



Universidad de las Ciencias Informáticas

Facultad 7

Componente para la propuesta de tratamiento a pacientes de la especialidad de oftalmología del sistema alas HIS.

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

Autores:

Yamila Morales Izaguirre

Víctor Eduardo Cabrera Sanz

Tutores:

Ing. Anny Campos Cosme

Ing. Yuniel García Sánchez

La Habana, junio de 2013

“Año 55 de la Revolución”

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del autor
Yamila Morales Izaguirre

Firma del autor
Víctor Eduardo Cabrera Sanz

Firma del tutor
Anny Campos Cosme

Firma del co-tutor
Yuniel García Sánchez

Datos de Contacto

Ing. Anny Campos Cosme

Graduada en el año 2010 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Ha trabajado vinculada al proyecto Sistema para la Atención Primaria de Salud alas APS. Actualmente labora en el Departamento de Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM) desempeñándose como analista. Ha tutorado varios trabajos de diploma hasta el momento.

Correo electrónico: acampos@uci.cu

Ing. Yuniel García Sánchez

Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2011. Actualmente labora en el Departamento de Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM).

Correo Electrónico: ygarcias@uci.cu

Agradecimientos

Queremos agradecer a nuestros tutores, por toda su entrega en este trabajo, a los profesores del proyecto que nos ayudaron y en general a todos los que contribuyeron a nuestra formación.

A nuestros compañeros de estos 5 años, gracias por su amistad.

De Yamila

Quiero agradecer a mis padres y a mi hermana por darme siempre su amor y cariño cuando más lo necesitaba.

A mi compañero de tesis y amigos por haberme apoyado en todo momento.

De Víctor

Quiero agradecer a mis padres por el esfuerzo que han realizado siempre para darme lo mejor.

A mi novia y a mi compañera de tesis por haberme apoyado incondicionalmente.

A mi hermana y mi sobrino, mi cuñado, mis abuelos, mi tía y mis primos y demás familiares.

Dedicatoria

De Yamila

A mis padres y a mi hermana por su apoyo y comprensión.

De Víctor

A mis padres, a mi hermana y demás familiares y amigos.

Resumen

La emisión de un tratamiento médico es un proceso que ocurre diariamente en las instituciones hospitalarias. En ocasiones este puede resultar engorroso debido a que el especialista no está acostumbrado a tratar con conceptos poco habituales, ya sea en algunos casos a causa de inexperiencia o desconocimiento. El especialista, como apoyo a dicho proceso, puede consultar las Guías Prácticas Clínicas en caso de presentar alguna duda. El Centro de Informática Médica (CESIM), desarrolla el Sistema de Información Hospitalaria alas HIS, como parte de sus productos. Dicho sistema no cuenta con una herramienta que apoye al médico en el proceso de emisión de un tratamiento, provocando dificultades en el desarrollo del mismo. El presente trabajo centra su objetivo en el desarrollo de un componente que apoye al médico en la emisión de un tratamiento a partir de un diagnóstico previamente realizado, como parte del módulo Consulta Externa del Sistema de Información Hospitalaria alas HIS. Su diseño está guiado por el Proceso Unificado de Desarrollo, empleando Eclipse 3.4 como entorno de desarrollo, Java como lenguaje de programación, PostgreSQL 8.4 como gestor de base de datos y Visual Paradigm 6.4 para el modelado del mismo. Con el desarrollo de la presente investigación se espera obtener un componente que apoye el proceso de emisión de tratamientos a pacientes de la especialidad de oftalmología mediante el uso de GPC que normen dicho proceso, brindar al especialista una alternativa viable de tratamiento en dependencia del diagnóstico realizado previamente, así como agilizar el proceso de atención medica en la consulta de oftalmología.

Palabras clave:

Guías Prácticas Clínicas, indicaciones médicas, oftalmología, tratamiento médico,

Índice de Contenido

Índice de Figuras	VIII
Índice de Tablas.....	IX
Introducción	1
Capítulo 1 Fundamentación teórica para el desarrollo de un componente que apoye la emisión de un tratamiento médico	1
1.1. Conceptos básicos relacionados con el dominio del problema.....	1
1.2. Guías de Práctica Clínica.....	2
1.3. Análisis de soluciones existentes.....	6
1.4. Tendencias y tecnologías actuales a considerar.....	10
1.5. Tecnologías utilizadas en el proceso de desarrollo.....	11
1.6. Tecnologías horizontales.....	15
1.7. Metodología de desarrollo de software a emplear.....	15
1.8. Herramientas	16
Capítulo 2 Características del sistema.....	19
2.1. Modelo de Dominio.....	19
2.2. Conceptos fundamentales del dominio.....	19
2.3. Diagrama del modelo de dominio.....	21
2.4. Propuesta de sistema.....	21
2.5. Especificación de los requisitos de software.....	22
2.6. Modelo de casos de uso del sistema.....	25
2.7. Descripción textual de los casos de uso del sistema.....	28

Capítulo 3 Análisis y diseño del sistema	30
3.1. Descripción de la arquitectura, fundamentación	30
3.2. Estrategias de integración	32
3.3. Modelo de diseño.....	33
3.4. Descripción de clases del diseño	36
Capítulo 4 Implementación	41
4.1. Modelo de datos.....	41
4.2. Modelo de implementación.....	46
4.3. Tratamiento de errores.	48
4.4. Seguridad.....	49
4.5. Estrategias de codificación. Estándares y estilos a utilizar.	50
Conclusiones	52
Recomendaciones	53
Bibliografía	54
Referencias bibliográficas	59
Anexos	63
Glosario de términos	71

Índice de Figuras

Figura 1: Modelo de dominio.	21
Figura 2: Diagrama de actores.	26
Figura 3: Diagrama de casos de uso del sistema: <i>Gestionar Tratamiento de GPC</i>	27
Figura 4: Diagrama de casos de uso del sistema: <i>Emitir Tratamiento</i>	27
Figura 5: Diagrama de paquetes de la solución propuesta.	34
Figura 6: Diagrama de clases del diseño: <i>Crear Tratamiento de GPC</i>	35
Figura 7: Diagrama de clases del diseño: <i>Emitir Tratamiento</i>	35
Figura 8: Modelo de datos de la solución propuesta.	42
Figura 9: Subsistemas de implementación por capas.	47
Figura 10: Diagrama de despliegue de la solución propuesta.	48
Figura 11: Crear tratamiento médico.	63
Figura 12: Adicionar presentación de medicamento.	64
Figura 13: Adicionar tratamiento médico.	65
Figura 14: Buscar tratamiento de GPC.	66
Figura 15: Listado de propuestas de tratamiento.	67
Figura 16: Listado de tratamientos médicos indicados, incluye presentación de medicamentos y tratamiento médico.	68
Figura 17: Seleccionar presentación de medicamento.	69
Figura 18: Adicionar tratamiento médico.	70

Índice de Tablas

Tabla 1: Actores del sistema.	26
Tabla 3: Descripción textual del caso de uso: <i>Crear Tratamiento de GPC</i>	28
Tabla 4: Descripción textual del caso de uso: <i>Crear indicación médica</i>	29
Tabla 5: Descripción textual del caso de uso: <i>Emitir Tratamiento Médico</i>	29
Tabla 6: Descripción de la clase controladora: <i>MostrarTratamiento</i>	38
Tabla 7: Descripción de la clase controladora: <i>CrearTratamientoMedico</i>	40
Tabla 8: Descripción de atributos comunes en las entidades.	43
Tabla 9: Descripción de la tabla: <i>tratamiento_enfermedad</i>	43
Tabla 10: Descripción de la tabla: <i>indicaciones</i>	44
Tabla 11: Descripción de la tabla: <i>medicamentos_tratamiento_enfermedad</i>	45
Tabla 12: Descripción de la tabla: <i>tratamiento_enfermedad_indicaciones</i>	45
Tabla 13: Descripción para las tablas de nomencladores.	46

Introducción

En las últimas décadas, el desarrollo científico tecnológico ha provocado el surgimiento de nuevos paradigmas tanto en la información como en las formas y medios usados para su transmisión. El surgimiento de las Tecnologías de la Información y las Comunicaciones (TIC), luego de estos cambios, brindó la posibilidad de crear sistemas informáticos para la gestión de la información.

En la actualidad, el empleo de sistemas informáticos en los principales sectores de la sociedad ha tenido un aumento considerable, gracias a los beneficios que estos aportan. Dentro de estos beneficios podemos encontrar el acceso rápido a la información y por ende mejora en la atención a los usuarios. Además de que la generación de informes e indicadores, permiten corregir fallas difíciles de detectar y controlar con un sistema manual, entre otros.

Es por ello que el gobierno cubano ha decidido llevar a cabo un proceso de informatización y automatización de los procesos que se realizan manualmente en cada uno de estos sectores, por lo que el perfeccionamiento de cada uno de ellos sería un punto clave para su posterior desarrollo. En tal sentido, actualmente se lleva a cabo la tarea de informatizar los procesos que se realizan en las diferentes instituciones y entidades del país.

El sector de la salud es uno de los más grandes y complejos con respecto a la cantidad de información que se maneja, por lo que su procesamiento y almacenamiento resulta complicado. Con el objetivo de contribuir a la informatización del mismo, un conjunto de instituciones que se dedican al desarrollo de software, cooperan en la construcción de soluciones informáticas, siendo la Universidad de las Ciencias Informáticas (UCI) una de las más reconocidas en este aspecto.

El Centro de Informática Médica (CESIM) de la UCI, constituye un núcleo productivo. Su objetivo es el desarrollo de productos, sistemas, servicios y soluciones de alta calidad y competitividad para la optimización del trabajo y mejoramiento de la atención médica. Entre sus productos se encuentra el Sistema de Gestión Hospitalaria alas HIS.

Dicho sistema está orientado a satisfacer las necesidades de almacenamiento, procesamiento e interpretación de los datos médico-administrativos generados en una institución hospitalaria. Constituyen un apoyo para las actividades y procesos de cualquier centro asistencial en todos sus niveles. (1)

Dicho sistema cuenta con 17 módulos entre los que se encuentra Consulta Externa, los médicos que laboran en dicha área se encargan de valorar, diagnosticar y en caso de ser necesario, remitir al paciente para otro servicio o especialidad de cualquier centro asistencial. Consulta Externa abarca un conjunto de especialidades las cuales comúnmente se dividen en: clínicas, quirúrgicas, médico-quirúrgicas y de laboratorios. Cada una de ellas se diferencia en los medios que utilizan para la atención al paciente.

Las especialidades clínicas asisten personalmente al paciente con actividades preventivas, diagnósticas y terapéuticas, sin utilizar técnicas quirúrgicas. Las quirúrgicas utilizan medios quirúrgicos como su nombre lo indica para tratar, modificar o extirpar físicamente la estructura patológica. Por otra parte las médico-quirúrgicas usan tanto técnicas invasivas o quirúrgicas como no invasivas (farmacológicas, etc.). Las especialidades de laboratorio realizan diagnósticos y sugieren tratamientos a los clínicos, y la relación con el propio paciente es reducida. (2)

Entre las especialidades médico-quirúrgicas se encuentra la oftalmología. La misma se encarga del estudio de las enfermedades del globo ocular, la musculatura ocular, sistema lagrimal, párpados y sus tratamientos. Está dedicada a la prevención y tratamientos, tanto médicos como quirúrgicos, de todo lo relacionado al ojo y sus anexos (párpados, vías lagrimales, órbita, etc.) como: miopía, astigmatismo, hipermetropía, cataratas, estrabismo, glaucoma, entre otros. (3)

Las afecciones que pueden ser diagnosticadas a un paciente en una consulta de oftalmología a causa de un síntoma como: dolor ocular, disminución de la visión, enrojecimiento o secreción ocular, cefalea, entre otras; son también llamadas enfermedades oculares. Estas pueden estar relacionadas con el resto del organismo o ser exclusivas del ojo. Muchas son degenerativas o dependientes de la edad.

Con el fin de establecer el tratamiento adecuado, estas enfermedades pueden ser diagnosticadas mediante diferentes pruebas o síntomas y posibles dolencias que dan lugar a un diagnóstico efectivo. (4) Este está asociado a un tratamiento, el cual es emitido por un especialista una vez que ya se ha diagnosticado el paciente.

Actualmente el proceso para emitir un tratamiento médico en el sistema alas HIS es engorroso, ya que el especialista debe agregar para cada tratamiento médico los medicamentos adecuados a partir de su presentación, la cual es brindada por el sistema. Esta presentación no muestra el nombre comercial del medicamento, sino los principios activos que lo componen.

El médico debe conocer cada uno de estos principios activos para poder recetar un determinado medicamento, por lo que se dificulta el proceso de selección de estos, ya que en ocasiones, todos estos conceptos pueden resultar complicados, siendo en algunos casos nuevos para los especialistas ya que están acostumbrados a tratar directamente con el nombre comercial de los medicamentos y realizar su indicación por el mismo. Además el médico debe introducir manualmente los datos referentes a dosis, frecuencia y cantidad de días de cada medicamento recetado y de los tratamientos médicos indicados.

Actualmente el sistema alas HIS no cuenta con una herramienta que facilite y apoye el proceso de emisión de un tratamiento médico. Teniendo en cuenta esto, y de acuerdo con los elementos expuestos anteriormente en la situación problémica surge la necesidad de desarrollar una herramienta que facilite al médico la emisión de un tratamiento para un diagnóstico determinado, permitiendo además al médico emitir sus criterios mediante la modificación de los tratamientos teniendo en cuenta las condiciones clínicas del paciente en cuestión.

Por lo anteriormente expuesto se plantea como **Problema a resolver** de la presente investigación: ¿Cómo realizar la propuesta de tratamientos a pacientes de la especialidad de oftalmología en el sistema alas HIS?

Para el desarrollo de la investigación se define como **objeto de estudio**: proceso de atención médica a pacientes de la especialidad de oftalmología en el sistema alas HIS, enfocado al **campo de acción** del proceso de emisión de tratamientos a pacientes de la especialidad de oftalmología utilizando Guías Prácticas Clínicas en el sistema alas HIS.

Para dar solución al problema planteado se define como **Objetivo general**: Desarrollar un componente que asista el proceso de emisión de tratamientos a pacientes a partir de la utilización de las Guías Prácticas Clínicas en la especialidad de oftalmología del sistema alas HIS.

Para dar cumplimiento al objetivo general se proponen las siguientes **Tareas de la investigación**:

1. Investigar los elementos del dominio asociados a la especialidad de Oftalmología y la emisión de tratamientos en la institución hospitalaria.
2. Evaluar las tendencias actuales de los Sistemas de Información Hospitalaria y aplicaciones para la emisión de tratamientos en la especialidad de Oftalmología en instituciones hospitalarias.

3. Asimilar la arquitectura definida por el Departamento Sistemas de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
4. Obtener, basado en el Proceso Unificado de Desarrollo, los artefactos resultantes del Modelado de Negocio, la Gestión de Requisitos, el Diseño y la Implementación.
5. Implementar las funcionalidades propuestas para la emisión de tratamientos dentro de la especialidad Oftalmología del Sistema de Información Hospitalaria alas HIS.

Con el desarrollo de la presente investigación y la realización del Componente para la propuesta de tratamiento a pacientes de la especialidad de oftalmología del sistema alas HIS se esperan obtener los siguientes beneficios:

- Existencia de un componente que apoye el proceso de emisión de tratamientos a pacientes de la especialidad de oftalmología mediante el uso de GPC que normen dicho proceso.
- Brindar al especialista una alternativa viable de tratamiento en dependencia del diagnóstico realizado previamente.
- Agilizar el proceso de atención medica en la consulta de oftalmología.

El documento, para un mejor entendimiento, estará estructurado de la siguiente manera:

Capítulo 1: Fundamentación teórica para el desarrollo de un componente que apoye la emisión de un tratamiento médico: Se realiza un estudio a nivel mundial de algunos sistemas que emiten tratamiento médico a pacientes. Permitiendo definir el desarrollo de un componente que apoye la emisión de un tratamiento, utilizando las metodologías y tecnologías definidas por la arquitectura del sistema alas HIS.

Capítulo 2: Características del sistema: Contiene un marco conceptual asociado a la información que será manipulada por el sistema. Se realiza la descripción de las características que presentará el sistema, especificando las funcionalidades y requerimientos con los que contará.

Capítulo 3: Análisis y diseño del sistema: Se realiza una descripción detallada de los elementos básicos del diseño del sistema, a partir de la justificación del uso de patrones y diagramas de clases del diseño. Se realiza la modelación y la construcción de la estructura de la aplicación.

Capítulo 4: Implementación: Se realiza una propuesta de solución para lograr una gestión más eficiente de los procesos hospitalarios asociados al área en cuestión. Además se describe la forma en que fueron implementados las clases y subsistemas en términos de componentes.

Capítulo 1 Fundamentación teórica para el desarrollo de un componente que apoye la emisión de un tratamiento médico

Este capítulo tiene como objetivo fundamental, tratar diferentes aspectos que sirven de soporte teórico para el desarrollo de la solución que se propone. Aborda además conceptos fundamentales vinculados al campo de acción y se describen las tecnologías, metodologías y herramientas utilizadas en el desarrollo de la solución.

1.1. Conceptos básicos relacionados con el dominio del problema.

Oftalmología

La oftalmología es la especialidad médica que se encarga del diagnóstico y tratamiento de las enfermedades del aparato de la visión. (7)

El avance de la oftalmología como una especialidad médico-quirúrgica diferenciada con respecto a la medicina y la cirugía general ha dado paso a la formación de áreas de conocimiento y de actividades específicas como son: Baja Visión, Catarata, Córnea y enfermedades externas, Oftalmología General, entre otras. (8)

Enfermedades oculares

Las enfermedades de la vista o enfermedades oculares, como comúnmente son conocidas, se manifiestan con una cantidad de determinados signos y síntomas. Muchas enfermedades de los ojos son inofensivas y no tienen consecuencias para el paciente. Otras pueden tener repercusiones graves y pueden provocar incluso ceguera. Por lo que es importante tratarlas lo antes posible. En ocasiones las enfermedades oculares pueden no presentar ningún síntoma, es por ello que es imprescindible realizar exámenes de la vista regulares para mantener una visión saludable. (8)

Tratamiento

Conjunto de medios de cualquier tipo: higiénicos, farmacológicos, quirúrgicos o bien físicos, que tendrán como finalidad primaria la curación o el alivio de enfermedades o algunos síntomas de estas, una vez que ya se ha llegado al diagnóstico de las mismas. Pueden ser definidos además como el conjunto de medios terapéuticos y de prescripciones higiénicas empleados con objeto de curar una enfermedad. (8)

Existen diversos tipos de tratamientos que pueden ser aplicados en dependencia de la afección que se presente, entre los cuales se destacan: los tratamientos médicos, estos emplean fundamentalmente fármacos; los quirúrgicos, estos emplean los medios de la cirugía para extraer el mal; los específicos, están orientados a atacar la causa que provoca la enfermedad y el paliativo intenta al máximo ofrecer al enfermo el mayor bienestar posible.

El acto médico por excelencia de un oftalmólogo es la refracción o graduación de la vista, en caso de que no se realice correctamente y con una determinada rigurosidad, pueden aparecer enfermedades de polo anterior e incluso otras, como respuesta a un sobreesfuerzo visual. Los tratamientos oculares más empleados para estos casos son: farmacológicos, ópticos y quirúrgicos.

1.2. Guías de Práctica Clínica

Desde los inicios de la medicina, el médico ha intentado basar sus decisiones en el mejor conocimiento científico disponible en cada momento. Sin embargo, la decisión clínica no se toma sólo en base a la propia experiencia o la información derivada de la evidencia científica; sino que también las circunstancias o las preferencias particulares determinan esa decisión.

Como la idea de protocolizar la asistencia no es nueva, en los noventa surgen las guías de práctica clínica de la necesidad de los sistemas hospitalarios de enfrentarse a una evidente variabilidad en la práctica clínica y de la confirmación de que estas variaciones pueden conllevar tratamientos o resultados inadecuados, así como desigualdades en la utilización de los recursos sanitarios. (9) (10)

La variedad de estilos de práctica clínica, la presión de numerosas fuerzas implicadas en el proceso asistencial, las divergencias en criterios sobre el valor relativo de diferentes opciones de tratamiento y los recursos limitados constituyen los principales elementos que motivan la necesidad de elaborar una GPC. Con el transcurso de los años han sido expuestas un conjunto de definiciones acerca de las GPC. A continuación se hace mención de algunas de ellas:

“Son una propuesta que apoya la decisión del médico en la elección de procedimientos y conductas que se ha de seguir ante un paciente en circunstancias clínicas específicas, para mejorar la calidad de la atención médica, reducir el uso de intervenciones innecesarias, ineficaces o dañinas, facilitar el tratamiento de los pacientes con el máximo beneficio y disminuir al mínimo el riesgo de daño.” (11)

“Las guías clínicas son recomendaciones sistemáticas basadas en la evidencia científica disponible, para orientar las decisiones de los profesionales y de los pacientes sobre las intervenciones sanitarias más adecuadas y eficientes en el enfoque de un problema específico relacionado con la salud en circunstancias concretas.” (6)

“Las GPC son instrumentos técnico-médicos que describen los procedimientos normalizados, explícitos, reproducibles y objetivos para el diagnóstico, útiles para sistematizar la conducta que se debe seguir en la atención de determinados problemas de salud y guiar la definición del plan de tratamiento respectivo mediante protocolos y algoritmos que permitan proporcionar una asistencia correcta y abierta a evaluación”. (12)

Todas estas definiciones coinciden en aspectos esenciales de las GPC, pero la más usada, y difundida es la expuesta por el Instituto de Medicina (*IOM* por sus siglas en inglés) de los Estados Unidos, en 1990: "conjunto de recomendaciones desarrolladas sistemáticamente para ayudar a los profesionales y a los pacientes en la toma de decisiones informadas sobre la atención sanitaria más apropiada, la elección de las opciones diagnósticas o terapéuticas más adecuadas en el enfoque de un problema de salud o un cuadro clínico específico". (6)

Las GPC son un instrumento de enlace entre la investigación y la práctica clínica, y es en este sentido en el que ayudan a los profesionales a asimilar, evaluar e implantar la cada vez mayor cantidad de información y literatura científica disponibles, así como las opiniones basadas en la mejor práctica clínica. (10)

Las GPC son un conjunto de recomendaciones para ayudar a los médicos en el proceso de toma de decisiones, sobre cuáles son las estrategias e intervenciones más adecuadas para resolver un problema clínico en circunstancias específicas, por lo que no definen detalladamente la atención al paciente en todo el proceso terapéutico, sino cómo tratar una patología, basándose en revisiones sistemáticas de la bibliografía actual. (6)

Objetivos de las GPC

Entre los objetivos más importantes que se buscan con las GPC están los de mejorar la práctica clínica y su calidad, en el sentido de que dan un soporte científico para ello, educar a los profesionales y a los pacientes ofreciéndoles las mejores evidencias científicas. Pretenden además: disminuir la variación

indeseada en el ejercicio clínico, sistematizar el enfoque de los principales motivos de atención médica que presenta la población, unificar en un marco de flexibilidad clínica, los criterios de atención médica institucional, orientar la toma de decisiones clínicas y favorecer la actualización continua y accesible del conocimiento en todos los niveles de atención. (13)

Características de las GPC

Con el fin de favorecer su implementación e implantación, las GPC deben tener credibilidad, por lo que deben ser desarrolladas por organizaciones o personal de atención a la salud de prestigio reconocido en el área. También su validez es importante, por lo que una vez aplicada debe lograr el beneficio previsto en términos de salud del paciente o económico. Además se deben poder reproducir, es decir que a partir de la misma evidencia científica y utilizando el mismo método, diferentes expertos llegarán a las mismas conclusiones.

En el grupo multidisciplinario responsable de la elaboración de las guías, en un hospital, se deben reflejar todas las opiniones consensuadas de los participantes en los diferentes procesos de atención, por lo que debe existir la representatividad. Deben ser aplicables y contener elementos básicos que garanticen la calidad y permitan flexibilidad en su aplicación.

En la elaboración de las GPC debe evitarse ambigüedades e imprecisiones, incluyendo lo referente a traducción de guías escritas en otro idioma. Se deben especificar el nivel de evidencia científica y el grado de solidez de sus recomendaciones, lo que evidencia su fortaleza. También es indispensable que se especifique el método por el cual se elaboró la guía, así como las personas y organizaciones que participaron en su elaboración, lo que se ve reflejado en su transparencia y es esencial que se defina con qué periodicidad se realizarán las revisiones y actualizaciones.

En las GPC no se pueden incluir todas las situaciones que se producen en la práctica médica real, es por ello que las mismas no definen detalladamente la atención al paciente en todo el proceso terapéutico, sino que son recomendaciones de cómo tratar una patología basándose en revisiones sistemáticas de la bibliografía actual.

Tipos de GPC:

En dependencia del fundamento en que se basen las recomendaciones de las GPC estas pueden clasificarse en tres tipos. En primer lugar se encuentran las GPC basadas en la opinión de expertos, en las que las recomendaciones se realizan según la experiencia personal de los mismos. En estas no existe sistematización en su construcción y no hay una base en la evidencia científica.

Por otro lado se encuentran las GPC basadas en el consenso, que son aquellas que se elaboran con metodología estructurada basada en evidencia, a partir de una reunión de expertos. En este tipo de guía, las recomendaciones son emitidas mediante discusión, con apoyo bibliográfico y experiencia clínica. Por último, se encuentran las GPC basadas en la evidencia, donde el proceso de elaboración se realiza mediante el análisis y evaluación de la evidencia científica disponible, asociada con la opinión y experiencia de los expertos.

Además de que se presentan rigurosa y explícitamente aspectos como identificación y composición del panel de expertos que las elabora, protocolización del sistema de identificación, recogida y evaluación crítica de las publicaciones analizadas, formulación individualizada sobre el nivel de evidencia que sustenta cada recomendación por separado. (15)

Impacto de las GPC

En la actualidad, se manifiesta un interés cada vez más creciente por la promoción, desarrollo y aplicación de GPC, tanto para la prevención, como para la atención de diversas situaciones clínicas. Su elaboración e implementación ha sido un elemento novedoso, que se ha introducido en la asistencia médica que se brinda en los servicios de salud, en los últimos años, ya que han posibilitado un punto de encuentro entre la investigación y la práctica clínica.

Internacionalmente es reconocido el trabajo realizado por varias organizaciones y redes científicas encaminadas a promover la elaboración, implementación y evaluación de las GPC y entre ellas se encuentra la Red Iberoamericana de Guías de Práctica Clínica, conformada en Porto Alegre, Brasil, en el año 2004 y tiene como objetivos compartir información, instrumentos, herramientas y experiencias sobre la elaboración e implantación de guías de práctica clínica y calidad de los servicios hospitalarios y realizar cursos y talleres para socios, asistentes locales no socios y otras actividades de formación sobre calidad sanitaria y guías clínicas. (16)

En Cuba existe un Programa Nacional de GPC, desde el año 2005, coordinado por la Dirección Nacional de Ciencia y Técnica del Ministerio de Salud Pública. Muchos han sido los grupos de autores en nuestro país, que han escrito "Guías de Prácticas Clínicas" de diversas especialidades, con la utilización de métodos basados en opinión de expertos o basadas en el consenso. Actualmente constituye una necesidad concentrar esfuerzos en elaborar guías de buena calidad en el contexto nacional para mejorar la atención médica.

1.3. Análisis de soluciones existentes.

Ámbito Nacional

HIPERTENCID

Sistema para el diagnóstico, control y seguimiento de pacientes con hipertensión arterial desarrollado por el Instituto Central de Investigaciones Digitales (ICID).

Hiperweb

Es una versión superior a HIPERTENCID, desarrollado en la UCI en el 2006, implementado sobre tecnología web. Este sistema además de proveer al médico de una herramienta para el apoyo a la toma de decisiones, permite organizar la historia clínica del paciente, así como mejorar su trabajo. Es un sistema distribuido, basado en una plataforma web, telemedicina, acceso simultáneo de usuarios restringido por roles, brinda servicios a nivel de intranet en los propios hospitales o internet, mediante misiones médicas cubanas.

En correspondencia con sus características Hiperweb es un sistema basado en técnicas de Inteligencia Artificial (IA) y en el manejo de configuraciones dinámicas mediante el empleo de tecnologías como XML. Está estructurado en dos módulos: indicación y control del tratamiento no farmacológico de los pacientes hipertensos, así como un módulo orientado a mejorar el estilo de vida de los pacientes. Desarrollado en el entorno de Visual Studio.NET 2003, ASP.NET y C# como lenguaje de programación, en la plataforma de Microsoft .NET, soporta además el sistema operativo Microsoft.

DITRITS

Es un *sistema experto* para el diagnóstico y tratamiento de infecciones de transmisión sexual, desarrollado por la universidad de Cienfuegos conjuntamente con especialistas de segundo grado del Hospital Provincial Dr. Gustavo Aldereguía Lima. Se caracteriza por utilizar un modelo de producción basado en reglas. Está conformado por dos módulos fundamentales, un ejecutable desarrollado en Borland Delphi 7 y otro en ArityProlog. Como aplicación constituye el primer producto cubano para el diagnóstico y tratamiento de Infecciones de Transmisión Sexual (ITS). La interacción con el usuario se produce en lenguaje natural de una forma interactiva.

Dicho sistema es una aplicación *shareware*, se ejecuta sobre la plataforma MS-DOS, Windows o OS/2. A pesar de las posibilidades y ventajas que brinda en esta área, DITRITS centra sus diagnósticos solo en ITS por lo cual tiene un dominio de aplicación restringido.

Ámbito Internacional

Causal ASociational NETwork (CASNET)

Es un *sistema experto* desarrollado en 1976 en la Universidad de Rutgers, con el objetivo de ayudar a los médicos en el diagnóstico y tratamiento del glaucoma (enfermedad ocular). Este sistema está estructurado por observaciones, tales como: síntomas, signos y resultados de las pruebas de laboratorio; estados patofisiológicos, estos son alteraciones que se producen en el funcionamiento de un órgano- ojo y estados de enfermedad donde las enfermedades se encuentran clasificadas en un *árbol taxonómico*, donde los nodos inferiores corresponden a especificaciones de los nodos superiores.

Una característica de este sistema es que los procedimientos de razonamiento interpretan los hallazgos de un paciente particular en términos de un modelo de una *red causal asociativa o semántica*. Esta caracteriza los mecanismos fisio-patológicos y el curso clínico de enfermedades tratadas y no tratadas.

Como parte de sus aportes están la generación de interpretaciones complejas a partir de un modelo cualitativo de un proceso de enfermedad y el razonamiento sobre el manejo del seguimiento detallado de

un paciente. Además la incorporación de opiniones alternativas de expertos sobre los temas bajo debate, y su prueba y actualización mediante una red colaborativa informática de investigadores en glaucoma.

MYCIN

Sistema experto desarrollado en 1976. Fue escrito en *LISP*, lo que le permitió gran flexibilidad pero a la vez pérdida de *modularidad* y claridad. Inicialmente estaba inspirado en *Dendral*. Su principal función consistía en el diagnóstico de enfermedades infecciosas de la sangre; era capaz de “razonar” el proceso seguido para llegar a estos diagnósticos, y de recetar medicaciones personalizadas a cada paciente según su estatura, peso, etc. A partir de los resultados de análisis de sangre, cultivos bacterianos y demás datos, el programa era capaz de determinar, o en lo menos, sugerir el microorganismo que estaba causando la infección.

Aunque MYCIN se desarrolló posteriormente a *DENDRAL*, es considerado pionero de los sistemas expertos, ya que introduce la separación entre la *base de conocimiento* y el *motor de inferencia*. MYCIN era un sistema lo bastante robusto como para determinar correctamente conclusiones cuando algunas evidencias eran incompletas o incorrectas, explicando sus propios procesos de razonamiento.

En investigaciones realizadas por Stanford Medical School, se determinó que MYCIN tenía una tasa de aciertos de aproximadamente el 65%, que mejoraba las estadísticas de la mayoría de los médicos no especializados en el diagnóstico de infecciones bacterianas de aquella época.

A pesar de que este sistema supuso el punto de partida para el desarrollo posterior de un conjunto de sistemas expertos, poco a poco fue cayendo en desuso por las debilidades que presentaba, en su mayoría inherentes a su técnica. La excesiva dificultad del mantenimiento del programa, condujo a que surgiera otra polémica de cuestiones éticas y legales, debido a que si el diagnóstico de un paciente era incorrecto, se caía en el problema de quien debería ser culpado, ¿el médico o el programador?

ONCOCIN

Es un *sistema experto* automático para monitorizar el tratamiento de pacientes enfermos de cáncer, desarrollado en la universidad de Standford. Proporciona al médico recomendaciones sobre cómo tratar al

paciente. Además ayuda al médico en el tratamiento y manejo de pacientes cancerosos, que se encuentran sometidos a prácticas de quimioterapia, conocidas como protocolos.

Este sistema asigna protocolos de tratamientos estándar a los pacientes en función de su historial, el resultado de los últimos análisis de laboratorio, el último examen clínico y el tratamiento que ya se está siguiendo contra la enfermedad. Incluye base de datos de la especificación de los distintos protocolos que existen para el tratamiento del cáncer, datos relativos a los pacientes, como los resultados de los tests.

Dicho sistema, para seleccionar una determinada terapia, toma en cuenta toda la información relacionada con el paciente, incluyendo el diagnóstico, tratamientos previos y resultados del laboratorio. Contiene conocimiento experto acerca de un numeroso grupo de enfermedades de Hodgkin y protocolos de linfoma. El conocimiento es representado utilizando reglas con encadenamiento hacia adelante y hacia atrás.

Como parte del estudio realizado de las soluciones existentes en el mundo se identificó que estos sistemas aunque en su momento lograron un buen desempeño partiendo de sus potencialidades, no resuelven el problema identificado porque solo son factibles en un estrecho dominio, por lo que no son aplicables para todas las enfermedades de la oftalmología. La mayoría están desarrollados bajo tecnologías privativas lo que impide acceder o modificar su código fuente, impidiendo incorporarlos al sistema alas HIS para su funcionamiento como parte del mismo. Además dichos sistemas no hacen uso de la GPC para la especialidad de oftalmología.

A pesar de que la mayoría de los sistemas estudiados son clasificados como sistemas expertos la propuesta que se realiza no lo es pues los sistemas de apoyo a la salud deben constituir una herramienta que ayude al médico en su trabajo pero permitiéndole tener completa responsabilidad por los criterios emitidos.

Por lo anteriormente expuesto se determina la necesidad de desarrollar un componente que apoye la emisión de un tratamiento médico en la especialidad de oftalmología como parte del módulo de Consulta Externa del Sistema de Información Hospitalaria alas HIS, que cumpla con la arquitectura definida por el mismo.

1.4. Tendencias y tecnologías actuales a considerar.

Como el componente de propuesta de tratamiento deberá formar parte del sistema alas HIS se hace necesario realizar un estudio de la arquitectura del mismo, así como las tendencias tecnológicas que permitirán el desarrollo de dicho componente.

1.4.1 Arquitectura de Software

Arquitectura cliente-servidor

El sistema alas HIS está basado en la arquitectura cliente-servidor, por lo que puede ser accesible desde disímiles ubicaciones. Esta arquitectura consiste básicamente en que un programa cliente realiza peticiones a otro programa servidor que le da respuesta. (17) Las aplicaciones de este tipo posibilitan tener los datos almacenados centralmente en bases de datos, accesibles desde un navegador web o una terminal móvil, ya que la lógica se ejecuta en el servidor y el diseño de la interfaz es transferido a dichas terminales.

La instalación y despliegue de las mismas es sencilla, pues solo necesita montarse el servidor de aplicaciones y este será accedido por los clientes, independientemente de la plataforma, la arquitectura de máquina y la localización física que tengan dichas computadoras.

Arquitectura en capas

El patrón en capas es un estilo de diseño cuyo objetivo fundamental es la separación y agrupamiento de los componentes del software atendiendo a la función que cumplen en el mismo. Para ello se tiene en cuenta las funcionalidades relacionadas con el usuario del sistema, así como la información que este gestiona y las operaciones que realiza sobre la misma según la complejidad que se necesita que presente el sistema. Tal división se realiza en tres capas: la capa de presentación, la capa de negocio y la capa de datos. (19)

Patrón arquitectónico Modelo Vista Controlador

El Modelo Vista Controlador (MVC), es un patrón de arquitectura que separa la parte lógica de una aplicación, de su presentación. Sirve para separar el lenguaje de programación del HTML (HyperText Markup Language por sus siglas en inglés) lo más posible y para poder reutilizar componentes con gran facilidad. El uso de dicho patrón brinda la posibilidad de agregar fácilmente nuevos tipos de datos según lo

requiera la aplicación, ya que son independientes del funcionamiento de otras capas, facilitando el mantenimiento en caso de errores.

Modelo: representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

Vista: es la información presentada al usuario. Una vista puede ser una página web o una parte de ella.

Controlador: actúa como intermediario entre el modelo, la vista y cualquier otro recurso necesario para generar una página web. (18)

1.5. Tecnologías utilizadas en el proceso de desarrollo

Como requisito principal de que el componente de propuesta de tratamiento a pacientes forme parte del sistema alas HIS, se hace necesario asimilar la arquitectura definida por el Departamento Sistemas de Gestión Hospitalaria para el desarrollo de sus aplicaciones. De esta forma siguiendo la línea de desarrollo de dicho departamento, a continuación se describe el lenguaje de programación, así como las tecnologías y herramientas empleadas. Las tecnologías se especificarán en dependencia de su ubicación en la arquitectura en capas.

1.5.1 Java

Lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Permite la ejecución de un mismo programa en múltiples sistemas operativos y ordenadores. Es además un lenguaje compilado, pues genera ficheros de clases compiladas, que son interpretadas por la máquina virtual de java la cual mantiene el control sobre las clases que se estén ejecutando.

Gracias a su eficiencia, versatilidad y portabilidad, Java permite desarrollar software en una plataforma y ejecutarlo en otras. Además permite crear programas que funcionen en un navegador y servicios web, así como desarrollar potentes y eficientes aplicaciones para servidores, procesamiento de formularios HTML, etc.

1.5.2 Capa de presentación

Su función principal consiste en exhibir el sistema al usuario, comunicándole la información y capturando la que este introduce en un mínimo de procesos. Esta capa se comunica únicamente con la capa de negocio. (20) En esta capa se dispone de las siguientes tecnologías.

Java Server Faces (JSF)

Es un framework de desarrollo basado en el patrón MVC (Modelo-Vista-Controlador) para aplicaciones java de tipo web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java Enterprise Edition (Java EE por sus siglas en inglés). Permite además crear interfaces de usuario (UI por sus siglas en inglés) para aplicaciones web, mediante componentes reutilizables. Permite el manejo de estados y eventos, así como la asociación entre los datos de la interfaz y los datos de la aplicación web. (21)

Ajax4JSF

Es una librería de código abierto. Integrada 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. (22) Presenta mejoras sobre los beneficios del framework JSF donde incluye el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. Permite definir un evento en una página, que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el árbol de componentes JSF. (23)

Facelets

Framework simplificado de presentación, en el cual es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. (24)

Incluye características, entre las que se destacan la facilidad en la creación del templating para los componentes y páginas y la habilidad de separar los UIComponents en diferentes archivos, cuenta además de un buen sistema de reporte de errores.

RichFaces 3.3.1

Es una biblioteca de componentes para JSF, constituye un avanzado framework para la integración de Ajax con facilidad en la capacidad de desarrollo de aplicaciones de negocio. Provee facilidades de validación y conversión de los datos proporcionados por el usuario y administración avanzada de recursos como imágenes, código java script y hojas de estilo en cascada (CSS).

Se integra completamente dentro del ciclo de vida JSF. Permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes listos para usar, altamente configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia, lo que mejora la experiencia de usuario. (25)

Seam UI

Framework de código abierto que enlaza diferentes tecnologías y estándares java, lo cual garantiza la plena comunicación de los elementos de la capa de presentación, de negocio y de acceso a datos. Con este framework basta agregar anotaciones propias de éste a los objetos entidad y session de EJB, lo que permite escribir menos código java y XML. Otra característica relevante es que se puede hacer validaciones en los POJOs (Plain Old Java Object) y además manejar directamente la lógica de la aplicación y de negocios desde las sessionsbean. (26)

1.5.3 Capa de negocio

En esta capa se reciben las peticiones del usuario y se envían las respuestas tras el proceso. En esta capa se establecen todas las reglas que deben cumplirse. La misma se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para almacenar o recuperar los mismos. (20) Dispone de las tecnologías que se mencionan a continuación.

Seam

Es una potente plataforma de desarrollo de código abierto para crear aplicaciones de Internet en Java. Integra tecnologías como JavaScript y XML (AJAX), Java Server Faces (JSF), Java Persistence (JPA), Enterprise Java Beans (EJB3.0) y Business Process Management (BPM) en un unificado conjunto de soluciones, completado con herramientas sofisticadas. (27)

Facilidades como la eliminación en gran medida de la complejidad existente desde el nivel de arquitectura hasta el nivel API. Permite además el desarrollo de aplicaciones web basadas en Plain Old Java Objects (POJO) y componentes de UI. Se integra con librerías de controles de código abierto basadas en JSF. Se emplea como framework de integración entre los componentes visuales y los de acceso a datos.

1.5.4 Capa de acceso a datos

La capa de acceso a datos contiene las clases que interactúan con la base de datos, que utilizan consultas para realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

Hibernate

Framework que provee herramientas de mapeo objeto/relacional y permite reducir significativamente el tiempo de desarrollo. Facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos eXtensible Markup Language (XML por sus siglas en inglés) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hibernate 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. (28)

Enterprise Java Bean 3.0

Constituyen una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems (actualmente JEE 5.0). Una de sus metas es la de poder escribir de manera fácil aplicaciones de negocio orientadas a objetos y distribuidas, basadas en el lenguaje de programación Java. Su propósito es el de proveer el soporte de la arquitectura de EJB y al mismo tiempo, reducir la complejidad para el desarrollo de aplicaciones empresariales. (29)

Java Persistence API

Proporciona un modelo de persistencia basado en POJO's para mapear bases de datos relacionales en Java. El Java Persistence API fue desarrollado por el grupo de expertos de EJB 3.0 como parte de Java

Specification Request (JSR 220 por sus siglas en inglés), aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes.

El mapeo objeto/relacional, es decir, la relación entre entidades Java y tablas de la base de datos, se realiza mediante anotaciones en las propias clases de entidad, por lo que no se requieren ficheros descriptores *XML*. También pueden definirse transacciones como anotaciones JPA.

1.6. Tecnologías horizontales

Como *tecnologías horizontales* se utilizaron:

1.6.1 Plataforma Java para Aplicaciones Empresariales 5 (JavaEE 5)

Plataforma de programación (parte de la plataforma java) para desarrollar y ejecutar software de aplicaciones en lenguaje de programación java con arquitectura de n niveles distribuida. Están contruidos sobre una infraestructura base: los servidores de aplicaciones.

1.6.2 Entorno en tiempo de ejecución java 6.0 (JRE 6)

JRE, acrónimo de Java Runtime Environment (entorno en tiempo de ejecución java) que se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas. La máquina virtual de Java (JVM por sus siglas en inglés) es una instancia de JRE en tiempo de ejecución.

1.7. Metodología de desarrollo de software a emplear.

Como metodología de desarrollo de software se utiliza el Proceso Unificado de Desarrollo (RUP por sus siglas en inglés) y el Lenguaje Unificado de Modelado (UML) como lenguaje de modelado.

1.7.1. Proceso Unificado Racional (RUP)

El Proceso Unificado de Desarrollo (RUP por sus siglas en inglés) es un proceso de desarrollo de software utilizado para el análisis, implementación y documentación de sistemas orientados a objetos. Iterativo e incremental, guiado por casos de uso y centrado en la arquitectura, constituyen sus principales características. Incluye artefactos y roles.

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización, por lo que abarca tanto pequeños como grandes proyectos software.

El proceso de desarrollo especificado en RUP está dividido en cuatro fases: la de inicio, define el alcance del proyecto; la de elaboración, se define analiza y diseña la solución; la de construcción, se implementa la solución y transición, donde se concluye el proyecto y se pone en producción.

RUP contiene las mejores prácticas de la ingeniería para el desarrollo exitoso de un proceso de desarrollo de software, que ofrecen al equipo de desarrollo un número significativo de ventajas pues lo provee de guías, plantillas y herramientas para una mejor forma de trabajo. El desarrollo iterativo del software, la administración de requerimientos, el modelado visual del software mediante UML, así como verificar continuamente la calidad del software y controlar los cambios del mismo constituyen las principales posibilidades que el mismo brinda mediante su uso.

1.7.2. Lenguaje Unificado de Modelado 2.0 (UML)

El Lenguaje de Modelado Unificado (UML por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

UML permite modelar sistemas conceptuales como lo son procesos de negocio y funciones de sistema. Además de elementos concretos como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. (30) Este lenguaje de modelado fusiona todas las técnicas que incorporaban los diferentes métodos en sus nuevas versiones para formar una herramienta completa, compartida entre todos los ingenieros de software que trabajan en el desarrollo orientado a objetos.

1.8. Herramientas

Eclipse 3.4

Eclipse es un entorno de desarrollo integrado (IDE por sus siglas en inglés) portable, de código abierto y multiplataforma. Es una potente y completa plataforma de programación, desarrollo y compilación de

elementos tan variados como sitios web, programas en C++ o aplicaciones Java. Brinda una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plug-ins.

Permite proporcionar toda sus funcionalidades a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Dicha arquitectura permite además integrar diversos lenguajes sobre un mismo IDE e introducir otras aplicaciones.

Entre sus principales características se puede destacar que posee un editor visual con sintaxis coloreada, permite la compilación incremental de código, modifica e inspecciona valores de variables, avisa de errores cometidos mediante una ventana secundaria y depura el código que reside en una máquina remota.(31) Es soportado en sistemas operativos como: GNU/Linux, Windows, Solaris 8, entre otros.

Visual Paradigm 6.4

Es una herramienta de Ingeniería de Software Asistida por Computadora (CASE por sus siglas en inglés) que utiliza UML para visualizar y diseñar elementos de software. Ofrece una gama de facilidades para el modelado de aplicaciones.

Soporta el ciclo de vida completo del desarrollo de software: modelado de negocio, análisis y diseño orientados a objetos, implementación, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases y provee soporte para la generación de código por lo que ofrece una ayuda para la construcción rápida de aplicaciones de calidad. Se integra con diversos IDE's como NetBeans, JDeveloper, Eclipse, JBuilder. Brinda la posibilidad de realizar ingeniería inversa para aplicaciones realizadas en Java, .NET, XML e Hibernate. Además tiene disponibilidad en múltiples plataformas. (32)

PostgreSQL 8.4

Es un Sistema Gestor de Base de Datos (SGBD) relacional. Puede ejecutarse sobre la mayoría de los sistemas operativos actuales. Incluye características orientadas a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, etc. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. (33) Como producto de su licencia *BSD* (Berkeley Software Distribution), PostgreSQL puede ser usado, modificado y distribuido por cualquiera, libre de costo para cualquier propósito, ya sea privado, comercial o académico.

Luego de realizado un estudio de los sistemas existentes se identificó que respecto a sus características estos sistemas no pueden ser aplicables al dominio del problema, ya que solo son factibles en un estrecho dominio y no son aplicables para todas las enfermedades de la oftalmología; de ahí la necesidad de implementar un componente que sea parte del sistema alas HIS y que ayude al médico en la emisión de un tratamiento oftalmológico a un paciente a partir de un diagnóstico médico, haciendo uso de las GPC.

Se realizó el estudio de la arquitectura del sistema alas HIS, así como las tendencias tecnológicas que permiten desarrollar el componente de propuesta de tratamiento a pacientes de la especialidad de oftalmología como parte de dicho sistema.

Capítulo 2 Características del sistema

Para el desarrollo de cualquier sistema se debe realizar un análisis de sus características, por lo que es necesario y constituye un eslabón fundamental la comprensión de los procesos existentes en el dominio del problema a resolver. En este capítulo se realiza la descripción de la solución propuesta. Se especifican los requerimientos funcionales que el componente debe cumplir, así como los elementos fundamentales para su diseño.

Debido a que no existe una definición clara de los procesos del negocio en el entorno en que se desplegará el componente, surge la necesidad de desarrollar un modelo de dominio que abarque las definiciones asociadas a la aplicación, así como las relaciones existentes entre ellas.

2.1. Modelo de Dominio

El modelo de dominio o modelo conceptual constituye una representación visual de los conceptos u objetos que se manejan en el dominio del sistema. Estos elementos no describen clases u objetos del software; sino entidades o conceptos del mundo real que están asociados al problema en cuestión. Permitiendo así ayudar a los usuarios, desarrolladores e interesados a utilizar un vocabulario común, para poder comprender el contexto en que se desarrolla el sistema.

Este modelo puede utilizarse además para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. (34) También puede ser tomado como el punto de partida para el diseño del sistema, por lo que es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema servirá.

2.2. Conceptos fundamentales del dominio

A continuación se describen un conjunto de conceptos asociados al dominio del problema. Facilitando un mejor entendimiento del diagrama del modelo del dominio.

Oftalmólogo: médico que se especializa en el cuidado de los ojos y el sistema visual, tanto desde su vertiente médica como desde el punto de vista quirúrgico, (4) se especializa en las estructuras, el funcionamiento del sistema visual, así como de las afecciones que este presenta. Se encarga además de

emitir un diagnóstico al paciente y de evaluar y planificar el tratamiento correspondiente a una enfermedad ocular en dependencia del diagnóstico emitido.

Diagnóstico: opinión emitida por un especialista como resultado de la observación y estudio del cuadro clínico (síntomas y signos) que presenta un paciente. Parte de la medicina que tiene por objetivo identificar una enfermedad basándose en los síntomas que presenta el paciente, el historial clínico y los exámenes complementarios (físicos, analíticos, entre otros). (35)

Paciente: término que se refiere a las personas que reciben un determinado servicio de un profesional de la salud. (36)

Tratamiento: está compuesto por indicaciones médicas y medicamentos que el médico o especialista le propone a un paciente en dependencia de la enfermedad que presente. (8)

Medicamento: conjunto de medios higiénicos, farmacológicos, quirúrgicos o físicos, los cuales tendrán como finalidad primaria la curación o el alivio de enfermedades o algunos síntomas de estas, una vez que se haya realizado el diagnóstico de las mismas. Los medicamentos sirven para prevenir, diagnosticar, tratar, aliviar o curar enfermedades. (37)

Indicación Médica: comprenden las decisiones terapéuticas que emanan del médico o especialista, las cuales deben ser respaldadas por notas y deben ser acatadas por el resto del equipo de salud. Estas indicaciones comprenden desde decisiones terapéuticas sencillas como son baño en cama, medición de constantes vitales, entre otras; hasta las más complejas como lo es la administración de medicamentos.

2.3. Diagrama del modelo de dominio

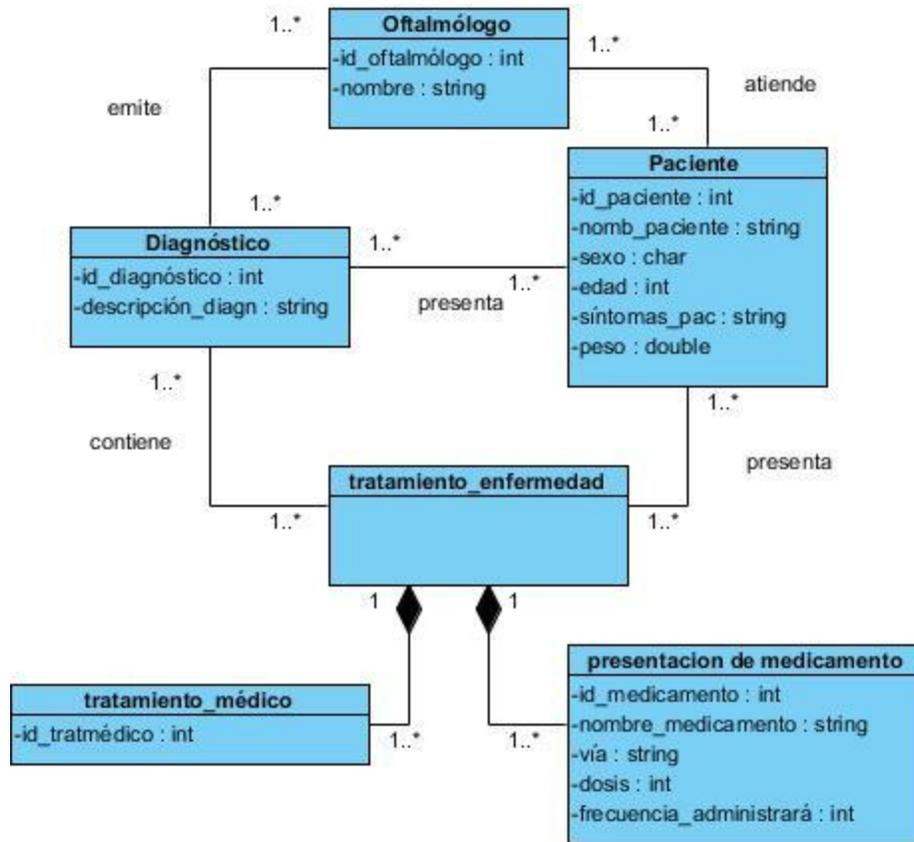


Figura 1: Modelo de dominio.

Durante el proceso de atención médica, el oftalmólogo emite un diagnóstico a un paciente en dependencia del cuadro clínico que este presenta, este diagnóstico contiene un tratamiento el cual está formado por tratamientos médicos e indicaciones de presentaciones de medicamentos.

2.4. Propuesta de sistema

El sistema alas HIS posee 17 módulos dentro de los que se encuentra Consulta Externa. En este módulo se gestiona toda la información relacionada con síntomas, antecedentes, diagnóstico y tratamiento de determinada enfermedad que posee un paciente, entre otras funcionalidades. El componente que se propone formará parte del módulo de Consulta Externa del sistema alas HIS.

Dicho componente tendrá como objetivo principal apoyar el proceso de emisión de un tratamiento médico oftalmológico a un paciente del cual ya se tiene un diagnóstico oftalmológico previamente realizado. Este será capaz de hacer una propuesta de tratamiento en dependencia del diagnóstico que se emitió con anterioridad, basándose en las GPC que frecuentemente utiliza el especialista para tratar determinada patología a partir de las mejores evidencias científicas disponibles.

2.5. Especificación de los requisitos de software

Los requisitos funcionales de un sistema son la descripción de las funciones que debe cumplir un software determinado, en términos más específicos describen lo que el sistema deberá hacer y el comportamiento que tendrá. Existen 2 tipos de requisitos de software: requisitos funcionales y requisitos no funcionales. A continuación se presentan las características y funcionalidades con las que contará dicho componente a partir de los requisitos funcionales y no funcionales encontrados.

2.5.1. Requisitos funcionales:

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera que este debe reaccionar a entradas particulares y de cómo se debe comportar en estas situaciones. Surgen a partir del análisis en profundidad de la situación problemática. (39) Son capacidades o condiciones que el sistema debe cumplir, también se puede decir que son las tareas que debe de realizar el sistema para satisfacer las necesidades del usuario. A continuación se muestran los requisitos funcionales identificados según la jerarquía que tendrán en el sistema:

RF1: Gestionar Tratamiento de GPC.

RF1.1: Crear tratamiento de GPC.

RF1.2: Ver detalles de tratamiento de GPC.

RF1.3: Modificar tratamiento de GPC.

RF1.4: Buscar tratamiento de GPC.

RF1.5: Ver datos de Tratamiento de GPC.

RF1.6: Eliminar tratamiento de GPC.

RF 2: Gestionar Indicación Médica.

RF2.1: Crear indicación médica.

RF2.2: Ver detalles de indicación médica.

RF2.3: Modificar indicación médica.

RF2.4: Buscar indicación médica.

RF2.5: Eliminar indicación médica.

RF 3: Emitir Tratamiento Médico.

2.5.2. Requisitos no funcionales:

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Estos a menudo se aplican al sistema en su totalidad, por lo que apenas se aplican a características o servicios individuales del sistema. Como su nombre sugiere, son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Estos requerimientos responden a las condiciones que debe cumplir el sistema para satisfacer el contrato o especificación, por lo que se deben satisfacer las necesidades del usuario

2.5.2.1. Usabilidad

La usabilidad se refiere a la capacidad de un software de ser comprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso; pero éste no depende sólo del producto, sino también del usuario, de la facilidad con que este interactúe con la herramienta, en menos pasos o más naturales a su formación específica.

De acuerdo a estas condiciones: el componente estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido, con una previa preparación. La navegabilidad interna no debe ser compleja, lo que hace que el acceso a las diferentes funcionalidades sea rápido.

2.5.2.2. Rendimiento

El componente deberá minimizar el volumen de datos en las peticiones y optimizar el uso de recursos críticos como la memoria RAM. Se respetarán además las buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual de Java, como la creación de objetos.

2.5.2.3. Portabilidad

La aplicación podrá ser desplegada sobre los Sistemas operativos:

Linux: Oracle Linux 5.5+, 6.x, Ubuntu 8.04 LTS (Long Term Support) Desktop Edition, Ubuntu Linux 10.04 y superior, Red Hat Enterprise Linux 5.5+, 6.x, Suse Linux Enterprise Server 10 SP2, 11.x, entre otras versiones.

Windows: 2000, XP, Server 2003, Server 2008, Server 2012, Vista, 7, 8.

2.5.2.4. Seguridad

La seguridad se basa en garantizar la confidencialidad, integridad y disponibilidad de los datos presentes en el sistema. Como es conocido, resulta prácticamente imposible mantener un sistema seguro al 100 %, por lo que este es un aspecto que debe vigilarse constantemente, debido a que pueden ocurrir riesgos que resulten significativos para el sistema.

El componente propuesto deberá mantener seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Por cada usuario del sistema

Todas las acciones realizadas, se registrarán, llevando el control de las actividades de cada usuario en todo momento, proporcionando un registro de actividades (log) de cada usuario en el sistema. Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la base de datos, y permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

2.5.2.5. Software

El componente formará parte del sistema alas HIS, podrá ser desplegado tanto en sistemas operativos Windows como Linux, utilizando la máquina virtual de Java (JVM-Java Enterprise Edition), JBoss como servidor de aplicaciones y PostgreSQL como gestor de base de datos. Los usuarios deberán disponer de un navegador web, Firefox 3.6, Google Chrome 14 o versiones superiores de ellos.

2.5.2.6. Interfaz

Los datos de las interfaces del componente contendrán datos legibles y bien estructurados, permitiendo además la correcta interpretación de la información. La entrada de datos incorrecta será detectada e informada al usuario.

2.5.2.7. Hardware

Estaciones de trabajo

Las estaciones de trabajo deberán ser de 1GB de memoria RAM y un microprocesador Intel® Core-2 Duo o Intel® Dual-Core, con sistema operativo Windows o Linux.

Servidores

Servidor de Base de Datos: PowerEdge R910, Procesador Intel® Xeon® CPU E7- 8837 @ 2.67GHz, 16GB de memoria RAM, 1199GB de disco duro y sistema operativo Linux.

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

2.6. Modelo de casos de uso del sistema

El modelo de casos de uso describe las funcionalidades propuestas del sistema a realizar, por lo que estas serán expresadas como casos de uso. Este modelo documenta el comportamiento del sistema, desde el punto de vista del usuario, permitiendo representar las funciones que se desean en el sistema a partir de casos de uso, representados por actores, y las relaciones existentes entre ellos.

Un actor es un rol que un usuario juega con respecto al sistema. El uso de la palabra rol indica más bien la labor que este realiza frente al sistema, y pueden ser desempeñados por una o varias personas, un equipo o un sistema automatizado. Los actores representan elementos que interactúan con el propio sistema.

Los actores del sistema identificados son:

Actor	Descripción
Usuario	Usuario global que permite la autenticación en el sistema, el sistema lo valida mediante un rol específico.
Administrador del Sistema	Es el encargado de realizar la inserción de las GPC.
Médico	Se encarga de emitir un diagnóstico, y un tratamiento para dicho diagnóstico a un paciente para una especialidad específica.
Oftalmólogo	Es el encargado de emitir un diagnóstico y un tratamiento oftalmológico en dependencia de la enfermedad ocular diagnosticada.

Tabla 1: Actores del sistema.

Definición de actores del sistema

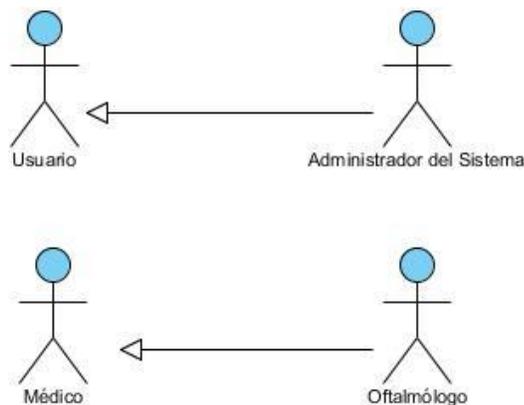


Figura 2: Diagrama de actores.

Diagrama de casos de uso del sistema

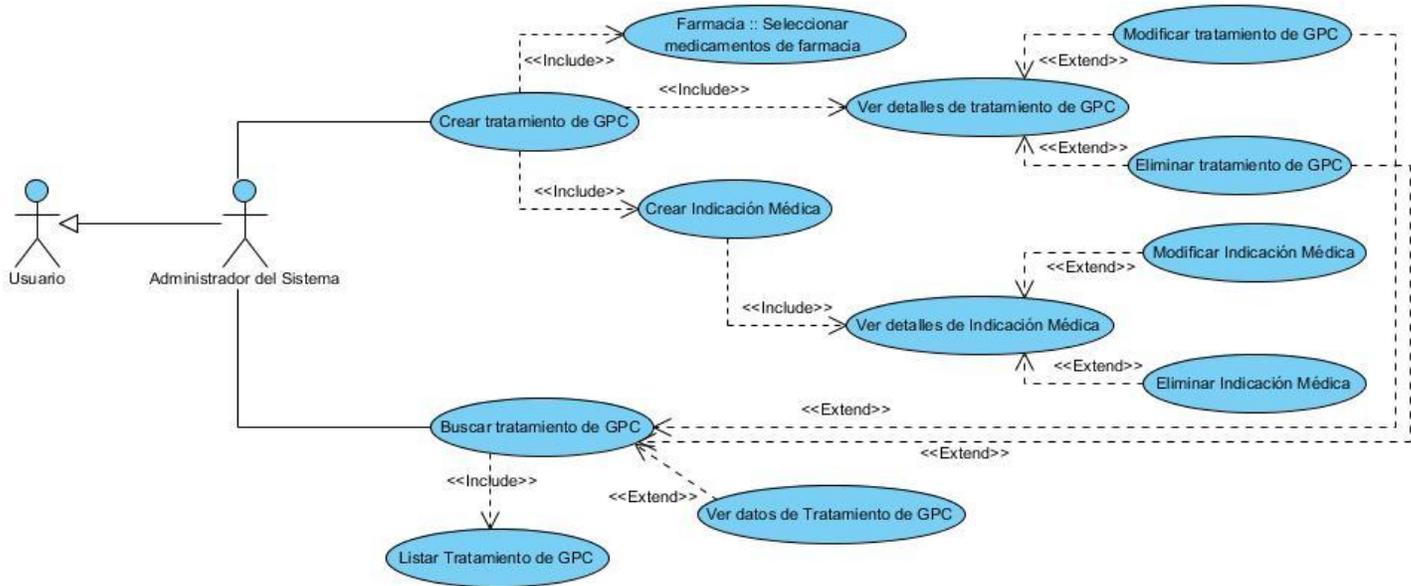


Figura 3: Diagrama de casos de uso del sistema: *Gestionar Tratamiento de GPC*.

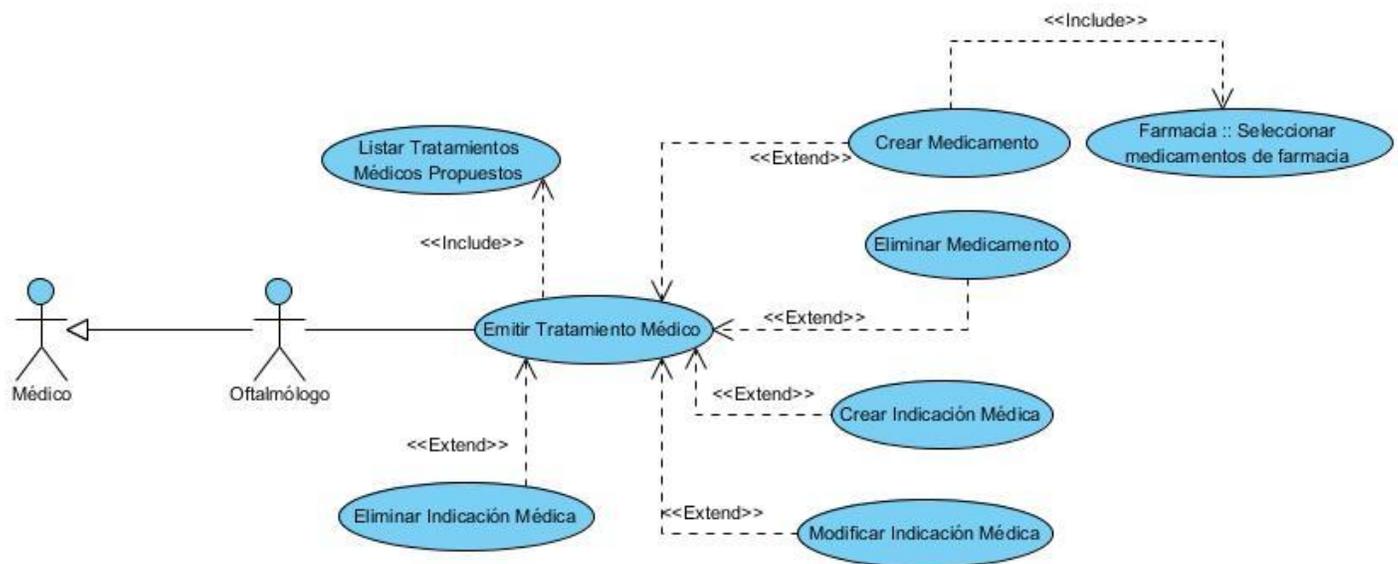


Figura 4: Diagrama de casos de uso del sistema: *Emitir Tratamiento*.

2.7. Descripción textual de los casos de uso del sistema

Con la descripción textual de un caso de uso se describe el flujo de actividades o procesos que serán objeto de automatización en el mismo, incluyendo además la forma de interacción de los actores con cada uno de ellos.

Crear Tratamiento de GPC.	
Actores:	Administrador del sistema.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear Tratamiento de GPC, el sistema brinda la posibilidad de introducir y seleccionar los datos para crear el tratamiento de GPC, el actor introduce y selecciona los datos del tratamiento, el sistema crea el tratamiento de GPC, el caso de uso termina.
Referencias:	RF1.1
Precondiciones:	El paciente debe haber sido diagnosticado con anterioridad.
Poscondiciones:	Se creó un tratamiento de GPC.

Tabla 2: Descripción textual del caso de uso: *Crear Tratamiento de GPC.*

Crear indicación médica	
Actores:	Administrador del sistema.
Resumen:	El caso de uso inicia cuando el actor accede a la opción Adicionar indicaciones médicas, el sistema brinda la posibilidad de introducir los datos para crear o adicionar la indicación médica, el actor introduce los datos de la indicación médica, el sistema crea la indicación médica, el caso de uso termina.
Referencias:	RF2.1

Precondiciones:	No existen.
Poscondiciones:	Se creó una indicación médica

Tabla 3: Descripción textual del caso de uso: *Crear indicación médica.*

Emitir Tratamiento Médico	
Actores:	Oftalmólogo
Resumen:	El caso de uso comienza cuando el oftalmólogo accede a la opción crear indicaciones médicas en el ver opciones de la hoja de consulta del paciente, el sistema muestra un listado de los tratamientos emitidos anteriormente en dependencia del diagnóstico realizado, el actor selecciona el tratamiento que desea emitir, el sistema permite realizar acciones como modificar, adicionar y eliminar las presentaciones de medicamentos e indicaciones médicas del tratamiento seleccionado, el sistema emite el tratamiento, el caso de uso termina.
Referencias:	RF3
Precondiciones:	El paciente debe haber sido diagnosticado con anterioridad.
Poscondiciones:	Se emitió un tratamiento médico.

Tabla 4: Descripción textual del caso de uso: *Emitir Tratamiento Médico.*

Como parte del desarrollo del presente capítulo se obtuvo una descripción detallada de la solución propuesta para el componente que se pretende realizar. Como resultado de un estricto análisis se definió un modelo de dominio para dicho componente en el cual se abarcaron las definiciones asociadas al dominio de la aplicación y las relaciones existentes entre ellas. Por otra parte se definieron los requisitos funcionales y no funcionales del sistema y los casos de uso que los implementan, los cuales quedaron agrupados en el diagrama de casos de uso del sistema.

Capítulo 3 Análisis y diseño del sistema

En el presente capítulo se realiza la descripción del diseño del componente propuesto. Se muestra la justificación del uso de los patrones para su construcción, así como los diagramas de clases del diseño correspondientes a dicho modelo.

3.1. Descripción de la arquitectura, fundamentación

El concepto de arquitectura de software se refiere a la estructuración del sistema, la cual debe realizarse desde etapas tempranas de su desarrollo. Dicha estructuración representa un diseño de alto nivel del sistema el cual busca satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo del mismo. En caso de que este diseño no se realice desde etapas tempranas del desarrollo del sistema, puede ocurrir que el producto final no satisfaga las necesidades de los clientes, por lo que el costo de incidir en correcciones relacionadas con problemas en la arquitectura a causa de un diseño retardado es muy elevado.

En otros términos la arquitectura de software o también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

Fundamentación del uso de patrones

En términos generales un patrón puede definirse como un modelo a seguir para la realización de alguna actividad. En dependencias del sistema que se desee implementar existen muchos tipos de patrones. Para el desarrollo del sistema propuesto serán aplicados los patrones de arquitectura y diseño que se detallan a continuación.

Patrones de arquitectura

Para el desarrollo del sistema propuesto y teniendo en cuenta las herramientas, tecnologías y metodologías propuestas se define como parte de la línea base de la arquitectura, la implementación del patrón Modelo Vista Controlador.

Dicho patrón es muy usado en aplicaciones web y permite la separación de los datos de una aplicación, la interfaz de usuario y la lógica del negocio, en tres componentes distintos: el modelo, donde se encuentran

los datos; la vista, que muestra la información del modelo al usuario; y el controlador, que gestiona las entradas del usuario; agrupando cada uno de estos componentes en 3 capas fundamentales: presentación, negocio y acceso a datos; esta separación posibilita cierta independencia a la hora de realizar cambios en dichos componentes sin que se afecten los demás o en cualquier otro caso, todo el sistema.

El uso del framework JSF, permite contar con la implementación de este patrón, permitiendo que la aplicación posibilite tener una separación clara entre cómo se muestra la información al usuario, cómo se manejan las acciones que el usuario desea realizar sobre el sistema y cómo se realizan estas acciones modificando y validando la información.

La vista traducida como la capa de presentación está conformada principalmente por páginas xhtml, las cuales son interfaces de usuario, que manejan las acciones realizadas sobre la interfaz por el usuario y recogen la información entrada por el mismo. El controlador se corresponde con la capa de negocio, dicha capa está conformada por las clases controladoras, las cuales se encargan del procesamiento de la información en correspondencia con el negocio en cuestión.

La información que es manejada en todo el sistema coincide con el modelo, y es una representación orientada a objetos, de las tablas de la base de datos del sistema, en forma de clases de entidad.

Patrones de diseño

En el desarrollo del componente de propuesta de tratamiento se emplean los patrones generales para la asignación de responsabilidades, comúnmente conocidos como GRASP. A continuación se muestran las características de los patrones empleados:

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: 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.

Su empleo se evidencia en la asignación de responsabilidades asignando a cada clase las responsabilidades o tareas que pueden realizar según la información que poseen, poniéndose de manifiesto los patrones *Experto* y *Creador*. Logrando así conservar el encapsulamiento ya que los objetos realizan lo que se les pide a partir de la información que poseen.

Por su parte el empleo de los patrones *Bajo acoplamiento* y *Alta cohesión*, se pone de manifiesto cuando se modelan las clases controladoras, encargadas de realizar las operaciones del sistema, el diseño de la aplicación permite la interacción entre las mismas sin afectar su reutilización y su correcto funcionamiento por separado.

3.2. Estrategias de integración

Con el objetivo de reducir tiempo, minimizar las redundancias y aprovechar el trabajo realizado anteriormente, la práctica de reutilización de código fuente trae consigo un conjunto de ventajas para toda aplicación que se encuentre en desarrollo. La creación de componentes reutilizables para evitar la duplicidad del mismo es la forma más eficiente de esta reutilización.

En el sistema alas HIS se utilizan varios componentes comunes en todos los módulos de dicho sistema, con el fin de lograr uniformidad en el desarrollo y mejorar la calidad del trabajo.

Componentes como la clase *ActiveModule* con el objetivo de conocer en qué módulo y entidad se encuentra el usuario que está utilizando el sistema, la clase *Bitácora* para tener el control de las trazas de todas las acciones que se realizan con la aplicación y la clase *User* para saber qué usuario está trabajando con la aplicación en tiempo real; se emplean en la reutilización de código.

El componente para la propuesta de tratamiento a pacientes de la especialidad de oftalmología del sistema alas HIS podrá ser reutilizado en cada uno de los módulos de dicho sistema en dependencia de la

especialidad para la cual el paciente requiera el tratamiento, con solo añadir las GPC correspondientes a dicha especialidad

3.3. Modelo de diseño

Al final de la fase de elaboración y al comienzo de la fase de construcción, el diseño es el centro de atención, este tiene como objetivo traducir los requisitos a una especificación, la cual describe cómo implementar el sistema. En este caso, para que el sistema pueda ser implementado sin ambigüedades, el proceso de diseño debe ser suficientemente específico.

En el diseño, el sistema es modelado de forma tal que sea capaz de soportar todos los requisitos, tanto los funcionales como los no funcionales, así como las restricciones que se le suponen. Mediante el modelo de diseño se hace un refinamiento del proceso de análisis realizado anteriormente. Se define además la arquitectura del sistema

Diagrama de paquetes

Para elaborar el modelo de diseño y con el objetivo de dividir el sistema en fragmentos manejables para su futura implementación, se define una estructura de paquetes, empleando el criterio de empaquetamiento por proceso, siguiendo la estructura de procesos definidos en el sistema. Cada uno de ellos estará compuesto por diversos subpaquetes, los cuales a su vez contienen los diagramas de clases del diseño para cada caso de uso.

Todos estos paquetes están graficados en dependencia de la relación que guardan entre sí, utilizando el paquete repositorio de clases para su funcionamiento.

El paquete repositorio de clases está compuesto por 3 subpaquetes, entidades, sesiones y vistas. El subpaquete de las entidades contiene las clases autogeneradas especificadas en el diseño, en dependencia de las tecnologías que serán usadas en la implementación y las personalizadas. Estas clases se autogeneran desde la base de datos utilizando el ORM Hibernate (generación de objetos java desde la base de datos). Las clases personalizadas son aquellas que se modifican para una mejor gestión de la información, por lo que pueden heredar de las autogeneradas.

El de las sesiones por su parte estará conformado por las clases controladoras autogeneradas por el entorno de desarrollo, por las clases controladoras personalizadas y por las controladoras del proceso.

El paquete de las vistas en cambio estará compuesto por contenidos web referentes a las páginas clientes y los formularios que las componen, además de contener las vistas que interactúan con el usuario.

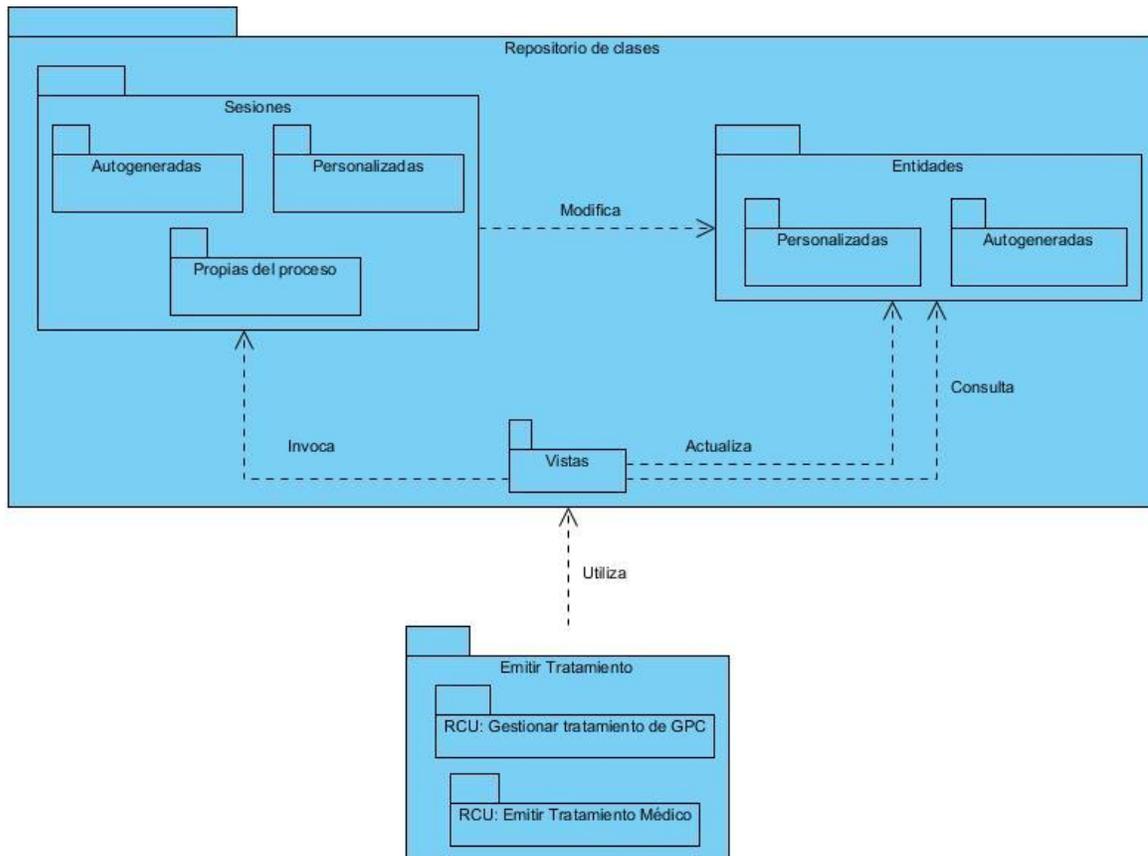


Figura 5: Diagrama de paquetes de la solución propuesta.

Diagrama de clases del diseño

Un diagrama de clases muestra un conjunto de clases que componen el sistema, interfaces y colaboraciones, así como las relaciones que existen entre ellos. Este diagrama se utiliza para modelar la vista de diseño estática y estructural de un sistema o de una parte del modelo. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

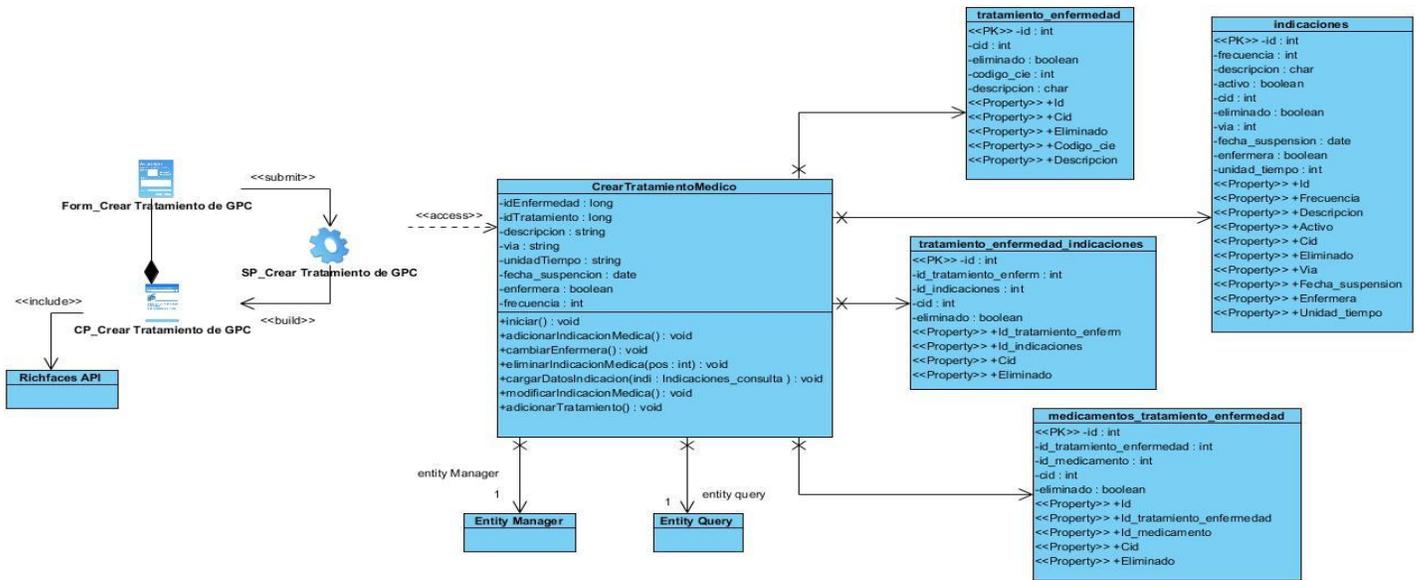


Figura 6: Diagrama de clases del diseño: *Crear Tratamiento de GPC*.

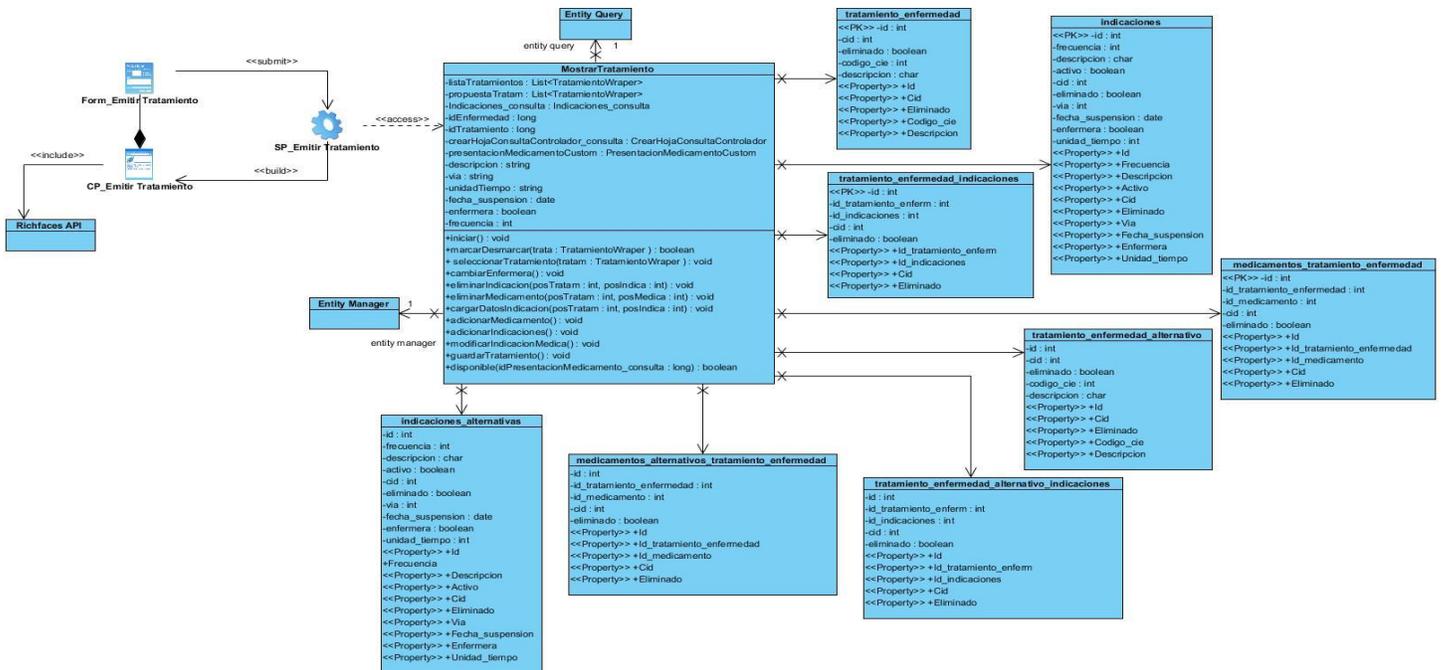


Figura 7: Diagrama de clases del diseño: *Emitir Tratamiento*.

3.4. Descripción de clases del diseño

Las clases del diseño están agrupadas en:

Páginas Clientes: Las páginas clientes o ClientPages (CP) como son llamadas comúnmente por sus siglas en inglés están compuestas por código HTML, CSS, JavaScript. Estas son interpretadas por los navegadores web presentándole al usuario la interfaz, que le permite interactuar con el sistema.

Páginas Servidoras: Son conocidas también como Server Pages (SP) por sus correspondientes siglas en inglés. Estas están compuestas por componentes Facelets, RichFaces, JSF, Seam UI, así como código HTML. Éste código es ejecutado en el servidor web, generando páginas clientes que pueden ser representadas por los navegadores web.

Formularios: Un formulario HTML es una sección de un documento enmarcado entre tags<form> y que puede contener elementos especiales llamados controles (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, entre otros.), y rótulos (labels) en esos controles. Los usuarios normalmente "completan" un formulario modificando sus controles (introduciendo texto, seleccionando objetos de un menú, etc.), y lo envían al servidor donde estos son procesados. Es una manera de obtener en el servidor información entrada por el usuario en el cliente.

Controladoras: Las clases controladoras implementan la lógica del negocio que se está informatizando, generalmente cada una se encarga de la implementación de un caso de uso o un proceso en dependencia de la complejidad de los mismos.

A continuación serán explicadas algunas de las clases identificadas para su implementación describiendo las responsabilidades que realizarán las páginas servidoras que corresponden a la lógica de negocio, con el objetivo de tener una mayor comprensión del funcionamiento que tendrá el sistema propuesto.

Nombre: MostrarTratamiento	
Tipo de clase: Controladora	
Atributo	Tipo
Nombre de cada uno de los atributos	Tipo de dato de cada uno de los atributos.

listaTratamientos	List<TratamientoWraper>
propuestaTratam	List<TratamientoWraper>
Indicaciones_consulta	Indicaciones_consulta
idEnfermedad	Long
idTratamiento	Long
crearHojaConsultaControlador_consulta	CrearHojaConsultaControlador
presentacionMedicamentoCustom	PresentacionMedicamentoCustom
descripcion	String
via	String
unidadTiempo	String
fecha_suspension	Date
enfermera	Boolean
frecuencia	integer
Para cada responsabilidad:	
Nombre:	void iniciar()
Descripción:	Recibe el id del tratamiento y el id de la enfermedad, se cargan los datos de los mismos y se inicializa todos los campos para mostrar el tratamiento.
Nombre:	boolean marcarDesmarcar(TratamientoWraper trata)
Descripción:	Permite marcar o desmarcar el tratamiento indicado al paciente.
Nombre:	void seleccionarTratamiento(TratamientoWrapertratam)
Descripción:	Guarda el tratamiento seleccionado.
Nombre:	void cambiarEnfermera()

Descripción:	Guarda el valor de la variable enfermera permitiendo conocer si el tratamiento lo realizara la enfermera.
Nombre:	void eliminarIndicacion(intposTratam,intposIndica)
Descripción:	Elimina una indicación de un determinado tratamiento.
Nombre:	void eliminarMedicamento(intposTratam,intposMedica)
Descripción:	Elimina un medicamento de un determinado tratamiento.
Nombre:	void cargarDatosIndicacion(intposTratam,intposIndica)
Descripción:	Carga los datos de una indicación específica a modificar.
Nombre:	void adicionarMedicamento()
Descripción:	Adiciona un nuevo medicamento a un tratamiento0.
Nombre:	void adicionarIndicaciones()
Descripción:	Adiciona una nueva indicación a un tratamiento.
Nombre:	void modificarIndicacionMedica()
Descripción:	Modifica una indicación de un determinado tratamiento.
Nombre:	void guardarTratamiento()
Descripción:	Guarda el tratamiento indicado.
Nombre:	boolean disponible(longidPresentacionMedicamento_consulta)
Descripción:	Muestra la disponibilidad de un medicamento en la farmacia.

Tabla 5: Descripción de la clase controladora: *MostrarTratamiento*.

Nombre: CrearTratamientoMedico	
Tipo de clase: Controladora	
Atributo	Tipo
idEnfermedad	Long
idTratamiento	Long
descripcion	String
via	String
unidadTiempo	String
fecha_suspension	Date
enfermera	Boolean
frecuencia	Integer
Para cada responsabilidad:	
Nombre:	void iniciar()
Descripción:	Recibe el id del tratamiento y el id de la enfermedad y se cargan los datos de los mismos y se inicializa todos los campos para mostrar el tratamiento.
Nombre:	void adicionarIndicacionMedica()
Descripción:	Adiciona una nueva indicación a un tratamiento.
Nombre:	void cambiarEnfermera()
Descripción:	Guarda el valor de la variable enfermera permitiendo conocer si el tratamiento lo realizara la enfermera.
Nombre:	void eliminarIndicacionMedica(int pos)
Descripción:	Elimina una indicación de un tratamiento.

Nombre:	void cargarDatosIndicacion(Indicaciones_consultaindi)
Descripción:	Carga los datos de una indicación específica a modificar.
Nombre:	void modificarIndicacionMedica()
Descripción:	Modifica una indicación de un determinado tratamiento.
Nombre:	void adicionarTratamiento()
Descripción:	Agrega un nuevo tratamiento.

Tabla 6: Descripción de la clase controladora: *CrearTratamientoMedico*.

Una vez realizado el estudio del análisis y diseño del componente a desarrollar se define la arquitectura a utilizar, justificando de esta forma el uso de patrones arquitectónicos y de diseño empleados. Se realizó además la presentación de los diagramas de clases del diseño, así como la descripción de cada uno de sus elementos con el objetivo de comprender la estructura de la solución propuesta para su posterior implementación.

Capítulo 4 Implementación

Luego de realizado el flujo de trabajo diseño, en este capítulo se procede a la realización del flujo de implementación, con el objetivo de especificar las principales características que contendrá el componente de propuesta de tratamiento en dicho aspecto. En este capítulo se realiza la elaboración del diagrama de componentes, así como el diagrama de despliegue con el objetivo de conocer la distribución física de la solución implementada, todo ello como parte del modelo de implementación.

4.1. Modelo de datos

El modelo de datos es el resultado de la traducción del análisis de los requisitos del sistema al esquema conceptual, mediante una representación gráfica de las entidades y sus relaciones. En términos generales un modelo de datos constituye la estructura o representación física de las tablas de una base de datos.

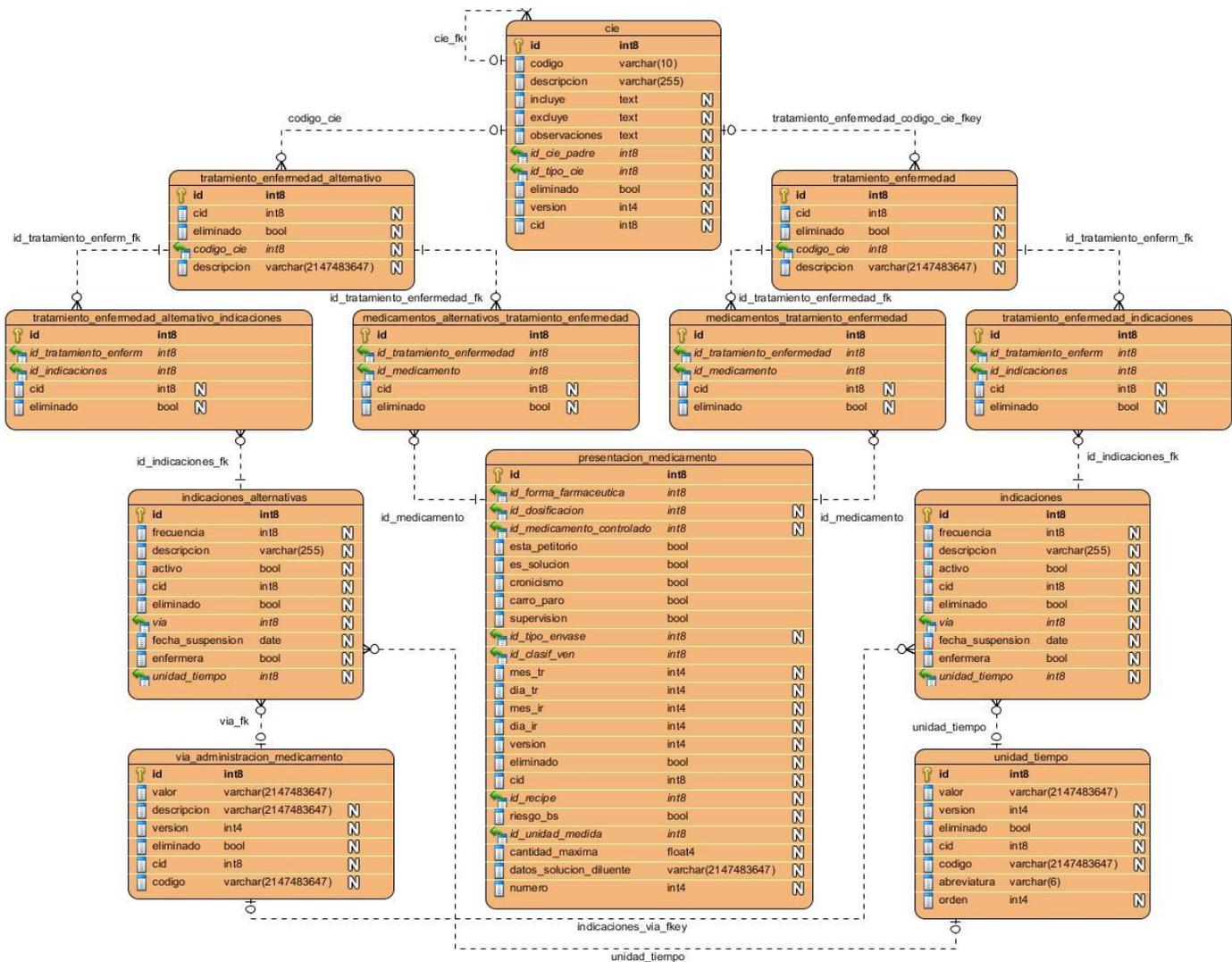


Figura 8: Modelo de datos de la solución propuesta.

4.1.1 Descripción de las tablas de la base de datos.

A continuación se describen las entidades más representativas del modelo anterior.

La siguiente tabla muestra los atributos comunes en todas las entidades para facilitar la implementación del sistema.

Atributo	Tipo	Descripción
id	Long	Identificador único para cada entidad.
versión	Long	Indica con qué versión de la entidad se está trabajando. Es usado para garantizar que se está trabajando con la versión de la entidad más actualizada que existe en la base de datos.
eliminado	Boolean	Permite la eliminación lógica con que cuenta el sistema, cuando está en verdadero indica que la entidad está eliminada.
cid	Long	Permite identificar quién realiza alguna acción sobre la entidad.

Tabla 7: Descripción de atributos comunes en las entidades.

tratamiento_enfermedad		
Entidad que recoge la información relacionada con el tratamiento.		
Atributo	Tipo	Descripción
codigo_enfermedad	Cie	Llave foránea que relaciona el tratamiento con el código de la enfermedad
descripción	String	Breve descripción del tratamiento.

Tabla 8: Descripción de la tabla: *tratamiento_enfermedad*.

indicaciones		
Entidad que recoge la información de las indicaciones médicas indicadas en un tratamiento.		
Atributo	Tipo	Descripción
frecuencia	integer	Valor que recoge la frecuencia del tratamiento
descripción	String	Breve descripción del tratamiento.
via_administracion	via_administracion_medicamento	Forma de administrar el medicamento
Fecha	Date	Duración del tratamiento
Enfermera	Boolean	Verifica si el tratamiento lo realiza la enfermera.
unidad_tiempo	Unidad_tiempo	Recoge la unidad de tiempo en que se realizara frecuencia del tratamiento

Tabla 9: Descripción de la tabla: *indicaciones*.

medicamentos_tratamiento_enfermedad
Entidad que contiene el id del medicamento asociado a un tratamiento.

Atributo	Tipo	Descripción
id_tratamiento_enfermedad	Long	Id de la entidad tratamiento_enfermedad
id_presentacion_medicamento	Long	Id de la entidad presentacion_medicamento

Tabla 10: Descripción de la tabla: *medicamentos_tratamiento_enfermedad*.

tratamiento_enfermedad_indicaciones		
Entidad que contiene el id de la indicación asociada a un tratamiento.		
Atributo	Tipo	Descripción
id_tratamiento_enfermedad	Long	Id de la entidad tratamiento_enfermedad
id_indicacion	Long	Id de la entidad indicación

Tabla 11: Descripción de la tabla: *tratamiento_enfermedad_indicaciones*.

Existen entidades llamadas nomencladores que representan los valores posibles que puede tomar un campo, ejemplo: la unidad de tiempo la cual puede ser solo uno de los valores: día(s), sem(s), hr(s); estos tipos de entidades tienen los siguientes campos, además de los antes mencionados como comunes para todas las entidades:

Nomenclador		
Atributo	Tipo	Descripción
valor	String	Valor que puede tomar el nomenclador
codigo	String	Código que permite identificar el valor del nomenclador

Tabla 12: Descripción para las tablas de nomencladores.

4.2. Modelo de implementación.

El modelo de implementación está formado por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes pueden ser encontrados datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. (40)

Su propósito es definir la organización del código, planificar las integraciones del sistema e implementar las clases y subsistemas encontrados durante el diseño. (41)

Diagrama de Componentes

Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables. Describen además la estructura de los componentes del sistema agrupados por paquetes lógicos. Siguiendo la arquitectura en capas la estructuración en subsistemas de implementación es la siguiente:

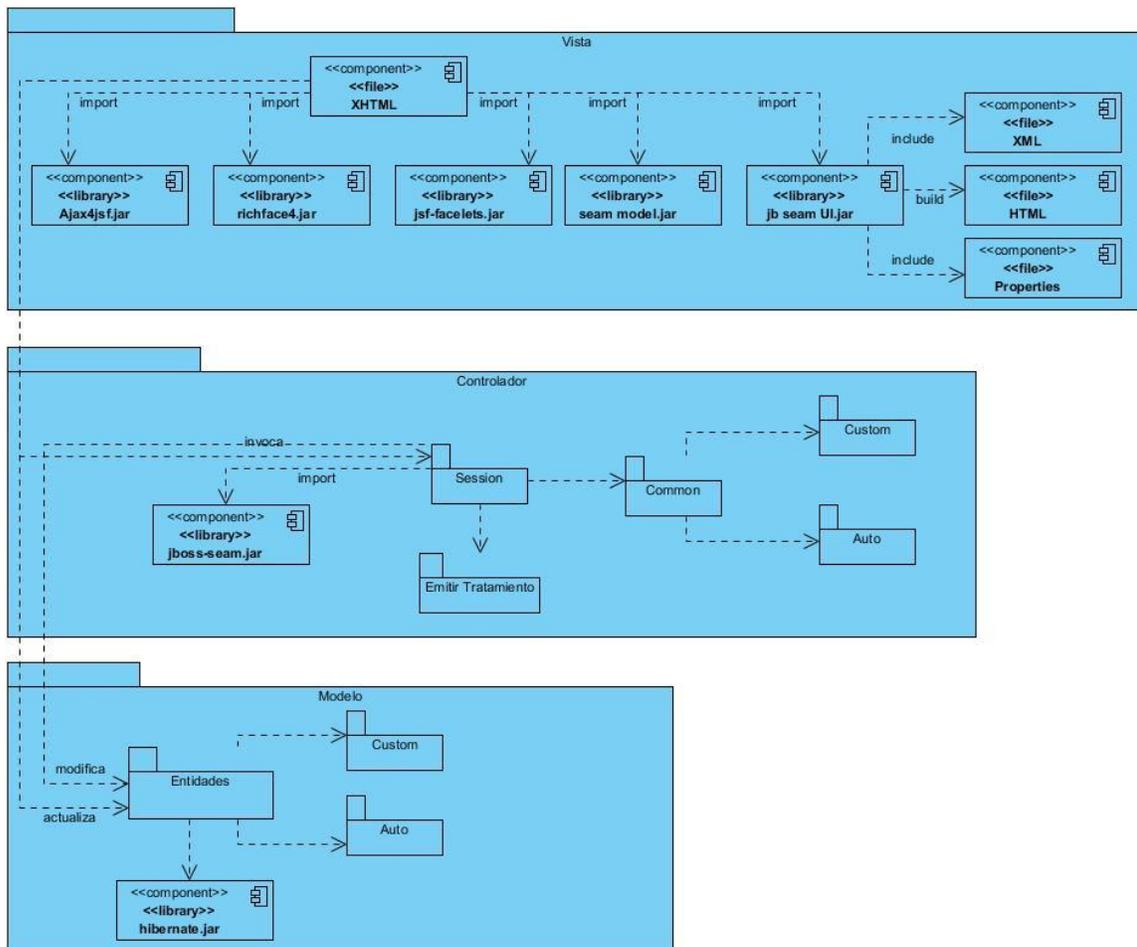


Figura 9: Subsistemas de implementación por capas.

Diagrama de Despliegue

El modelo de despliegue describe la distribución física del sistema en tiempo de ejecución. Muestran las relaciones físicas de los nodos presentes en un sistema y el reparto de los componentes sobre dichos nodos.

Un nodo representa un elemento físico que en la mayoría de las veces tiene memoria y capacidad de procesamiento.

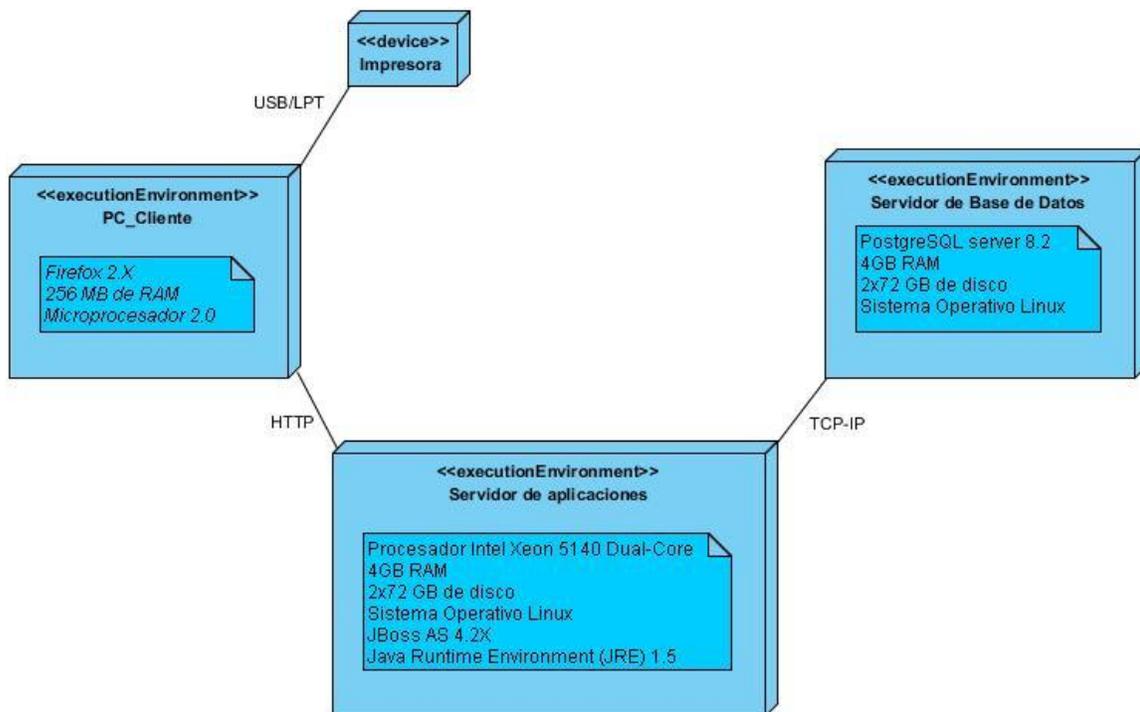


Figura 10: Diagrama de despliegue de la solución propuesta.

4.3. Tratamiento de errores.

Durante la ejecución de un sistema pueden fracasar diferentes rutinas conocidas como excepciones. Una excepción es la indicación de que ocurrió un problema durante la ejecución de un programa, de ahí que interrumpa el flujo normal de las sentencias. El tratamiento de excepciones permite restaurar un estado en el que la rutina pueda continuar con la ejecución, lo cual permite obtener un sistema más robusto y fiable.

El tratamiento de errores en el componente propuesto se realiza en las regiones críticas de código, es decir, donde los datos son insertados o modificados en la base de datos, así como en el proceso de validación de datos provenientes de la interfaz de usuario.

Para el manejo de todos ellos se utiliza el bloque try catch de la siguiente forma:

```
try
{
//declaración que causa la excepción
```

```
}  
catch (NombredeExcepcionobj)  
{  
//código para tratar el error  
}
```

En el bloque try se escribirá el código para detectar cuando ocurra algún fallo y en el catch (capturar) como su nombre lo indica, se capturarán y manejarán dichas excepciones, mostrando mensajes en la interfaz de usuario mediante el componente del frameworkSeam: FacesMessages, el cual se encarga de mostrar los mensajes que se manejan a través del objeto facesMessages inyectado en las clases controladoras, tratando a los mensajes por tipo (error, alerta y notificación).

4.4. Seguridad.

Para todo tipo de aplicación la seguridad es un elemento primordial y de suma importancia, tomando mayor relevancia cuando se gestiona información relacionada con la salud de las personas.

Debido a la gran cantidad de problemas éticos y legales relacionados con la confidencialidad y seguridad de la información que involucran los procesos que se llevan a cabo en las instituciones, es necesario tener un estricto control de la información que se maneja con el fin de garantizar que la misma no sea accedida ni modificada por personas que no tengan acceso a ella.

En el sistema alas HIS las funcionalidades encargadas de la seguridad son: iniciar y cerrar sesión de trabajo, registrar trazas y administrar seguridad.

El inicio en la sesión de trabajo, se realiza mediante el acceso al sistema a partir de la inserción del nombre de usuario y contraseña correspondientes. El sistema por su parte verifica que los datos introducidos sean válidos, por lo que el usuario tendrá acceso al módulo al que desea entrar, según los permisos otorgados. Mediante el cierre de la sesión y salida del módulo, permite la conclusión de las tareas realizadas en el sistema, así como el cierre de la sesión de trabajo.

El registro de trazas permite archivar las acciones realizadas por el usuario como: inicio o cierre de sesión, acceso a un módulo, modificación de un atributo de una entidad o cualquier otra operación. Para cada uno de estos casos la aplicación registra una traza en la base de datos.

A partir de la funcionalidad administrar seguridad se asignan los permisos a los usuarios para la navegación en el sistema, el sistema alas HIS brinda además la posibilidad de asignar o denegar permisos a roles y usuarios en las funcionalidades de los módulos.

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

Los estándares de codificación son un conjunto de reglas aplicadas por un grupo de desarrolladores con el fin de lograr uniformidad en el código de un sistema. Con su utilización no se solucionan problemas o errores existentes, más bien se evita que ocurran, logrando así la obtención de un código de alta calidad. Disminuyen además el riesgo de que los desarrolladores introduzcan errores.

Con la aplicación continua de un estándar de codificación bien definido se utilizan técnicas de programación apropiadas, y, posteriormente, se efectúan revisiones del código de rutinas, de ahí la posibilidad de que el proyecto software se convierta en un sistema fácil de comprender y mantener.

Para la solución del problema tratado en este trabajo se ha utilizado el estándar de la SUN Microsystems (Stanford University Network) para java, con la utilización de la notación Camello, para denotar variables, parámetros y métodos.

4.5.1. Variables, parámetros y métodos.

En esta notación, el identificador para las variables, los parámetros y los métodos se define escribiendo las palabras de la siguiente forma, la primera con minúsculas y a partir de la segunda palabra, en caso de existir, con letra inicial mayúscula.

A continuación se especifican algunas restricciones para la nomenclatura, basada en el estándar a utilizar:

4.5.2. Empleo de márgenes en el código.

Los márgenes serán de dos espacios por bloque de código. Los inicios y cierres ({ }) de ámbito estarán alineados debajo de la declaración a la que pertenecen, si existe solo una instrucción serán evitados. Se

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

4.5.3. Comentarios, separadores, líneas, espacios en blanco y márgenes.

Comentarios: Los comentarios se realizarán al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (tipos de datos, y objetivo del parámetro).

Líneas en blanco: Se dejará una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

Espacios en blanco: Se usarán espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto.

4.5.4. Constantes.

Apariencia de constantes: Se declararán las constantes con todas sus letras en mayúscula.

A partir del desarrollo del flujo de trabajo de implementación se obtuvieron los artefactos: modelo de datos, subsistemas de implementación y diagrama de despliegue. Se definió además la forma en la que el sistema maneja los errores y como garantiza la confidencialidad y privacidad de la información que se maneja en términos de seguridad. También se especificaron las restricciones del código, que deben ser cumplidas por los desarrolladores.

Conclusiones

Con el desarrollo del componente para la propuesta de tratamiento médico a pacientes con diagnóstico de la especialidad de oftalmología del Módulo Consulta externa del Sistema de Información Hospitalaria alas HIS, se concluye que:

- El proceso de desarrollo llevado a cabo permitió la correcta identificación de los requisitos del sistema y su cumplimiento durante el diseño y la implementación.
- El modelado de los procesos permitió definir aquellas actividades que fuesen funcionalidades del sistema.
- El análisis de los sistemas existentes demostró que en su mayoría están desarrollados bajo tecnologías privativas, por lo que no cumplen con los requisitos necesarios de integración al sistema alas HIS como parte del mismo.
- Para el desarrollo de este componente se asimiló la arquitectura propuesta por el Departamento Sistemas de Gestión Hospitalaria. El diseño propuesto y las tecnologías empleadas se basaron en dicha arquitectura.
- La solución emplea los estándares de denominación médica lo que permitirá su evolución y usabilidad internacional.
- La implementación se basó en tecnologías de desarrollo disponibles sin costo y que aseguran el cumplimiento de los requerimientos.

Recomendaciones

Por la experiencia adquirida en el desarrollo del presente trabajo de diploma, se recomienda:

El componente permita, adjuntar un documento que respalde un tratamiento, luego de haberlo creado o modificado.

El componente permita asociar dos o más diagnósticos con un tratamiento médico.

Bibliografía

1. UNAM- Facultad de Medicina, Departamento de Servicios de Cómputo. Manual de Introducción a la Informática Médica. Sistema de Información Hospitalaria. [En línea] 2003. [Citado el: 20 de Marzo de 2013.] <http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf>.
2. Slideshare. *Slideshare*. [En línea] [Citado el: 20 de Marzo de 2013.] <http://www.slideshare.net/taniagcar/la-medicina-y-sus-especialidades-14657835>.
3. Niveles de Atención de Salud. *Niveles de Atención de Salud*. [En línea] [Citado el: 20 de Marzo de 2013.] http://www.bvs.cl/php/decsws.php?tree_id=SP2.001.012&lang=es#.etrt-5877.
4. SOCIEDAD OFTALMOLÓGICA DE LA COMUNIDAD VALENCIANA . *SOCIEDAD OFTALMOLÓGICA DE LA COMUNIDAD VALENCIANA* . [En línea] [Citado el: 20 de Marzo de 2013.] <http://www.socv.org>.
5. Fisterra. *Fisterra*. [En línea] [Citado el: 20 de Marzo de 2013.] <http://www.fisterra.com/guias-clinicas/mas-sobre-guias/que-son-y-para-que-sirven-las-gpc/>.
6. **Field, MJ y Lohr , KN.** *Clinical practice guidelines: Directions for a new Agency. Institute of Medicine.* Washington D.C. : National Academic Press, 1990. pág. p 58. 7895-yeubs.
7. Diccionario Médico. *Diccionario Médico*. [En línea] [Citado el: 20 de Marzo de 2013.] http://www.portalesmedicos.com/diccionario_medico/index.php/Oftalmologia.789-yus .
8. Red Clínica. *Red Clínica*. [En línea] [Citado el: 18 de Abril de 2013.] <http://www.redclinica.cl/HospitalClinicoWebNeo/index.aspx?channel=6327&appintanceid=17806&pubid=10019>.
9. Onmeda.es Para tu salud. *Onmeda.es Para tu salud*. [En línea] [Citado el: 24 de 04 de 2013.] http://www.onmeda.es/enfermedades/enfermedades_oculares/.
10. Portales Médicos Diccionario Médico. *Portales Médicos*. [En línea] [Citado el: 5 de Abril de 2013.] http://www.portalesmedicos.com/diccionario_medico/index.php/Tratamiento.

11. **Cochrane , AL.***Efectividad y eficiencia. Reflexiones al azar sobre los servicios sanitarios.* 2.^a ed. Barcelona : Asociación Colaboración Cochrane Española y Fundación Salud, Innovación y Sociedad, 2000.
12. **Woolf , SH, y otros, y otros.***Potential benefits, limitations, and harms of clinical guidelines.* BMJ. 1999. 318:527-30.
13. **Sackett , DL, y otros, y otros.***Evidence based medicine: what is it and what it isn't.* BMJ 1996. 312: 71-2.
14. **Instituto Mexicano del Seguro Social.**
15. **Madrid MA** *Las guías reducen la variabilidad clínica más que los protocolos.* s.l. : Diario Médico, 1999 marzo 29.
16. **García Caballero , M.***¿Qué son las GPC? Diferencias con protocolos, algoritmos y vías clínicas.* Guías de práctica clínica en la asistencia médica diaria. Málaga: Universidad de Málaga : s.n., 2003. págs. 35-46.
17. **García Gutiérrez, J. F. y Bravo Toledo, R.***Guías de Práctica Clínica en Internet.* Atención Primaria, Escuela Andaluza de Salud Pública (Granada) y Área 10 de Atención Primaria. Getafe (Madrid). Madrid : Atención Primaria en la Red, 15 de Junio de 2001. pág. Vol 18 Núm 1.
18. *Calidad de las guías de práctica clínica cubanas.* **Toledo Fernández, Ana Margarita, y otros, y otros.** 2011, Revista Cubana de Salud Pública. 37(3):349-358.
19. Desarrolloweb. *Arquitectura cliente-servidor.* [En línea] Agosto de 2007. [Citado el: 20 de Marzo de 2013.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
20. **Bascón Pantoja, Ernesto.***El patrón de diseño Modelo-Vista-Controlador (MVC).* 2004. 879-asd.
21. **Sánchez González, Carlos.** Aplicaciones en capas. *Capítulo 3.* [En línea] 28 de Septiembre de 2004. [Citado el: 25 de Abril de 2013.] <http://oness.sourceforge.net/proyecto/html/ch03s02.html>.
22. **Maldonado, Daniel M.** El CoDiGo K. *Arquitectura de programación en 3 capas.* [En línea] [Citado el: 10 de Enero de 2013.] <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/..54a7-asyg>.

23. **Barello, R. M. (s.f.)**. Endesa Framework AME. [En línea] [Citado el: 10 de 01 de 2013.] <http://ame.endesa.es/confluence/display/AMEBASE/JSF.78965-asdkjd>.
24. JBoss Community. JBoss Ajax4jsf.Introducción. [En línea] 2007. [Citado el: 10 de Enero de 2013.] <http://JBossCommunity.org.5a8s7d-asd>.
25. **Garrett, Jesse James**. Maestros del web. [En línea] [Citado el: 05 de Enero de 2013.] <http://www.maestrosdelweb.com/editorial/ajax/..daisud-598>.
26. Facelets. [En línea] 29 de Agosto de 2007. [Citado el: 10 de Enero de 2013.] <http://www.google.com/notebook/public/06237054388688325750/BDSW5QgoQ38aBIMsi.4569-yaha>.
27. **Kegan, J.** 10 RichFaces Proof of Concept. [En línea] Agosto de 2008. [Citado el: 10 de Enero de 2013.] <http://www.labor.state.ny.us/cioshares/richFacesPOC.pdf.458-uaajs>.
28. **Cordovez, Juan**. fap-devel. [En línea] [Citado el: 10 de Enero de 2013.] <http://code.google.com/p/fap-devel..fap-devel>.
29. Seamframework.org. [En línea] [Citado el: 26 de Enero de 2013.] <http://www.seamframework.org/.5875-uisng>.
30. **Herrera, Cristhian**. <http://www.adictosaltrabajo.com>. [En línea] [Citado el: 8 de Enero de 2013.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=EJB3vsSpring..7855-uusji>.
31. Java Persistence API FAQ. Sun Microsystems. (s.f.). [En línea] [Citado el: 8 de Enero de 2013.] <http://java.sun.com/javaee/overview/faq/persistence.jsp.1478-iosp>.
32. **González Cornejo, José Enrique**. ¿Qué es UML? [En línea] [Citado el: 20 de 01 de 2013.] <http://www.docirs.cl/uml.htm>.
33. Descripción de Eclipse SDK 3.3.2. [En línea] [Citado el: 23 de Enero de 2013.] <https://www.ohloh.net/p/pgadmin>.
34. [En línea] [Citado el: 22 de Enero de 2013.] <http://arley triana.blogspot.com/2009/07/implementacion-del-patron-clasico-de.html.iqhewir-5497>.
35. PostgreSQL-Investigacion. [En línea] [Citado el: 11 de Enero de 2013.] <http://es.scribd.com/doc/36570462/postgreSQL-investigacion.aseqw-7854>.

36. **Cabrera Ben, Layda de los Ángeles y Torres Amador, Jose Manuel** .*Sistema experto para el tratamiento médico de pacientes con Hipertensión Arterial y Diabetes Mellitus*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2012. 958-usbf.
37. Salud Doctissimo. *Salud Doctissimo*. [En línea] [Citado el: 23 de Marzo de 2013.] <http://salud.doctissimo.es/diccionario-medico/diagnostico.html>.
38. The Free Dictionary. *The Free Dictionary*. [En línea] [Citado el: 23 de Marzo de 2013.] <http://es.thefreedictionary.com/paciente>.
39. cedimcat. [En línea] [Citado el: 23 de Marzo de 2013.] <http://www.cedimcat.info/html/es/dir2434/doc10636.html> .
40. [En línea] [Citado el: 23 de Marzo de 2013.] <http://definicion.de/indicacion/>.
41. **Arias, Yilianne y López, Yurien**.*Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS*. UCI. La Habana : s.n., 2010. pág. 78, Tesis. 5786-yais.
42. MeRinde. *MeRinde*. [En línea] [Citado el: 2 de Mayo de 2013.] http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
43. Modelo de Implementación:Diagramas de Componentes y Despliegue. [En línea] [Citado el: 2 de Mayo de 2013.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
44. **Shu-Hsien, Liao**.*Expert system methodologies and applications—a decade review from 1995 to 2004, Expert Systems with Applications*. January 2005. págs. Pages 93-103. Vol. Volume 28. 7458-yusni.
45. *Sistema de Información de Salud Alas-HIS*. Ciudad Habana : s.n. 748-ose.
46. **Wolf, SH**.*Practice guidelines: a new reality in medicine.II. Methods of developing guidelines*. s.l. : Arch Intern Med, 1992. 152:946-52.
47. IAMBE Instituto Argentino de Medicina Basada en Evidencias. *En pos de una modalidad moderna en la asistencia médica: Medicina Basada en Evidencias*. [En línea] [Citado el: 20 de marzo de 2013.] http://www.iambe.org.ar/que_es_mbe.htm.

48. **Guerra Romero, L.** *La medicina basada en la evidencia: un intento de acercar la ciencia al arte de la práctica clínica.* s.l. : Med Clin (Barc), 1996. 107:377-82.
49. **Fors López, Dra. Martha María.** *Guía de Práctica Clínica para el manejo de episodios depresivos y trastornos depresivos recurrentes pacientes entre 18 y 64 años.* Ciudad de la Habana : s.n., 2011.

Referencias bibliográficas

1. UNAM- Facultad de Medicina, Departamento de Servicios de Cómputo. Manual de Introducción a la Informática Médica. Sistema de Información Hospitalaria. [En línea] 2003. [Citado el: 20 de Marzo de 2013.] <http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf>.
2. Slideshare. *Slideshare*. [En línea] [Citado el: 20 de Marzo de 2013.] <http://www.slideshare.net/taniagcar/la-medicina-y-sus-especialidades-14657835>.
3. Niveles de Atención de Salud. *Niveles de Atención de Salud*. [En línea] [Citado el: 20 de Marzo de 2013.] http://www.bvs.cl/php/decsws.php?tree_id=SP2.001.012&lang=es#.etr-5877.
4. SOCIEDAD OFTALMOLÓGICA DE LA COMUNIDAD VALENCIANA . *SOCIEDAD OFTALMOLÓGICA DE LA COMUNIDAD VALENCIANA* . [En línea] [Citado el: 20 de Marzo de 2013.] <http://www.socv.org>.
5. Fisterra. *Fisterra*. [En línea] [Citado el: 20 de Marzo de 2013.] <http://www.fisterra.com/guias-clinicas/mas-sobre-guias/que-son-y-para-que-sirven-las-gpc/>.
6. **Field, MJ y Lohr , KN.***Clinical practice guidelines: Directions for a new Agency. Institute of Medicine.* Washington D.C. : National Academic Press, 1990. pág. p 58. 7895-yeubs.
7. Diccionario Médico. *Diccionario Médico*. [En línea] [Citado el: 20 de Marzo de 2013.] http://www.portalesmedicos.com/diccionario_medico/index.php/Oftalmologia.789-yus .
8. Portales Médicos Diccionario Médico. *Portales Médicos*. [En línea] [Citado el: 5 de Abril de 2013.] http://www.portalesmedicos.com/diccionario_medico/index.php/Tratamiento.
9. **Cochrane , AL.***Efectividad y eficiencia. Reflexiones al azar sobre los servicios sanitarios.* 2.^a ed. Barcelona : Asociación Colaboración Cochrane Española y Fundación Salud, Innovación y Sociedad, 2000.
10. **Woolf , SH, y otros, y otros.***Potential benefits, limitations, and harms of clinical guidelines.* BMJ. 1999. 318:527-30.
11. **Sackett , DL, y otros, y otros.***Evidence based medicine: what is it and what it isn't.* BMJ 1996. 312: 71-2.

12. **Instituto Mexicano del Seguro Social.**

13. *Madrid MA Las guías reducen la variabilidad clínica más que los protocolos.* s.l. : Diario Médico, 1999 marzo 29.

14. **García Caballero , M.** *¿Qué son las GPC? Diferencias con protocolos, algoritmos y vías clínicas.* Guías de práctica clínica en la asistencia médica diaria. Málaga: Universidad de Málaga : s.n., 2003. págs. 35-46.

15. **García Gutiérrez, J. F. y Bravo Toledo, R.** *Guías de Práctica Clínica en Internet.* Atención Primaria, Escuela Andaluza de Salud Pública (Granada) y Área 10 de Atención Primaria. Getafe (Madrid). Madrid : Atención Primaria en la Red, 15 de Junio de 2001. pág. Vol 18 Núm 1.

16. *Calidad de las guías de práctica clínica cubanas.* **Toledo Fernández, Ana Margarita, y otros, y otros.** 2011, Revista Cubana de Salud Pública. 37(3):349-358.

17. Desarrolloweb. *Arquitectura cliente-servidor.* [En línea] Agosto de 2007. [Citado el: 20 de Marzo de 2013.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.

18. **Bascón Pantoja, Ernesto.** *El patrón de diseño Modelo-Vista-Controlador (MVC).* 2004. 879-asd.

19. **Sánchez González, Carlos.** *Aplicaciones en capas. Capítulo 3.* [En línea] 28 de Septiembre de 2004. [Citado el: 25 de Abril de 2013.] <http://oness.sourceforge.net/proyecto/html/ch03s02.html>.

20. **Maldonado, Daniel M.** *El CoDiGo K. Arquitectura de programación en 3 capas.* [En línea] [Citado el: 10 de Enero de 2013.] <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/..54a7-asyg>.

21. **Barello, R. M. (s.f.).** *Endesa Framework AME.* [En línea] [Citado el: 10 de 01 de 2013.] <http://ame.endesa.es/confluence/display/AMEBASE/JSF.78965-asdkjd>.

22. JBoss Community. *JBoss Ajax4jsf.Introducción.* [En línea] 2007. [Citado el: 10 de Enero de 2013.] <http://JBossCommunity.org.5a8s7d-asd>.

23. **Garrett, Jesse James.** *Maestros del web.* [En línea] [Citado el: 05 de Enero de 2013.] <http://www.maestrosdelweb.com/editorial/ajax/..daisud-598>.

24. Facelets. [En línea] 29 de Agosto de 2007. [Citado el: 10 de Enero de 2013.] <http://www.google.com/notebook/public/06237054388688325750/BDSW5QgoQ38aBIMsi>. 4569-yaha.
25. **Kegan, J.** 10 RichFaces Proof of Concept. [En línea] Agosto de 2008. [Citado el: 10 de Enero de 2013.] <http://www.labor.state.ny.us/cioshares/richFacesPOC.pdf>. 458-uajs.
26. **Cordovez, Juan.** fap-devel. [En línea] [Citado el: 10 de Enero de 2013.] <http://code.google.com/p/fap-devel/>. fap-devel.
27. Seamframework.org. [En línea] [Citado el: 26 de Enero de 2013.] <http://www.seamframework.org/5875-uisng>.
28. **Herrera, Cristhian.** <http://www.adictosaltrabajo.com>. [En línea] [Citado el: 8 de Enero de 2013.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=EJB3vsSpring..> 7855-uusji.
29. Java Persistence API FAQ. Sun Microsystems. (s.f.). [En línea] [Citado el: 8 de Enero de 2013.] <http://java.sun.com/javaee/overview/faq/persistence.jsp>. 1478-iosp.
30. **González Cornejo, José Enrique.** ¿Qué es UML? [En línea] [Citado el: 20 de 01 de 2013.] <http://www.docirs.cl/uml.htm>.
31. Descripción de Eclipse SDK 3.3.2. [En línea] [Citado el: 23 de Enero de 2013.] <https://www.ohloh.net/p/pgadmin>.
32. [En línea] [Citado el: 22 de Enero de 2013.] <http://arley triana.blogspot.com/2009/07/implementacion-del-patron-clasico-de.html>. iqhewir-5497.
33. PostgreSQL-Investigacion. [En línea] [Citado el: 11 de Enero de 2013.] <http://es.scribd.com/doc/36570462/postgreSQL-investigacion>. aseqw-7854.
34. **Cabrera Ben, Layda de los Ángeles y Torres Amador, Jose Manuel .***Sistema experto para el tratamiento médico de pacientes con Hipertensión Arterial y Diabetes Mellitus*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2012. 958-usbf.
35. Salud Doctissimo. *Salud Doctissimo*. [En línea] [Citado el: 23 de Marzo de 2013.] <http://salud.doctissimo.es/diccionario-medico/diagnostico.html>.

36. The Free Dictionary. *The Free Dictionary*. [En línea] [Citado el: 23 de Marzo de 2013.] <http://es.thefreedictionary.com/paciente>.
37. cedimcat. [En línea] [Citado el: 23 de Marzo de 2013.] <http://www.cedimcat.info/html/es/dir2434/doc10636.html> .
38. [En línea] [Citado el: 23 de Marzo de 2013.] <http://definicion.de/indicacion/>.
39. **Arias, Yilianne y López, Yurien.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS*. UCI. La Habana : s.n., 2010. pág. 78, Tesis. 5786-yais.
40. MeRinde. *MeRinde*. [En línea] [Citado el: 2 de Mayo de 2013.] http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
41. Modelo de Implementación:Diagramas de Componentes y Despliegue. [En línea] [Citado el: 2 de Mayo de 2013.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
42. **Shu-Hsien, Liao.** *Expert system methodologies and applications—a decade review from 1995 to 2004, Expert Systems with Applications*. January 2005. págs. Pages 93-103. Vol. Volume 28. 7458-yusni.
43. *Sistema de Información de Salud Alas-HIS*. Ciudad Habana : s.n. 748-ose.
44. **Woolf, SH.** *Practice guidelines: a new reality in medicine.II. Methods of developing guidelines*. s.l. : Arch Intern Med, 1992. 152:946-52.
45. IAMBE Instituto Argentino de Medicina Basada en Evidencias. *En pos de una modalidad moderna en la asistencia médica: Medicina Basada en Evidencias*. [En línea] [Citado el: 20 de marzo de 2013.] http://www.iambe.org.ar/que_es_mbe.htm.
46. **Guerra Romero, L.** *La medicina basada en la evidencia: un intento de acercar la ciencia al arte de la práctica clínica*. s.l. : Med Clin (Barc), 1996. 107:377-82.
47. **Fors López, Dra. Martha María.** *Guía de Práctica Clínica para el manejo de episodios depresivos y trastornos depresivos recurrentes pacientes entre 18 y 64 años*. Ciudad de la Habana : s.n., 2011.

Anexos

Gestionar tratamiento de GPC

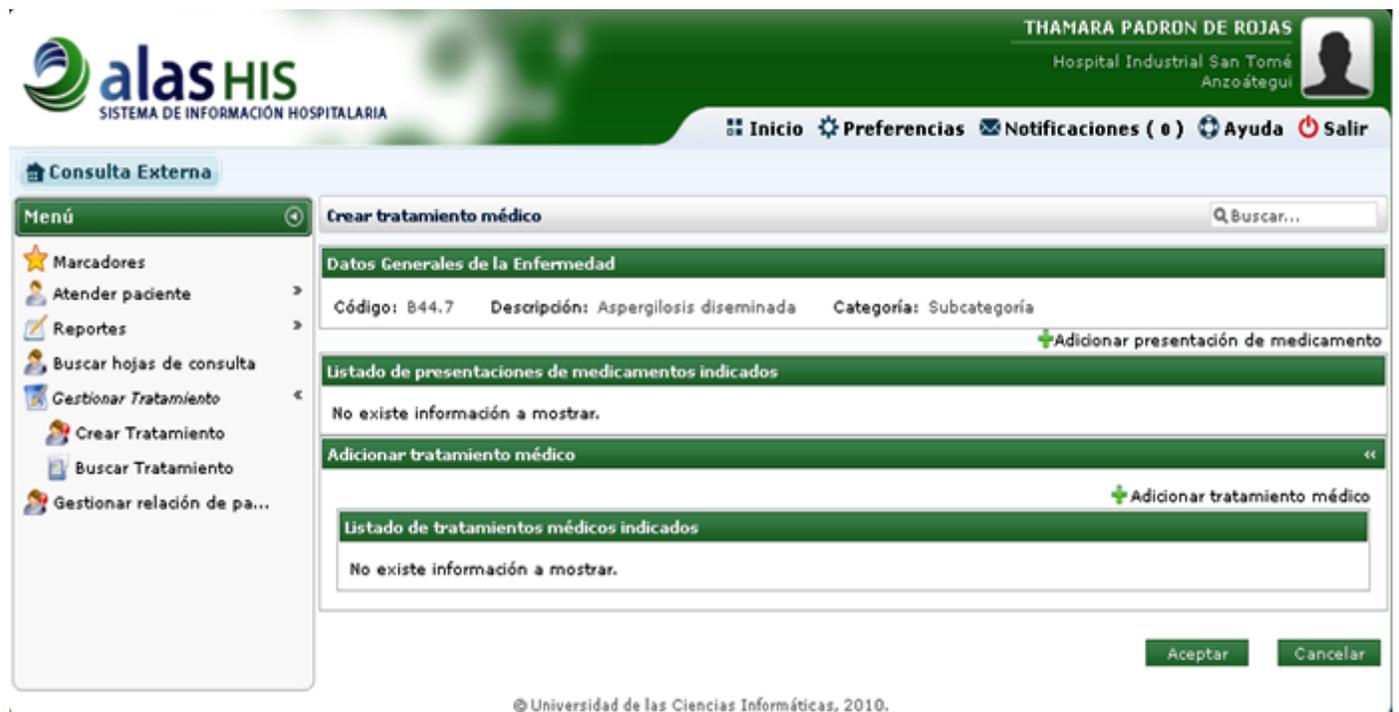


Figura 11: Crear tratamiento médico

Seleccionar presentación de medicamento Q Buscar...

Criterios de búsqueda «

Tipo récipe: Código ATC: Principio activo:

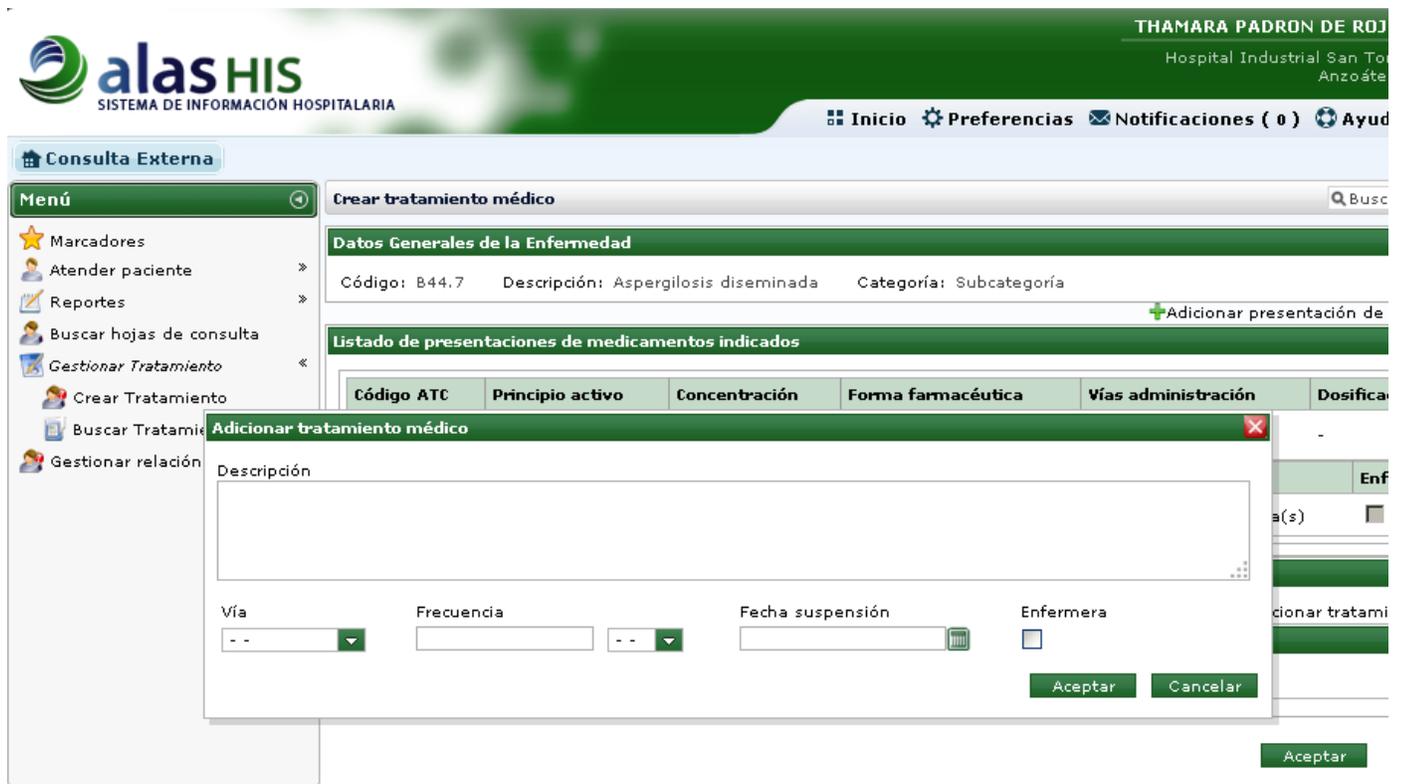
Forma farmacéutica:

Listado de presentaciones de medicamentos

	Código ATC	Principio activo	Concentración	Forma farmacéutica	Vías administración	Dosificación	Tipo récipe
<input type="checkbox"/>	L01DB01	• Doxorubicina	100.0 mg	INY	IV	-	Corriente
<input type="checkbox"/>	B01AB05	• Enoxaparina	80.0 mg/0.8ml	INY	SC	-	Corriente
<input type="checkbox"/>	A02BA02	• Ranitidina	50.0 mg/2ml	INY	IM IV	-	Corriente
<input type="checkbox"/>	N02AX02	• Tramadol	100.0 mg/2ml	INY	IV	-	Estupefacientes
<input type="checkbox"/>	R03AC12 R01AD08	• Salmeterol • Fluticasona	25.0 mcg 125.0 mcg	AEROSOL	SOLUC-INH	-	Corriente
<input type="checkbox"/>	D05AX02 D07AC01	• Calcipotriol • Betametasona	0.05 mg/g 0.5 mg/g	UNG	TOP	-	Corriente
<input type="checkbox"/>	M03AC03	• Vecuronio	4.0 mg	INY	IV	-	Corriente
<input type="checkbox"/>	N06BC01 R06AB04 N02BE01	• Cafeina • Clorfenamina • Paracetamol	32.0 mg 4.0 mg 650.0 mg	CAPS	VO	-	Corriente
<input type="checkbox"/>	J06BA02	• Inmunoglobulina humana normal, por intravascolar adm	500.0 mg	INY	IM	-	Corriente
<input type="checkbox"/>	C08DA01	• Verapamilo	40.0 mg	GRAG	VO	-	Corriente



Figura 12: Adicionar presentación de medicamento



© Universidad de las Ciencias Informáticas, 2010.

Figura 13: Adicionar tratamiento médico.

The screenshot displays the 'Buscar Tratamiento' (Search Treatment) interface within the 'alas HIS' system. The header includes the user's name 'THAMARA PADRON DE ROJAS' and the hospital 'Hospital Industrial San Tomé Anzoátegui'. The main menu on the left lists various actions like 'Atender paciente', 'Reportes', and 'Gestionar Tratamiento'. The search area features a 'Código:' dropdown menu with '<Seleccione>' selected, and buttons for 'Buscar' and 'Cancelar'. Below this is a 'Listado de tratamientos' table with columns for 'Código' and 'Descripción'. The table lists several treatment codes and their corresponding descriptions, such as 'B44.2 Tratamiento para Aspergilosis amigdalina' and '(C00-D48) Tratamiento para Tumores [neoplasias]'. Each row includes icons for document, edit, and delete. The footer contains the copyright notice '© Universidad de las Ciencias Informáticas, 2010.'

Código	Descripción			
B44.2	Tratamiento para Aspergilosis amigdalina			
B45.1	Tratamiento para Criptococosis cerebral			
B44.7	Tratamiento para Aspergilosis diseminada			
(C00-D48)	Tratamiento para Tumores [neoplasias]			
B44.9	Tratamiento para Aspergilosis, no especificada			
B46.8	Tratamiento para Otras cigomicosis			
B46.5	Tratamiento para Mucormicosis, sin otra especificación			
Z04.5a	Tratamiento para Examen de la víctima o del acusado consecutivo a otra lesión infligida			
(A00-B99)	Tratamiento para Ciertas enfermedades infecciosas y parasitarias			
(C00-D48)	Tratamiento para Tumores [neoplasias]			

Figura 14: Buscar tratamiento de GPC.

Emitir tratamiento médico

Crear indicaciones médicas Buscar...

Datos generales del paciente No.H.C.: 1.25.468



Nombre:	yamila	Cédula:	7878	Tipo de paciente:	Tesista
Primer apellido:	morales	Fecha de nacimiento:	25/02/1990	Edad:	23 año(s)
Segundo apellido:	-	Sexo:	Femenino	ABO/Rh:	-

Datos de indicación médica

Hospital: Hospital Industrial San Tomé Especialidad: Oftalmología

Indicaciones médicas:

Propuestas de Tratamientos <<

Listado de Enfermedades

Código	Descripción	Tipo	
B44.2	Tratamiento para Aspergilosis amigdalina	GPC	
B44.7	Tratamiento para Aspergilosis diseminada	GPCP	
B44.7	Tratamiento para Aspergilosis diseminada	GPCP	
B44.7	Tratamiento para Aspergilosis diseminada	GPCP	
B44.7	Tratamiento para Aspergilosis diseminada	GPCP	
B44.7	Tratamiento para Aspergilosis diseminada	GPC	

Indicación de presentaciones de medicamentos <<

[+ Adicionar presentación de medicamento](#)

Listado de presentaciones de medicamentos indicados

No existe información a mostrar.

Indicación de tratamientos médicos >>

Aceptar
Cancelar

Figura 15: Listado de propuestas de tratamiento.

Indicación de presentaciones de medicamentos <<

+ Adicionar presentación de medicamento

Listado de presentaciones de medicamentos indicados

Código ATC	Principio activo	Concentración	Forma farmacéutica	Vías administración	Dosificación
N02AX02	• Tramadol	100.0 mg/2ml	INY	IV	-
Dosis		Vía	Frecuencia	Cantidad días	Enfermera Stat
1 u		IV	2 día(s)	3 día(s)	<input type="checkbox"/> <input type="checkbox"/>
Código ATC	Principio activo	Concentración	Forma farmacéutica	Vías administración	Dosificación
A02BA02	• Ranitidina	50.0 mg/2ml	INY	IM IV	-
Dosis		Vía	Frecuencia	Cantidad días	Enfermera Stat
4 u		<Seleccione>	5 día(s)	6 día(s)	<input type="checkbox"/> <input type="checkbox"/>

Generar indicaciones médicas

Indicación de tratamientos médicos <<

+ Adicionar tratamiento médico

Listado de tratamientos médicos indicados

Descripción			
probando			
Vía	Frecuencia	Fecha suspensión	Enfermera
IV	9 día(s)	31/05/2013	<input type="checkbox"/>

Figura 16: Listado de tratamientos médicos indicados, incluye presentación de medicamentos y tratamiento médico.

alashis
SISTEMA DE INFORMACIÓN HOSPITALARIA

THAMARA PADRON DE ROJAS
Hospital Industrial San Tomé Anzoátegui

Inicio Preferencias Notificaciones (0) Ayuda Salir

Consulta Externa

Menú

- Marcadores
- Atender paciente
- Consultar relación de...
- Consultar relación de...
- Reportes
- Buscar hojas de consulta
- Gestionar Tratamiento
- Gestionar relación de pa...

Seleccionar presentación de medicamento

Criterios de búsqueda

Tipo receta: Código ATC: Principio activo:

Forma farmacéutica:

Listado de presentaciones de medicamentos

Código ATC	Principio activo	Concentración	Forma farmacéutica	Vías administración	Dosificación	Tipo receta
<input type="checkbox"/> L01DB01	• Doxorubicina	100.0 mg	INY	IV	-	Corriente
<input type="checkbox"/> B01AB05	• Enoxaparina	80.0 mg/0.8ml	INY	SC	-	Corriente
<input type="checkbox"/> A02BA02	• Ranitidina	50.0 mg/2ml	INY	IM IV	-	Corriente
<input type="checkbox"/> N02AX02	• Tramadol	100.0 mg/2ml	INY	IV	-	Estupefacientes
<input type="checkbox"/> R03AC12 <input type="checkbox"/> R01AD08	• Salmeterol • Fluticasona	25.0 mcg 125.0 mcg	AEROSOL	SOLUC-INH	-	Corriente
<input type="checkbox"/> D07AC01 <input type="checkbox"/> D05AX02	• Betametasona • Calcipotriol	0.5 mg/g 0.05 mg/g	UNG	TOP	-	Corriente
<input type="checkbox"/> M03AC03	• Vecuronio	4.0 mg	INY	IV	-	Corriente
<input type="checkbox"/> N06BC01 <input type="checkbox"/> N02BE01 <input type="checkbox"/> R06AB04	• Cafeína • Paracetamol • Clorfenamina	32.0 mg 650.0 mg 4.0 mg	CAPS	VO	-	Corriente
<input type="checkbox"/> J06BA02	• Inmunoglobulina humana normal, por intravascular adm	500.0 mg	INY	IM	-	Corriente
<input type="checkbox"/> C08DA01	• Verapamilo	40.0 mg	GRAG	VO	-	Corriente

© Universidad de las Ciencias Informáticas, 2010.

Figura 17: Seleccionar presentación de medicamento.

Indicación de tratamientos médicos

+ Adicionar tratamiento médico

Listado de tratamientos médicos indicados

Descripción			
probando			
Vía	Frecuencia	Fecha suspensión	Enfermera
IV	9 día(s)	31/05/2013	<input type="checkbox"/>

Descripción			
Vía	Frecuencia	Fecha suspensión	Enfermera
--	--		<input type="checkbox"/>

Aceptar Cancelar

Figura 18: Adicionar tratamiento médico.

Glosario de términos

Árbol taxonómico: es una estructura que permite agrupar elementos de forma jerárquica, según su parentesco, especificaciones, etc.

BSD: licencia de software otorgada principalmente para los sistemas Berkeley Software Distribution (BSD). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la General Public License (GPL: Licencia Pública General) estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Dendral: sistema experto especializado en el dominio de las estructuras moleculares y su inferencia.

Framework: también conocido como marco de trabajo. Es una estructura conceptual y tecnológica de soporte definido, con la cual, otro proyecto de software puede ser más fácilmente organizado y desarrollado.

IOM (Institute of Medicine): organización independiente que trabaja para proporcionar asesoramiento imparcial y autorizado para el público y los que toman decisiones. Fundada en 1970, forma parte de las Academias Nacionales, la cual comprende la Academia Nacional de Ciencias, la Academia Nacional de Ingeniería, el Consejo Nacional de Investigación, y el IOM.

Lisp: es un lenguaje funcional, donde cada instrucción es una descripción de la función. Se apoya en la utilización de funciones matemáticas para el control de los datos, empleando la lista como elemento fundamental.

Modularidad: término que se refiere a la capacidad de representar el conocimiento mediante reglas independientes y de propagar la evidencia mediante computaciones locales.

Red causal asociativa o semántica: constituye una representación de un universo lógico en el que el conocimiento se describe en base a relaciones en un dominio de objetos, donde los vértices representan los elementos del dominio y los arcos las relaciones entre ellos.

Shareware: término que se refiere a programas realizados generalmente por programadores independientes, aficionados o empresas pequeñas que quieren dar a conocer su trabajo permitiendo que su programa sea utilizado gratuitamente por todo aquel que desee probarlo.

Sistema experto: término que se refiere a un programa inteligente que usa el conocimiento y procedimientos de inferencia para resolver problemas que son lo suficiente difíciles como para requerir la intervención de un experto humano para su resolución.

XML: siglas de eXtensible Markup Language. Lenguaje que permite jerarquizar y estructurar la información y describir los contenidos dentro de un documento, así como la reutilización de partes del mismo.

Base de conocimientos (BC): Contiene conocimiento modelado extraído del diálogo con un experto.

Motor de inferencia: Modela el proceso de razonamiento humano.

Tecnologías Horizontales: proporcionan mecanismos que usando la familia XML, resuelven tareas genéricas tales como la interacción, la presentación, la seguridad, entre otras.