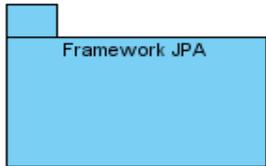
Figura 14. Clase EntityManager

Gestionar las entidades proveyendo servicios de persistencia.

**Descripción:**

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

<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 15. Clase EntityManager</p>	Agregar consultas que pueden aplicarse a las entidades del modelo.
<b>Descripción:</b>	
Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. Permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición.	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 16. Clase EntityManager</p>	Agrupar las operaciones sobre datos persistentes en una unidad transaccional.
<b>Descripción:</b>	
Permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.	
<b>Nombre:</b>	<b>Propósito:</b>
	Agrupar las entidades que contienen la información de las auditorías del sistema.

Figura 17. Clase Entidades Bitácora	
<b>Descripción:</b>	
<p>Contiene el conjunto de entidades que poseen la información de los usuarios y sus trazas en cuanto a sesión utilizada, módulos accedidos, funcionalidades permitidas, acciones realizadas y atributos modificados. Son utilizadas por la clase Bitácora.java para realizar las auditorías del sistema.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 18. Paquete Framework HIBERNATE</p>	<p>Proveer el mapeo objeto/relacional con la base de datos.</p>
<b>Descripción:</b>	
<p>Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional ó ORM de código abierto (Object Relational Mapping) y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada permite generar Bases de Datos en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2, MySQL, entre otras.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 19. Paquete Framework JPA</p>	<p>Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.</p>

<b>Descripción:</b>	
<p>Conjunto de clase agrupadas en componentes que constituyen la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
	<p>Proveer la interacción de la capa de presentación con la de negocio.</p>
	<b>Observaciones:</b>
<p>Figura 20. Clase SEAM Servlet</p>	<p>No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de Negocio.</p>
<b>Descripción:</b>	
<p>Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. Enrutar las peticiones hacia los Beans que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio.</p>	

### 3.5.1 Gestionar Registro de Actividades Comunitarias.

A continuación se presenta una descripción de las clases específicas del requisito Gestionar Registro de Actividades Comunitarias. *Ver tabla 6.*

*Tabla 5: Descripción. Gestionar Registro de Actividades Comunitarias.*

<b>Capa de Presentación</b>
-----------------------------

Nombre:	Propósito:
 <p>Figura 21. Clase agregar.siaps</p>	Proveer la interacción con el usuario.
Descripción:	
<p>La clase agregar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá insertar todos los datos necesarios para proceder con la Actividad Comunitaria y que todo quede correctamente registrado. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>	
Nombre:	Propósito:
 <p>Figura 22. Frmsiaps</p>	Enviar los datos a las páginas servidoras.
Descripción:	
<p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para insertar todos los datos necesarios para proceder con la Actividad Comunitaria y que todo quede correctamente registrado. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>	
Nombre:	Propósito:
	Proveer la interacción con el usuario.

Figura 23. Clase listar.siaps	
<b>Descripción:</b>	
<p>La clase listar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá buscar cualquier registro hecho en alguna fecha o por algún tipo de Actividad Comunitaria en específica. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p style="margin-top: 10px;">Figura 24. Frmsiaps</p>	<p>Enviar los datos a las páginas servidoras.</p>
<b>Descripción:</b>	
<p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para listar todos los datos necesarios de los Registros de Actividades Comunitarias. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p style="margin-top: 10px;">Figura 25. Clase modificar.siaps</p>	<p>Proveer la interacción con el usuario.</p>
<b>Descripción:</b>	
<p>La clase modificar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá acceder a modificar cualquier registro hecho con anterioridad. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa</p>	

su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 26. Frmsiaps</p>	Enviar los datos a las páginas servidoras.
<b>Descripción:</b>	
La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para modificar todos los datos necesarios de cualquier registro hecho con anterioridad. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 27. Clase detalles.siaps</p>	Proveer la interacción con el usuario.
<b>Descripción:</b>	
La clase detalles.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permite ver los detalles de cualquier registro escogido previamente, en la cual se mostrará una serie de datos que no se muestran en el registro. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.	
<b>Nombre:</b>	<b>Propósito:</b>
	Enviar los datos a las páginas servidoras.

Figura 28. Frmsiaps	
<b>Descripción:</b>	
<p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para mostrar todos los datos necesarios de un Registro de Actividad Comunitaria. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>	
<b>Capa de Negocio</b>	
<b>Nombre:</b>	<b>Propósito:</b>
 CCRegistrar_Act_Comunitaria.java  Figura 31. Clase CCRegistrar_Act_Comunitaria.java	Proveer una respuesta a las peticiones realizadas en la vista.
<b>Descripción:</b>	
<p>La clase CCRegistrar_Act_Comunitaria.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de insertar los datos sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 TbRegistroActComunitaria_ErfermeriaList.java  Figura 32. Clase	Proveer una respuesta a las peticiones realizadas en la vista.

TbRegistroActComunitaria_EnfermeriaList.java	
<b>Descripción:</b>	
<p>La clase TbRegistroActComunitaria_EnfermeriaList.java es una clase que se genera de forma automática al realizar la ingeniería inversa y que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de listar los datos solicitados sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 CLRegistrar_Act_Comunitaria.java  Figura 33. CLRegistrar_Act_Comunitaria.java	Proveer una respuesta a las peticiones realizadas en la vista.
<b>Descripción:</b>	
<p>La clase CLRegistrar_Act_Comunitaria.java es una clase que se ejecuta del lado del servidor. Hereda todas las funcionalidades de la clase autogenerada TbRegistroActComunitaria_EnfermeriaList.java y permite guardar los cambios originales de la misma evitando que los mismos se pierdan al realizar la ingeniería inversa. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 CDRegistrar_Act_Comunitaria.java  Figura 35. CDRegistrar_Act_Comunitaria.java	Proveer una respuesta a las peticiones realizadas en la vista.

<b>Descripción:</b>	
<p>La clase CDRRegistrar_Act_Comunitaria.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de mostrar los datos de la Actividad Comunitaria escogida con anterioridad que es solicitado desde la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>CMRegistrar_Act_Comunitaria.java</p> <p>Figura 34. Clase CMRegistrar_Act_Comunitaria.java</p>	<p>Proveer una respuesta a las peticiones realizadas en la vista.</p>
<b>Descripción:</b>	
<p>La clase CMRegistrar_Act_Comunitaria.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de modificar los datos de cualquier registro hecho con anterioridad que se solicite sobre la página cliente correspondiente. Hace uso del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.</p>	
<b>Capa de Datos</b>	
<b>Nombre:</b>	<b>Propósito:</b>

<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>TbRegistroActComunitaria_Enfermeria</b></p> <pre style="margin: 0;">-id : int -id_tnTipoActividadCom : int -id_tnLugarActividadCom : int -cantidadParticipantes : int -observaciones : char -cid : int -version : int -fecha : Date -id_enfermera : int</pre> </div> <p style="text-align: center;">Figura 39. Clase TbRegistroActComunitaria_Enfermeria.java</p>	<p>Proveer el mapeo con la base de datos.</p>
---	---

**Descripción:**

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

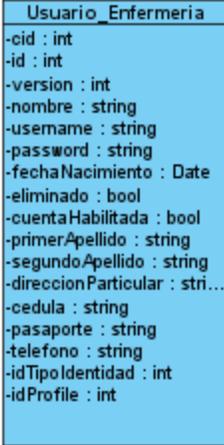
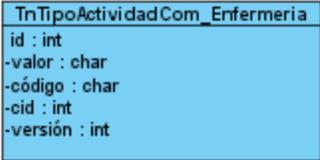
**Nombre:**

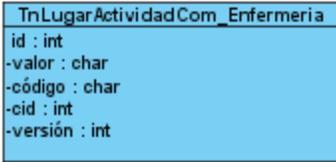
**Propósito:**

<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>Enfermera_Enfermeria</b></p> <pre style="margin: 0;">-id_usuarioEnfermeria : int -cid : int -version : int -matriculaColegioEnfermeria : s... -eliminado : bool -id : int</pre> </div> <p style="text-align: center;">Figura 45. Clase Enfermera_Enfermeria.java</p>	<p>Proveer el mapeo con la base de datos.</p>
--	---

**Descripción:**

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

JPA.	
<b>Nombre:</b>	<b>Propósito:</b>
 <pre style="font-family: monospace; font-size: 0.8em; margin: 0;"> Usuario_Enfermeria -cid : int -id : int -version : int -nombre : string -username : string -password : string -fechaNacimiento : Date -eliminado : bool -cuentaHabilitada : bool -primerApellido : string -segundoApellido : string -direccionParticular : stri... -cedula : string -pasaporte : string -telefono : string -idTipoIdentidad : int -idProfile : int                     </pre> <p style="text-align: center; margin-top: 10px;">Figura 45. Clase Usuario_Enfermeria.java</p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	
<p>La clase Usuario_Enfermeria.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de las enfermeras. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <pre style="font-family: monospace; font-size: 0.8em; margin: 0;"> TnTipoActividadCom_Enfermeria id : int -valor : char -código : char -cid : int -versión : int                     </pre> <p style="text-align: center; margin-top: 10px;">Figura 47. Clase TnTipoActividadCom_Enfermeria.java</p>	<p>Proveer el mapeo con la base de datos.</p>

<b>Descripción:</b>	
<p>La clase TnTipoActividadCom_Enfermeria.java es una clase que se ejecuta del lado del servidor. En ella se guardan las descripciones o nombres referentes a los tipos de Actividades Comunitarias. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p style="text-align: center;">Figura 48. Clase TnLugarActividadCom_Enfermeria.java</p>	<p>Proveer el mapeo con la base de datos.</p>
<b>Descripción:</b>	
<p>La clase TnLugarActividadCom_Enfermeria.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a los lugares donde se realizan las Actividades Comunitarias. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>	

**Nota:** Para obtener más información de las clases del diseño de los procesos Atención Comunitaria, Domiciliaria así como de la Planificación de Intervenciones de Enfermería, remitirse al Expediente de Proyecto. [25]

## **CAPÍTULO 3. DISEÑO DEL SISTEMA**

---

En el capítulo se realizó una descripción de la propuesta de la solución al problema. Se especificaron y justificaron los patrones de diseño utilizados. Además, se documentaron los artefactos obtenidos en el desarrollo del diseño propuesto que constituyen la entrada fundamental para el desarrollo del Flujo de Trabajo de Implementación. A partir de este punto se puede comenzar a construir la aplicación, teniendo en cuenta todos los requerimientos y funcionalidades derivados de esta etapa.

### Capítulo 4. Implementación.

En este capítulo se tratarán los principales aspectos relacionados con la implementación del sistema, constituye la continuidad del Modelo de Diseño. Fundamenta la necesidad de integración del Módulo de Enfermería Web con otros sistemas. Se obtienen los principales artefactos correspondientes a este flujo como el Diagrama de Despliegue de la solución propuesta y se definen también elementos clave para el tratamiento de errores y los estilos de código a utilizar en la aplicación.

#### 4.1 Patrones de arquitectura.

Para el diseño de la aplicación se utiliza el patrón Modelo-Vista-Controlador (MVC). Este patrón es útil principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de mejor manera, facilitando la programación en diferentes capas de forma paralela e independiente.

La vista (o presentación) es la representación de los datos en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web la vista es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.

La capa de presentación está desarrollada básicamente con JSF, usando la librería de componentes RichFaces, esta se integra fácilmente con el framework Seam y permite generar vistas no necesariamente basadas en HTML (PDF, etc.), adiciona además controles listos para usar y el framework de extensión AJAX para los controles JSF. Incluye conversión y validación de campos, 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 dos motores de renderización, mejorando el rendimiento en general además de que brinda la capacidad añadida de la tecnología de plantillas de Facelets. Por su parte los controles para interfaz de usuario de Seam adicionan mejoras a JSF, desde validación e integración de la navegación en la interfaz de usuario basada en flujos de navegación o procesos del negocio.

El controlador es el encargado de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando los datos en caso de que sea requerido. El framework de integración Seam, un poderoso y moderno framework creado para unificar todas las tecnologías estándares JSF, EJB, JPA, además de BPM (Business Process Management). Fue creado desde el inicio para eliminar la

complejidad a nivel desde arquitectura hasta API, se integra con librerías de controles de código abierto basadas en JSF como RichFaces e ICEFaces.

Una de sus principales innovaciones es en el campo de la administración de estado, mientras que en los frameworks tradicionales todo el estado es administrado básicamente en la sesión HTTP, Seam provee el contexto conversacional, así como el asociado a procesos del negocio, con estos se logra un uso más eficiente de la memoria. Integra además el concepto de espacios de trabajo, permitiendo que el usuario tenga en varias ventanas del navegador actividades del negocio con contextos completamente aislados.

Seam integra de forma transparente la administración de procesos del negocio vía JBoss JBPM, haciendo muy fácil implementar y optimizar complejas colaboraciones e interacciones con el usuario. El concepto de conversación permite definir funcionalidades realmente atómicas en las que se involucren varios pedidos al servidor donde los cambios hechos a las entidades solo son persistidos en la base de datos al final de conversación, gracias al contexto de persistencia extendido implementado por Seam y basado en Hibernate.

Como interfaces para comunicarse con sistemas legados se establecen servicios web que serían la fachada para los componentes Seam reales. El modelo es la representación de la información que maneja la aplicación. El modelo en si son los datos puros que puestos en el contexto del sistema provee de información al usuario o a la aplicación misma. Para el acceso a datos se usa la implementación de JPA de Hibernate, minimizando por un lado las configuraciones en XML sin chequeo de tipos y por otro lado usando los servicios del contenedor de EJB y los contextos de persistencias administrados por Seam, se elimina gran parte del código en cuanto a transacciones. Además se pueden establecer validaciones gracias a los Hibernate Validators.

En el aspecto de la seguridad, toda la autorización desde el acceso a directorios, páginas, controles, opciones del menú, servicios del negocio, está basado en reglas. Lo que permite que ninguna de las reglas del negocio esté escrita como código en la aplicación y que el cambio de alguna de ellas, no requiera cambio alguno en el código, solo en la definición de alguna regla en un fichero de configuración. El Seam Security Framework permite todo esto gracias a su integración con el potente motor de reglas JBoss Rules.

### 4.2 Propuesta de integración entre módulos.

En la actualidad casi ninguna aplicación de la salud se concibe como un sistema aislado y el Módulo Enfermería del Subsistema Web no es la excepción. Se contempla su integración con el sistema alas SIAPS como plataforma única para la gestión, procesamiento y transmisión de la información clínica en la APS. La comunicación se realizará entre las bases de datos.

Se relaciona con el Módulo de Medicina Familiar del cual toma todos los datos de los pacientes y sus familiares, así como los de la comunidad en general.

El Módulo Configuración del Subsistema Web le permite al Módulo de Enfermería del Subsistema Web obtener la información relacionada con los Departamentos, el Clasificador Internacional de Enfermedades (CIE-10) y Centros Laborales, para la gestión de sus actividades, además de proporcionar los datos del Personal de Salud y las Entidades a las que pertenece.

A continuación se ejemplifica la integración en la Fig. 7.



Fig 7: Integración del Módulo de Enfermería.

### 4.3 Diagrama de Despliegue.

El objetivo de los diagramas de despliegue es mostrar cómo y dónde se desplegará el sistema. Este modelo muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. En esencia, un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema.

De manera general el diagrama de despliegue del Módulo de Enfermería Web quedaría de la siguiente manera:

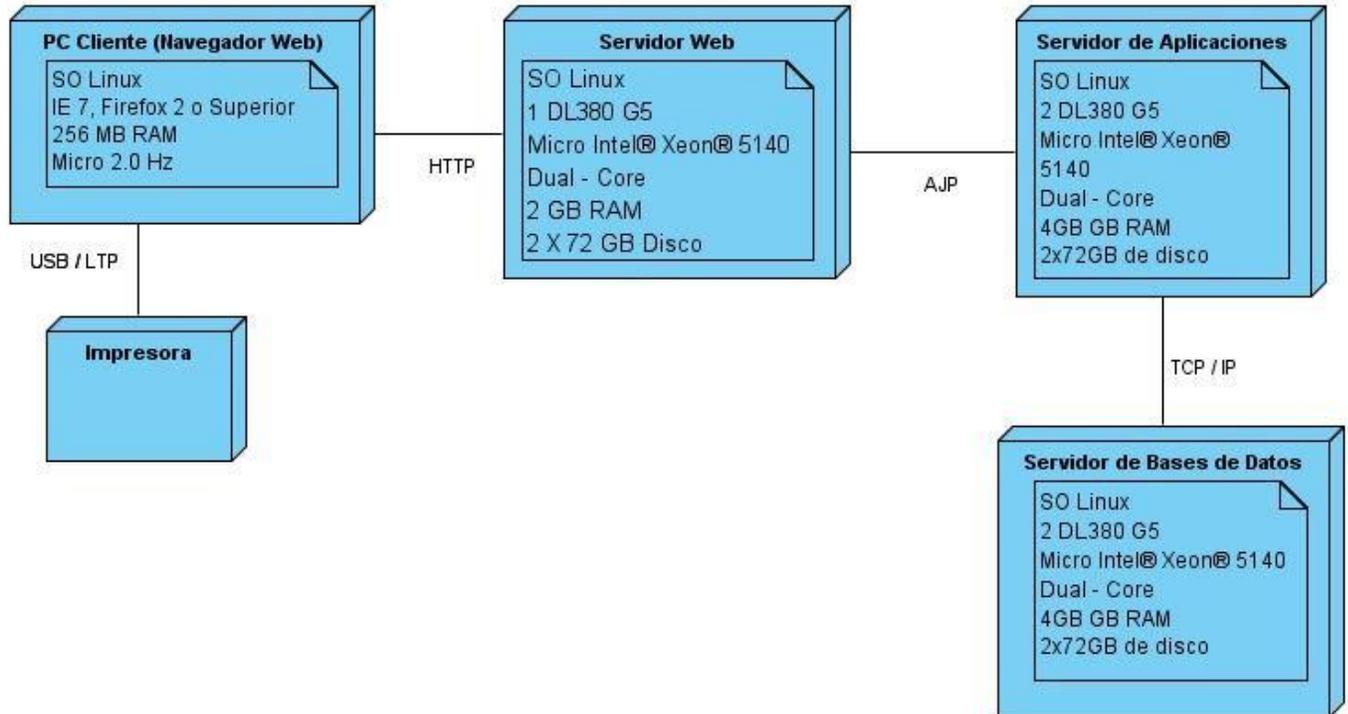


Fig 8: Diagrama de despliegue.

## Descripción de los elementos

**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.

**Procesadores:**

**Servidor(es) de Aplicaciones:**

Sistema Operativo Linux

2 DL380 G5,

Procesador Intel® Xeon® 5140 Dual - Core

4GB de memoria

2x72GB de disco

**Servidor de Base de Datos:**

Sistema Operativo Linux

1 DL380 G5

Procesador Intel® Xeon® 5140 Dual – Core

4GB de memoria

2x72GB de disco

**Servidor de Almacén de Datos:**

Sistema Operativo Linux

1 DL380 G5

Procesador Intel® Xeon® 5140 Dual – Core

4GB de memoria

2x72GB de disco

**Servidor Web:**

Sistema Operativo Linux

1 DL380 G5

Procesador Intel® Xeon® 5140 Dual - Core

2 GB de memoria

2x72GB de disco

### **Estaciones Clientes:**

Sistema Operativo Linux.

IE 7, Firefox 2 o versiones superiores.

256 Mb de memoria RAM

Microprocesador de 2.0 Hz

## **4.4 Tratamiento de excepciones**

Las excepciones son el mecanismo recomendado para el tratamiento de los errores que se produzcan durante la ejecución de las aplicaciones (divisiones por cero, intentos de lectura de archivos dañados). Las mismas crean objetos con información acerca de ellas, que pertenecen a la clase Exception y son llamadas en la ejecución del programa en los puntos donde se encuentra un error, permitiendo la ejecución del programa, aún en presencia de un error.

En el sistema se propone realizar el tratamiento de excepciones en todos los fragmentos de código donde pueda ocurrir un error, especialmente en los lugares donde se manipulan los datos, donde son insertados o modificados en la base de datos y además en las validaciones de los datos que son insertados por los usuarios en la interfaz de usuario. El control de la navegación, en caso de ocurrir una excepción que implique una redirección, se maneja mediante los “.pages.xml”, los mismos se encargan de capturar globalmente las excepciones y ejecutar las instrucciones determinadas.

Para el control de las demás excepciones es utilizado el componente FacesMessages del framework Seam el cual se encarga de mostrar los mensajes que se manejan a través del objeto facesMessages inyectado en las clases controladoras. El mismo trata los mensajes por tipo (error, alerta y notificación).

## **4.5 Estrategias de codificación. Estándares y estilos a utilizar.**

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La forma de mantener del código es la facilidad con que el sistema de software

puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Aunque la legibilidad y los métodos para mantener el código son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones.

Es fundamental mantener una buena indentación para lograr una estructura uniforme en los bloques de código así como para los diferentes niveles de anidamiento. Se recomienda dejar cuatro espacios en blanco desde la instrucción anterior para el inicio y fin de bloque `{}`. Lo mismo sucede para el caso de las instrucciones `if`, `else`, `for`, `while`, `do`, `while`, `switch`, `foreach`.

Deben escribirse comentarios al principio de cada clase y método brindando una breve descripción de los propósitos generales de cada funcionalidad. Los nombres de clases deben ser palabras completas, en mayúsculas y minúsculas, con la primera letra de cada palabra en mayúscula. Los nombres de clases deben ser simples y descriptivos, utilizando palabras completas y acrónimos o abreviaturas (a no ser que la abreviatura sea ampliamente conocida, como URL o HTML).

Los nombres de interfaces deberían seguir las mismas reglas indicadas para las clases. Los métodos deberían ser verbos, en mayúsculas y minúsculas, con la primera letra en minúscula, y la primera letra de cada una de las palabras internas en mayúscula. Todos los nombres de variables de instancia o de clase deben estar constituidos por palabras con la primera letra de la primera palabra en minúscula y la primera letra de las palabras internas en mayúscula. Los nombres de variables deben ser cortos y significativos. La elección de un nombre de variable debe ser mnemotécnica, es decir, pensado para que un lector casual al verla comprenda su uso. Se deben evitar las variables de una sola letra, excepto en variables temporales de corto uso. Nombres comunes para este tipo de variables son: `i`, `j`, `k`, `m` y `n` para enteros; `c`, `d`, y `e` para caracteres. Con respecto al idioma se debe utilizar el español y las palabras no se acentuarán.

A continuación se muestran ejemplos de tres de las funcionalidades desarrolladas; ver Fig. 9, 10 y 11. Para observar otras de las funcionalidades desarrolladas ver Anexos 2, 3 y 4.

**Registrar Actividad Comunitaria** Q Buscar...

---

**Criterios de búsqueda**

Fecha de Inicio:   Fecha de Fin:   Tipo de Actividad:  

[+ Agregar](#)

---

**Listado de Actividades**

<input type="checkbox"/>	Tipo de Actividad	Lugar de Actividad	Fecha Actividad	Cantidad de Participantes		
<input type="checkbox"/>	taller educativo	parque	2011-03-15	5		
<input type="checkbox"/>	taller educativo	uci	2011-03-25	1		
<input type="checkbox"/>	charla	plaza	2011-03-22	34		
<input type="checkbox"/>	charla	uci	2011-03-24	10		
<input type="checkbox"/>	taller	parque	2011-03-18	10		
<input type="checkbox"/>	charla	uci	2011-03-18	10		
<input type="checkbox"/>	charla	uci	2011-03-11	22		
<input type="checkbox"/>	taller educativo	plaza	2011-03-31	67		
<input type="checkbox"/>	taller	parque	2011-03-16	23		
<input type="checkbox"/>	taller	plaza	2011-03-01	12		

Fig 9: Buscar y Listar Actividades Comunitarias.



**Patrón Funcional: Sexualidad**

**Datos Familiares**

Convive con:	Madre, Padre, Abuelos	No. Personas Grupo Familiar:	5	Condiciones Vivienda:	Buena
Rol Familiar:	Jefe Núcleo	Condiciones Económicas:	B		

Cuida de Alguien:	Depende de Alguien:	Problemas de Relación Social:
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Problemas de Relación Familiar:	Problemas de Relación Laboral:	Cambios de Domicilio:
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Se siente solo:	Comparte Problemas:	Pertence Grupos Sociales:
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Pérdidas de Seres Queridos**

Estados Reflejados:

<Seleccione> ▼

**Listado de Estados Reflejados**

**Estados**

**Listado de Estados Reflejados**

No existe información a mostrar.

Comentarios:	Resultado:
<div style="border: 1px solid #ccc; height: 20px; width: 100%;"></div>	<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">                 &lt;Seleccione&gt; ▼             </div>

Aceptar

Cancelar

*Fig 11: Registrar Patrón Funcional Sexualidad*

En este capítulo se relacionaron los principales elementos de la implementación del sistema. Se trazó la estrategia de integración con el resto de los módulos. Se definió una propuesta para el despliegue para la representación estructural del sistema. Además se describieron las convenciones de código o estándares de codificación definidos para la implementación y se expusieron los distintos métodos usados para el tratamiento de errores en la aplicación.

### Conclusiones.

Mediante la elaboración del presente trabajo y luego de darle cumplimiento al objetivo y a las tareas trazadas, se arribó a las siguientes conclusiones:

- Se realizó un análisis de los sistemas existentes encargados en la gestión de los procesos de la APS, quedando demostrada la necesidad de desarrollar un sistema informático para facilitar la gestión de estos procesos; ya que los sistemas estudiados no satisfacen las necesidades existentes.
- Se informatizaron los procesos de negocio de las áreas de Atención Comunitaria, Domiciliaria y Planificación de Intervenciones de Enfermería y además quedaron implementados los componentes web, que permiten facilitar la gestión de la información de estos procesos; los cuales fueron integrados al módulo de Enfermería del SIAPS.

### Recomendaciones.

Con la culminación del presente trabajo y el cumplimiento del objetivo propuesto se proponen las siguientes recomendaciones:

- Desplegar los componentes desarrollados en todas las instituciones médicas donde se gestionen los procesos de la APS.
- Implementar la funcionalidad visualizar actividades de enfermería por día, semana y mes; que le permita al usuario una mejor información sobre las actividades desempeñadas por las enfermeras.
- Definir e implementar reportes estadísticos que contribuyan a obtener información precisa y detallada.
- Integrar los componentes desarrollados a otros sistemas informáticos desplegados en el país que contemplen la gestión de los procesos de la APS.

### Referencias Bibliográficas.

1. Organización Mundial de la Salud. *Atención Primaria de la Salud*. [En línea] 2011. [http://www.who.int/topics/primary\\_health\\_care/es/](http://www.who.int/topics/primary_health_care/es/)
2. **Muratte, Yasmila Valdés**. *Ministerio de Salud Pública*. Ciudad Habana : Editorial Ciencias Médicas, 2006.
3. Servicio Madrileño de Salud. *Papel de la enfermera en la Atención Primaria competencias, funciones e intervenciones*. [En línea] Octubre de 2007. [http://www.codem.es/Docs/doc\\_primaria\\_oct-2007.pdf](http://www.codem.es/Docs/doc_primaria_oct-2007.pdf).
4. Consejo Internacional de Enfermería. *Servir a la comunidad y garantizar la calidad: Las enfermeras al frente de la atención primaria de salud*. [En línea] 2008. <http://www.seapaonline.org/UserFiles/File/Actividades%20de%20Seapa/2008/DIE%202008.pdf>.
5. Ídem a la 4.
6. Organización Mundial de la Salud. *Definición de salud*. [En línea] 2008. <http://definicion.de/salud/>.
7. Organización Mundial de la Salud. *Sistemas de Salud*. [En línea] 2007. [http://www.who.int/topics/health\\_systems/](http://www.who.int/topics/health_systems/).
8. Organización Mundial de la Salud. *Conferencia Internacional sobre Atención Primaria de Salud de Alma-Ata*. [En línea] Septiembre de 1978. <http://www.inclusion-ia.org/espa%F1ol/Norm/AlmaAta-02-1.pdf>.
9. Definición ABC. *Definición de Enfermera*. [En línea] 12 de mayo de 2010. <http://www.definicionabc.com/salud/enfermera.php>.
10. Salud comunitaria. *Monográfico sobre Atención Comunitaria*. [En línea] 2006. <http://saludcomunitaria.wordpress.com/2008/11/22/monografico-sobre-atencion-comunitaria>.
11. **Villadiego, Benjamin Salas**. LA VISITA DOMICILIARIA. [Online] Agosto 23, 2007. <http://www.scribd.com/doc/263477/LA-VISITA-DOMICILIARIA>.
12. **Dña. Ana Barranco Jefe de Producto MoviCare de ICX**. *Informática y Salud . Sesiones Tecnológicas 2003*. [En línea] 2003. <http://www.seis.es/seis/is/is42/inforsalud2003t2.htm#t2Barranco>.
13. **Reynoso, Carlos y Kicillof, Nicolás**. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
14. **J. Homes, Barry y T. Joyce, Daniel**. *Object-Oriented programming with Java*.

## REFERENCIAS BIBLIOGRÁFICAS

---

15. **Becerril C., Francisco.** *Java a su alcance*. ISBN 970-10-1774-9.
16. **Geary, David y Horstmann, Cay.** *Core JavaServer™ Faces*. 2007. 978-0-13-173886-7.
17. **Red Hat.** *RichFaces Developer Guide*. [En línea] 2007.
18. **Ramos, Ing Juan Alonso.** Introducción a Ajax4jsf. [En línea] Abril 9, 2007. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
19. **Liu, Dr Xinyu.** The source for java Technology Collaboration. *Developing applications with Facelets, JSF and JSP*. [En línea] Agosto 29, 2006. <http://today.java.net/pub/a/today/2006/08/29/developing-with-facelets-jsf-jsp.html>.
20. **SeamFramework.org.** *The Seam Framework - Next generation enterprise Java development*. [En línea] 2009. <http://seamframework.org/>.
21. **Alcantar Hernández, Fernando.** GestioPolis. *Mapeo entre clases persistentes utilizando el software Hibernate My Eclipse para conectar a una base de datos con XML*. [En línea] 2008. <http://www.gestipolis.com/administracion-estrategia/hibernate-para-bases-de-datos-con-xml.htm>.
22. **Jboss Community, Hibernate.** *Relational Persistence for Java and .NET*. [En línea] 2010. <http://www.hibernate.org/>.
23. **Martinez, Rafael.** PostgreSQL-es. [En línea] Marzo de 2009. <http://www.postgresql-es.org/>.
24. **Sánchez Fornaris, Maite, Alcantara Rabí, Elvia Dayanis y Hernández Luque, Eyllin.** Vinculado. *Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica*. [En línea] 2010. [http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html).
25. **Pérez Albear, Lisandra.** Repositorio de documentos para el Departamento de Atención Primaria de Salud del Centro de Infomática Médica. [En línea] 2010. <https://repositorio.cesim.prod.uci.cu/svn/aps/enfermeria/WEB/EXPEDIENTE%20DE%20PROYECTO/INGENIERIA/ARQUITECTURA%20Y%20DISENO/>

## Bibliografía

- **Alcantar Hernández, Fernando.** GestioPolis. *Mapeo entre clases persistentes utilizando el software Hibernate My Eclipse para conectar a una base de datos con XML.* [En línea] 2008. <http://www.gestiopolis.com/administracion-estrategia/hibernate-para-bases-de-datos-con-xml.htm>.
- *Aprenda Java como si estuviera en primero.* Universidad de Navarra. España : s.n., 2000.
- **Becerril C., Francisco.** *Java a su alcance.* ISBN 970-10-1774-9.
- Código abierto como alternativa. *Jboss Application Server 4.2.* [En línea] 2009. <http://www.osalt.com/es/jboss>.
- Colegio Oficial de Diplomados en Enfermería de Madrid CODEM. *Grado, Postgrado y Doctorado en Enfermería, el gran cambio de la profesión.* [En línea] 2008. [http://www.codem.es/Tribuna/Hemeroteca/tribuna\\_214.pdf](http://www.codem.es/Tribuna/Hemeroteca/tribuna_214.pdf).
- Consejo Internacional de Enfermería. *Servir a la comunidad y garantizar la calidad: Las enfermeras al frente de la atención primaria de salud.* [En línea] 2008. <http://www.seapaonline.org/UserFiles/File/Actividades%20de%20Seapa/2008/DIE%202008.pdf>.
- Cursos de Informática. *Tratamiento de Excepciones.* [En línea] 2007. <http://elvex.ugr.es/decsai/builder/intro/6.html>.
- Definición ABC. *Definición de Enfermera.* [En línea] 12 de mayo de 2010. <http://www.definicionabc.com/salud/enfermera.php>.
- **Dña. Ana Barranco Jefe de Producto MoviCare de ICX.** Informática y Salud . *Sesiones Tecnológicas 2003.* [En línea] 2003. <http://www.seis.es/seis/is/is42/inforsalud2003t2.htm#t2Barranco>.
- **Geary, David y Horstmann, Cay.** *Core JavaServer™ Faces.* 2007. 978-0-13-173886-7.
- Infomed. *Actuación de Enfermería en APS.* [En línea] 2000. [http://www.sld.cu/sistema\\_de\\_salud/metodologica/enfermeria.html](http://www.sld.cu/sistema_de_salud/metodologica/enfermeria.html).
- Informática en Salud 2009. *Estrategias de Informatización del Sistema Nacional de Salud.* [En línea] 2009. <http://www.informatica2009.sld.cu>.

- **J. Homes, Barry y T. Joyce, Daniel.** *Object-Oriented programming with Java.*
- Jboss Community, Hibernate. *Relational Persistence for Java and .NET.* [En línea] 2010. <http://www.hibernate.org/>.
- Junta de Andalucía. *RichFaces.* [En línea] 2008. <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>.
- **Liu, Dr Xinyu.** The source for java Technology Collaboration. *Developing applications with Facelets, JSF and JSP.* [En línea] Agosto 29, 2006. <http://today.java.net/pub/a/today/2006/08/29/developing-with-facelets-jsf-jsp.html>.
- **Martinez, Rafael.** PostgreSQL-es. [En línea] Marzo de 2009. <http://www.postgresql-es.org/>.
- **Muratte, Yasmila Valdés.** *Ministerio de Salud Pública.* Ciudad Habana : Editorial Ciencias Médicas, 2006.
- Nursing Center. *Nursing Center.* [En línea] julio/agosto de 1997. [http://www.nursingcenter.com/library/JournalArticle.asp?Article\\_ID=102101](http://www.nursingcenter.com/library/JournalArticle.asp?Article_ID=102101).
- Organización Mundial de la Salud. *Atención Primaria de la Salud.* [En línea] 2011. [http://www.who.int/topics/primary\\_health\\_care/es/](http://www.who.int/topics/primary_health_care/es/)
- Organización Mundial de la Salud. *Conferencia Internacional sobre Atención Primaria de Salud de Alma-Ata.* [En línea] Septiembre de 1978. <http://www.inclusion-ia.org/espa%F1ol/Norm/AlmaAta-02-1.pdf>.
- Organización Mundial de la Salud. *Definición de salud.* [En línea] 2008. <http://definicion.de/salud/>.
- Organización Mundial de la Salud. *Sistemas de Salud.* [En línea] 2007. [http://www.who.int/topics/health\\_systems/](http://www.who.int/topics/health_systems/).
- Organización Mundial de la Salud. *Sistemas de Salud.* [En línea] 2007. [http://www.who.int/topics/health\\_systems/es](http://www.who.int/topics/health_systems/es).
- **Pérez Albear, Lisandra.** Repositorio de documentos para el Departamento de Atención Primaria de Salud del Centro de Informática Médica. [En línea] 2010. <https://repositorio.cesim.prod.uci.cu/svn/aps/enfermeria/WEB/EXPEDIENTE%20DE%20PROYECTO/INGENIERIA/ARQUITECTURA%20Y%20DISENO/>

- Portal Nacional de Enfermería. *Libros de enfermería*. [En línea] 2006. <http://www.enfermeria.sld.cu/libros.html>.
- Ramos, Ing Juan Alonso. Introducción a Ajax4jsf. [En línea] Abril 9, 2007. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
- Red Hat. *RichFaces Developer Guide*. [En línea] 2007.
- Reynoso, Carlos y Kicillof, Nicolás. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
- Salud comunitaria. *Monográfico sobre Atención Comunitaria*. [En línea] 2006. <http://saludcomunitaria.wordpress.com/2008/11/22/monografico-sobre-atencion-comunitaria>.
- Sánchez Fornaris, Maite, Alcántara Rabí, Elvia Dayanis y Hernández Luque, Eyllin. Vinculado. *Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica*. [En línea] 2010. [http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html).
- SeamFramework.org. *The Seam Framework - Next generation enterprise Java development*. [En línea] 2009. <http://seamframework.org/>.
- Servicio Madrileño de Salud. *Papel de la enfermera en la Atención Primaria competencias, funciones e intervenciones*. [En línea] Octubre de 2007. [http://www.codem.es/Docs/doc\\_primaria\\_oct-2007.pdf](http://www.codem.es/Docs/doc_primaria_oct-2007.pdf).
- Villadiego, Benjamin Salas. LA VISITA DOMICILIARIA. [Online] Agosto 23, 2007. <http://www.scribd.com/doc/263477/LA-VISITA-DOMICILIARIA>.

## Glosario de Términos

**Atención domiciliar:** Proceso de atención a pacientes hospitalizados en los hogares.

**Atención comunitaria:** Proceso de actividades que son realizadas en la comunidad. Pueden ser con un carácter de educación para la salud y muchas con el objetivo de prevenir enfermedades.

**las SIAPS:** Producto informático nombrado Sistema Integral para la Atención Primaria de Salud, encargado de gestionar los procesos de la APS.

**Informe de Terreno:** Documento que muestra el control de los pacientes a los que se le realiza el terreno.

**MINSAP:** Ministerio de Salud Pública de Cuba.

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

**SNS:** Sistema Nacional de Salud.

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

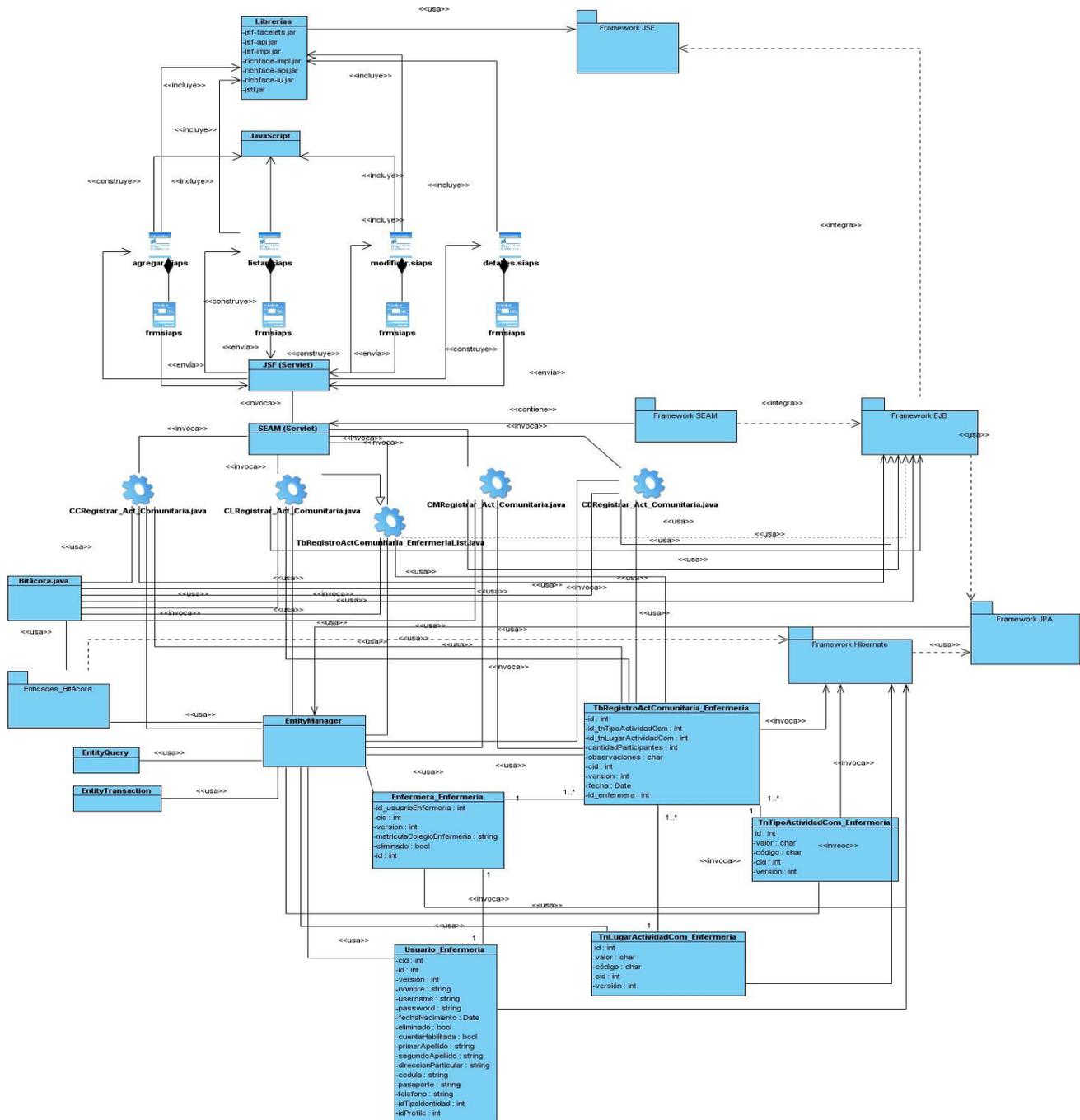
**Componente:** Parte física y reemplazable de un sistema que se ajusta a, y proporciona la realización de, un conjunto de interfaces.

**Área de Salud:** Área geográfica a la que presta sus servicios una Unidad de salud que contemple el Programa de Trabajo del Médico y la Enfermera de la Familia. Son las estructuras fundamentales del sistema sanitario, responsables de la gestión unitaria de los centros y establecimientos del servicio de salud de la Comunidad Autónoma en su demarcación territorial y de las prestaciones sanitarias y programas sanitarios por ellos desarrollados.

**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.

Anexos.



Anexo 1: Diagrama Gestionar Actividades Comunitarias.

**Agregar Actividad Comunitaria**

**Datos de la Actividad Comunitaria**

Actividad: <input type="text" value="&lt;Seleccione&gt;"/>	Lugar de la Actividad: <input type="text" value="&lt;Seleccione&gt;"/>	Fecha: <input type="text"/>
Cantidad de Personas: <input type="text"/>	Observaciones: <input type="text"/>	

Anexo 2: Agregar Actividad Comunitaria.

**Patrón Funcional: Percepción / cognición**

Horas Sueño/Día:  Siestas:  Ronca:

Apnea de Sueño:  Horario Sueño Regular:

**Problemas con el Sueño**

Problemas con el Sueño:

**Listado de Problemas con el Sueño**

**Problemas**

**Listado de Problemas con el Sueño**

No existe información a mostrar.

**Efectos por no Dormir**

Efectos:  Ayudas para Dormir:

**Listado de Efectos por no Dormir**

**Efectos**

**Listado de Efectos por no Dormir**

No existe información a mostrar.

**Descanso/Relajación**

Medicamentos para Descansar:  Horas Descanso/Día:  Influye en sus Actividades:

Comentarios:

Resultado:

Anexo 3: Patrón Funcional Cognitivo Perceptivo.

**Histórico de Valoraciones** Q Buscar...

**Rango de Fechas**

Fecha Desde:  Fecha Hasta:

---

**Línea de Tiempo de Valoraciones de Enfermería**

Fecha Desde:  Fecha Hasta:

2011-04-11	2011-04-12	2011-04-15	2011-04-25	2011-05-25

---

**Listado de Patrones Valorados**

Nombre	
Nutricional metabólico	
Percepción manejo de la salud	
Eliminación	

Anexo 4: Línea De tiempo de Valoración de Enfermería.