

Universidad de las Ciencias Informáticas

Centro de Informática Médica / Facultad 2



Desarrollo de la hoja de consulta de Pediatría del módulo Consulta Externa para el Sistema de Información Hospitalaria XAVIA HIS

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

Autor:

Pedro Vicente López Pérez

Tutores:

Ing. Daniellys La O Cervante

Ing. Asiel Pérez García

Co-tutor:

MsC. Maidelis Milanés Luque

La Habana, junio 2018

“Año 60 de la Revolución”



“Un niño siempre puede enseñar tres cosas a un adulto: a ponerse contento sin motivo, a estar siempre ocupado con algo y a saber exigir con todas sus fuerzas aquello que desea.”

Pablo Coelho

Declaración de autoría

Declaro que soy el único autor del trabajo con título “Desarrollo de la hoja de consulta de Pediatría del módulo Consulta Externa para el Sistema de Información Hospitalaria XAVIA HIS” y autorizo a la Universidad de las Ciencias Informáticas sus derechos patrimoniales con carácter exclusivo.

Firma del autor

Pedro Vicente López Pérez

Firma del tutor

Ing. Daniellys la O Cervante

Firma del tutor

Ing. Asiel Pérez García

DATOS DEL CONTACTO

Pedro Vicente López Pérez

Estudiante de 5to año de la carrera Ingeniería en Ciencias Informáticas. Pertenece a la Universidad de las Ciencias Informáticas. Está vinculado actualmente al Centro de Soluciones de Informática Médica donde se desempeña como programador en el Departamento Desarrollo de Aplicaciones.

Correo electrónico: pvlopez@estudiantes.uci.cu

Ing. Daniellys la O Cervante

Graduado de la Universidad de las Ciencias Informáticas en junio de 2013. Especialista B en Ciencias Informáticas perteneciente al Departamento Desarrollo de Aplicaciones del Centro de Informática Médica. Actualmente se desempeña como planificadora del proyecto Implantación del Sistema XAVIA HIS en el Centro Nacional de Cirugía de Mínimo Acceso. Tiene experiencia además en otros roles, como son analista de software y administradora de la calidad. Ha tutorado otras dos tesis que han obtenido la máxima calificación.

Correo electrónico: dlao@uci.cu

Ing. Asiel Pérez García

Graduado de la Universidad de las Ciencias Informáticas. Especialista B en Ciencias Informáticas perteneciente al Departamento Desarrollo de Aplicaciones del Centro de Informática Médica. Es uno de los desarrolladores de los módulos con los que cuenta el sistema XAVIA HIS.

Correo electrónico: asielpg@uci.cu

AGRADECIMIENTOS

Ante todo, quisiera agradecer a mi madre y mi hermana por todo el apoyo que me han dado durante todos estos años, a mi familia por ayudarme a seguir adelante en todo momento, a mis amigos por su ayuda incondicional, a los tutores por ayudarme a completar este trabajo y a todas las demás personas que hicieron posible este momento.

Pedro Vicente López Pérez

DEDICATORIA

Dedico este trabajo a las personas que me apoyaron incondicionalmente y que me dieron ánimos para completar mis cinco años de estudio en la universidad de manera satisfactoria y así llegar a este momento único en la vida.

RESUMEN

El XAVIA HIS es un sistema integrado de gestión de la información hospitalaria desarrollado en el Centro de Informática Médica de la Universidad de las Ciencias Informáticas. El sistema cuenta con el módulo Consulta Externa que gestiona todo lo referente a las consultas en un centro médico. El objetivo del presente trabajo es el desarrollo de la hoja de consulta de Pediatría del módulo Consulta Externa para el sistema XAVIA HIS para facilitar el trabajo de los médicos pediátricos, proveyéndoles un sistema que facilite la gestión de la información relacionada con los procesos de la especialidad de Pediatría dentro del área de Consulta Externa en instituciones de salud cubanas. Para el desarrollo de la solución, se identificaron los procesos asociados a esta rama de la medicina; se hizo un análisis acerca de los sistemas existentes en el mundo para identificar la existencia de los que pudieran solucionar la situación que se plantea y las tendencias que se siguen para su desarrollo, arrojando como resultado la necesidad de implementar una hoja de consulta que se adaptara al XAVIA HIS. Con la implementación de la hoja de consulta de Pediatría del módulo Consulta Externa para el sistema XAVIA HIS, se facilitará el manejo de toda la información referente a la misma, lo que posibilita el desarrollo del sistema de salud en Cuba, asegurará la integridad y seguridad de la información personal de los pacientes, lo que permitirá que la información registrada no se extravíe debido a que estará registrada de forma digital y permitirá optimizar el servicio que se le brinda al paciente, pues toda su información de salud se registrará en formato digital, estaría entendible y accesible en todo el momento de su atención.

Palabras claves: pediatría, consulta externa, hojas de consulta, XAVIA HIS.

ABSTRACT

The XAVIA HIS is an integrated hospital information management system developed at the Medical Informatics Centre of the University of Computer Science. The system has an Outpatient module that manages everything related to consultations in a medical center. The objective of this work is the development of the Pediatrics consultation sheet of the External Consult module for the XAVIA HIS system to facilitate the work of pediatric doctors, providing them with a system that facilitates the management of information related to the processes of the Pediatrics specialty within the External Consult area of Cuban health institutions. For the development of the solution, the processes associated with this branch of medicine were identified; an analysis was made of the existing systems in the world to identify the existence of those that could solve the situation that arises and the trends that are followed for its development, resulting in the need to implement a consultation sheet that is adapted to the XAVIA HIS. With the implementation of the Pediatrics consultation sheet of the External Consult module for the XAVIA HIS system, the management of all the information related to the same will be facilitated, which makes possible the development of the health system in Cuba, will ensure the integrity and security of the personal information of the patients, which will allow the registered information not to be lost due to the fact that it will be digitally registered and will allow the optimization of the service provided to the patient, since all the health information will be recorded in digital format, it will be understandable and accessible at all times of their care.

Keywords: pediatrics, external consult, consultation sheets, XAVIA HIS

ÍNDICE

ÍNDICE	9
Introducción.....	11
Capítulo I. Fundamentación Teórica.....	15
1.1 Antecedentes de la Historia Clínica Pediátrica	15
1.2 Conceptos Básicos.....	16
1.3 Tendencias actuales de los Sistemas de Gestión de la Información de Pediatría	17
1.3.1 Tendencias actuales de los Sistemas de Gestión de la Información de Pediatría en el ámbito internacional.....	17
1.3.2 Tendencias actuales de los Sistemas de Gestión de la Información de Pediatría en Cuba	18
1.4 Ambiente de desarrollo usado para la solución del problema.....	19
1.4.1 Metodología de desarrollo de software: Agil Unified Process.....	20
1.4.2 Lenguaje de modelado: Unified Modeling Language	20
1.4.4 Tecnologías utilizadas en la implementación.....	20
1.3.6 Herramientas empleadas.....	25
Conclusiones Parciales Capítulo 1	26
Capítulo 2. Características del sistema	27
2.1 Flujo actual de los procesos involucrados en el campo de acción.....	27
2.2 Objeto de automatización.....	27
2.3 Modelo de Negocio	27
2.3.1 Actores y Trabajadores del Negocio	28
2.3.2 Diagrama de procesos del negocio	28
2.4 Propuesta del sistema	30
2.5 Especificación de los requerimientos de software	31
2.5.1 Requerimientos funcionales	31
2.5.1.1 Descripción detallada de los requisitos funcionales asociados al proceso atender al paciente de pediatría.....	31
2.5.2 Requerimientos no funcionales.....	43
Conclusiones Parciales Capítulo 2	43
Capítulo 3. Diseño del Sistema, Implementación y Pruebas	45
3.1 Descripción de la Arquitectura	45
3.1.1 Arquitectura Cliente-Servidor	45
3.2.1 Patrones de Arquitectura.....	45
3.2 Modelo del Diseño del sistema.....	46

3.2.1 Diagrama de paquetes.....	48
3.2.2 Diagrama de clases del diseño	49
3.2.3 Descripción de las clases del diseño	50
3.3 Modelo de datos	51
3.4 Estrategias de integración.....	54
3.5 Implementación de las hojas de consulta.....	55
3.5.1 Estructura del modelo de implementación	55
3.5.2 Tratamiento de errores	56
3.5.3 Seguridad.....	57
3.5.4 Estrategias de codificación. Estándares y estilos a utilizar	57
3.6 Pruebas a la hoja de consulta de Pediatría	58
3.6.1 Niveles de pruebas.....	59
3.6.2 Métodos o modelos de pruebas aplicados.....	60
Conclusiones Parciales Capítulo 3	61
Conclusiones.....	63
Recomendaciones	64
Referencias Bibliográficas.....	65
Glosario de Términos.....	68
Anexos.....	69

Introducción

El sistema de salud cubano es mundialmente reconocido por su excelencia y su eficiencia. A pesar de contar con recursos sumamente limitados, Cuba ha logrado universalizar el acceso a la salud a todas las categorías de la población y conseguir resultados similares a los de las naciones más desarrolladas (Cañedo Andalia et al. 2016). Uno de los sectores beneficiados con este tipo de atención es el infantil. El cuidado de los niños siempre ha ocupado un lugar prioritario para el sistema de salud en Cuba, porque como expresara Fidel Castro Ruz: “Debemos pensar en los niños de hoy, que son el pueblo de mañana. Hay que cuidarlos y velar por ellos como los pilares con que se funda toda obra verdaderamente hermosa y útil” («Citas | Fidel soldado de las ideas» 2018).

Con el objetivo de lograr una correcta atención a los infantes y adolescentes, los pediatras confeccionan la Historia Clínica Pediátrica (HCP) que es el documento o instrumento médico-legal, en el que se registran los antecedentes biológicos, patológicos y evolutivos del niño. La correcta obtención de datos durante el proceso de confección de la HCP es un elemento esencial para elaborar un juicio diagnóstico acertado, que permita a su vez, trazar las pautas del tratamiento y dar la mejor solución posible a cada problema de salud que se enfrenta (Hernández Pérez, Romero Ruíz y Rosales Hernández 2015).

La HCP resulta de vital importancia, pues es imprescindible para el control de la salud y seguimiento de la enfermedad; para la detección precoz de problemas relacionados con crecimiento, desarrollo y nutrición; para la exploración del estilo de vida y ambiente del niño y su familia; y para la prevención de futuras dificultades (Rueda y Quevedo 2002).

El trabajo con los datos que ofrece la HCP puede realizarse a través de un Sistema de Información Hospitalaria (HIS por sus siglas en inglés) donde la misma representa una hoja de consulta que estaría compuesta por:

- Datos de Identidad General: Nombre, Primer Apellido, Segundo Apellido, Edad, Sexo, Color de la piel, Fecha de Nacimiento, Lugar de Nacimiento, Dirección Particular, Nombre del Padre, Nombre de la Madre, Consultorio.
- Historia Socio Ambiental: Datos Específicos de la Madre, Datos Específicos del Padre, Datos de la Convivencia, Datos de la Vivienda.
- Inmunizaciones: BCG, DPT, HVB, Antipoliomielítica, Antimeningococcica, PSR, Toxoide Tetánico, Antitífoidica, Dúplex.
- Intervenciones Quirúrgicas: Fecha, Tipo de Intervención, Evolución.
- Antecedentes: Embarazo, Parto, Post-Natales, Desarrollo Psicomotor y Adolescencia, Alimentación.
- Antecedentes Patológicos: Personales, Del Padre, la Madre, el Hermano, y otros.
- Resultados del Examen Físico: General, Regional y por Sistemas.

- Historia de la Enfermedad Actual: Contiene un resumen de la enfermedad que se está estudiando, así como el tratamiento recibido.
- Discusión Diagnóstica de la Evolución: Se describe la evolución que se ha producido en el paciente en cuanto a la enfermedad que se está tratando.
- Hospitalización: Resumen del Egreso, Indicaciones y Tratamiento, Estado en el momento del egreso, Lugar donde continuara el tratamiento.
- Registro de Temperaturas: Una hoja donde se registra todos los cambios de temperatura del paciente durante el tiempo de enfermedad.

En la Universidad de las Ciencias Informáticas (UCI) actualmente se desarrolla el Sistema de Información Hospitalaria XAVIA HIS, el cual se compone de varios módulos entre los que se encuentra el de Consulta Externa. Este cuenta con una hoja de consulta general que recoge todos los datos de la atención médica del paciente, pero actualmente el sistema no contempla la información específica que se registra en la HCP. La misma se debe registrar en formato duro lo que trae como consecuencia en ocasiones la duplicidad y la pérdida de información, así como la presencia de una escritura pobre, ilegible e incompleta que dificulta la interpretación de los datos que se pretenden enviar. Además, la información médica del paciente queda expuesta al riesgo de extravío o deterioro, debido a que la historia convencional en su formato de papel, sólo puede existir en un lugar y en un momento determinado, en condiciones determinadas. Asimismo, la búsqueda de información generada en diferentes consultas, por parte de los especialistas, se torna tediosa y difícil debido a que la misma se incrementa gradualmente a medida que el paciente es atendido.

Como consecuencia de la situación planteada anteriormente se define como **problema a resolver**: ¿Cómo facilitar la gestión de la información de los procesos relacionados con la especialidad de Pediatría en el área de Consulta Externa en instituciones de salud cubanas?, centrándose en el **objeto de estudio**: Proceso de gestión de información en el área de Consulta Externa en instituciones de salud cubanas; enmarcado en el **campo de acción**: Proceso de gestión de información de la especialidad de Pediatría en el área de Consulta Externa en instituciones de salud cubanas. Para resolver el problema identificado se propone el siguiente **objetivo general**: Desarrollar la hoja de consulta de la especialidad de Pediatría del módulo Consulta Externa para el sistema XAVIA HIS.

Con motivo de dar cumplimiento al objetivo trazado se han definido las siguientes **tareas de la investigación**:

1. Análisis del estado del arte relacionado con los procesos de gestión de la información en la especialidad de pediatría del área Consulta Externa a niveles nacional e internacional, para identificar la existencia de sistemas que pudieran solucionar la situación que se plantea.

2. Aplicación de la metodología, plataforma, tecnologías, librerías, herramientas y pautas definidas por la dirección del centro para el desarrollo de sus aplicaciones.
3. Validación de los artefactos que se generan durante los flujos de trabajo: Modelado de Negocio, Gestión de Requerimientos, Diseño e Implementación, que posibilitarán el desarrollo de las funcionalidades definidas para la solución en cuestión.
4. Desarrollo de las funcionalidades de la hoja de consulta de Pediatría del módulo Consulta Externa para el Sistema de Información Hospitalaria XAVIA HIS.
5. Validación de la solución para verificar el cumplimiento del objetivo propuesto.

Para desarrollar la presente investigación se utilizaron los siguientes **métodos científicos**:

Métodos teóricos:

- Analítico-Sintético: para resumir, enunciar y describir los requerimientos funcionales de la solución propuesta, los rasgos que lo caracterizan y lo distinguen. Además, ofrece la posibilidad de extraer los elementos más importantes que se relacionan con el objeto de estudio.
- Modelación: con el fin de representar los artefactos que se utilizan en el área de Consulta Externa de instituciones de salud cubanas de la especialidad de Pediatría.

Métodos empíricos:

- Observación: como instrumento para adquirir conocimiento sobre el campo de acción a través de la investigación directa de las herramientas.
- Investigación documental: para obtener información mediante la recolección y selección de documentos relacionados con la Pediatría, con la finalidad de obtener resultados que pudiesen ser base para el desarrollo de la solución.
- Entrevista: para obtener información oral de trabajadores de la Empresa de Soluciones Informáticas Softel con motivo de adquirir información sobre el sistema Galen Clínicas como parte del estudio del estado del arte.

La implementación de la solución propuesta brindará **beneficios** tales como:

- Facilitará el manejo de toda la información referente a la hoja de consulta de Pediatría, lo que posibilita el desarrollo del sistema de salud en Cuba.
- Asegurará la integridad y seguridad de la información personal de los pacientes, lo que permitirá que la información registrada no se extravíe debido a que estará registrada de forma digital.
- Permitirá optimizar el servicio que se le brinda al paciente, pues toda su información de salud se registrará en formato digital, estaría entendible y accesible en todo el momento de su atención.

Con motivo de brindar un mejor entendimiento, el documento está estructurado de la siguiente manera:

Capítulo 1 Fundamentación teórica: se explican y justifican los principales conceptos que se abordan en el documento. Se realiza un estudio del estado del arte sobre los Sistemas de Información Hospitalaria que gestionan información referente a la especialidad de Pediatría. Se presenta una explicación de las herramientas, metodología y tecnologías que serán utilizadas durante el desarrollo de la solución propuesta.

Capítulo 2 Características del Sistema: contiene un marco conceptual asociado a la información que será manipulada por el sistema, así como una breve explicación de la propuesta de solución. Se describen, además, los procesos asociados al negocio, así como las funcionalidades, y el objeto de automatización, quedando explícitamente descritos mediante casos de uso del sistema.

Capítulo 3 Diseño y Validación de la Solución: se aborda el tema relacionado con la fase de Análisis y Diseño del sistema. Como parte de la solución, se modelan los diagramas de esta etapa del ciclo de desarrollo del software, obteniéndose una modelación detallada y la construcción de la estructura de la solución propuesta. Se codifica la aplicación basado en el diseño y estándares definidos. Se valida la solución con pruebas de software y de cumplimiento del objetivo.

Capítulo I. Fundamentación Teórica

Durante la fundamentación teórica de la investigación se definieron los conceptos fundamentales asociados al desarrollo de la solución propuesta. Además, se realizó un análisis del estado del arte relacionado con los procesos de gestión de la información en la especialidad de pediatría del área Consulta Externa a niveles nacional e internacional, para identificar la existencia de sistemas que pudieran solucionar la situación que se plantea. Asimismo, se precisó el ambiente de desarrollo utilizado para la solución del problema.

1.1 Antecedentes de la Historia Clínica Pediátrica

La Historia Clínica (HC) ha ido evolucionando y adaptándose a cada momento histórico. En los orígenes de la HC se encuentran los papiros de Mesopotamia y el Antiguo Egipto que registraban los datos de fallecimientos (Laín Entralgo 1978).

A principios del siglo XX se creó una HC por paciente (Siegler, 2010), siendo a finales de 1960 cuando se ideó una forma diferente de estructurar la información en base a una lista de problemas y de ordenar las evoluciones en el contexto específico (Wedd, 1968). Posteriormente, el médico basó sus registros en lo que sus sentidos podían percibir y esto constituyó el centro del proceso de documentación. Con el descubrimiento de nuevos instrumentos de exploración y mediante la amplificación de los sentidos, el enfoque del registro en ese entonces cambió de las observaciones del paciente a las observaciones del médico. Con la aparición de los exámenes complementarios (exámenes de laboratorio, diagnóstico por imágenes u otros) se generó una diversidad de fuentes desde donde se aportaban datos a los registros médicos.

La orientación a problemas fue uno de los cambios estructurales del registro médico en su evolución histórica y originalmente fue concebida para posibilitar su informatización. Actualmente, la mayoría de los registros médicos están orientados a las fuentes que generan la información (notas de evolución, exámenes complementarios, indicaciones médicas, lista de problemas y otros) y un ordenamiento cronológico para cada una de ellas (Hernández Pérez, Romero Ruíz y Rosales Hernández 2015).

En la antigüedad la pediatría no existía como una disciplina médica, sino que la salud de los niños quedaba bajo responsabilidad de sus madres y comadronas. A partir del Renacimiento comenzó a considerarse las enfermedades de los niños como una especialidad médica y, recién en el siglo XIX, la pediatría se convirtió en una rama científica. Francia y Alemania fueron los primeros países en desarrollar los conocimientos y servicios pediátricos modernos. Los hospitales infantiles no tardaron en esparcirse por Europa y Estados Unidos, antes de extenderse hacia otras regiones del mundo (Genta Mesa 2006).

Actualmente en las organizaciones de salud existe una generación masiva de información médica. Cada día los procesos de registro, seguimiento y tratamiento del paciente deben mejorarse,

innovarse y apoyarse en tecnologías para hacer más eficiente y eficaz las actividades rutinarias del hospital, centro de salud o clínica. No basta con tener datos e información, hay que procesarla, analizarla, interpretarla y utilizarla, por ello los HIS tienen como propósito permitir la optimización de los recursos humanos y materiales para satisfacer las necesidades de las áreas operativas, administrativas, clínicas y de investigación en las organizaciones de salud.

Los principales beneficios de los HIS son:

- aumentar la eficiencia,
- mejorar la recaudación,
- mejorar la calidad de la atención al paciente.

Estos beneficios son creados por las diferentes aplicaciones e impactan en las organizaciones de diferentes maneras. Contar con documentación más rigurosa de la atención y una presentación más organizada de la misma, agiliza el ciclo de atención médica (Otero 2011).

1.2 Conceptos Básicos

Pediatría: es la rama de la medicina que se especializa en la salud y las enfermedades de los niños. Se trata de una especialidad médica que se centra en los pacientes desde el momento del nacimiento hasta la adolescencia, sin que exista un límite preciso que determine el final de su validez. El término procede del griego paidos (niño) e iatrea (curación), pero su contenido es mucho mayor que la curación de las enfermedades de los niños, ya que la pediatría estudia tanto al niño sano como al enfermo. Cronológicamente, la pediatría abarca desde el nacimiento hasta la adolescencia. Dentro de ella se distinguen varios períodos: recién nacido (0-6 días), neonato (7-29 días), lactante (lactante menor; 1-12 meses de vida, lactante mayor; 1-2 años), preescolar (2-5 años), escolar (6-12 años), pre-adolescente (10-12 años) y adolescente (12-18 años) (Gilardon et al. 2004).

Historia Clínica Pediátrica: Es el documento médico legal en el que se registra los datos de identificación y de los procesos relacionados con la atención del paciente, en forma ordenada, integrada, secuencial e inmediata de la atención que el médico y otros profesionales de la salud brindan al mismo; así como también mantiene un registro de los antecedentes biológicos, patológicos y evolutivos del niño (Turabián Fernández y Pérez Franco 2004).

Atención al paciente pediátrico: La atención al paciente pediátrico es el proceso primordial que se lleva a cabo en el área de consulta de externa de todos los hospitales pediátricos existentes en Cuba. Este proceso incluye fundamentalmente al paciente, al especialista pediatra y al encargado de los registros médicos que es quien debe hacer entrega antes de la consulta de la HCP del paciente citado (Cabello Morales 2001).

Sistema de Información Hospitalaria: Los sistemas de información hospitalaria o sistemas de información en salud (siglas: SIH, o HIS en inglés), denominado también expediente electrónico, pese a que este último es sólo la parte clínica del proceso, consiste en un programa o programas de cómputo instalados en un hospital. Estos permiten (Cerritos, Fernández y Gatica 2003):

- Llevar un control de todos los servicios prestados a los pacientes.
- Obtener estadísticas generales de los pacientes.
- Obtener datos epidemiológicos.
- Detallar el coste de la atención prestada a cada paciente.
- Llevar un estricto expediente clínico en forma electrónica.
- Facilitar el acceso y obtener los datos sobre el tratamiento del paciente de forma más segura, con prontitud y eficiente.
- Que las agencias de Salud puedan tener un control sobre las enfermedades y prevenir brotes o epidemias.
- Tener los datos más precisos en el proceso de acreditación y les facilita el acceso a los proveedores de servicios de salud.

1.3 Tendencias actuales de los Sistemas de Gestión de la Información de Pediatría

Actualmente existen en el mundo sistemas que gestionan la información médica de los pacientes pediátricos en las diferentes instituciones de salud, entre las cuales es bueno destacar por su uso y calidad: Pediatría, ExpDX, SAM Pediatría v9.2 y por parte de nuestro país se encuentra el Galen Clínicas.

1.3.1 Tendencias actuales de los Sistemas de Gestión de la Información de Pediatría en el ámbito internacional

Pediatría

Pediatría es un software de gestión de clínicas pediátricas. Todos los pacientes que acuden al centro sanitario son almacenados en fichas individuales en la base de datos del programa a través de las que se pueden ver detalles como vacunas, alimentación y patologías.

Las consultas de los pacientes son anotadas en una lista junto al motivo de las mismas y varios datos de seguimiento: peso, altura, índice de masa corporal, etcétera. Los antecedentes familiares también son comprobables mediante las fichas. Así pues, es posible llevar a cabo un completo diagnóstico de los pacientes anotando todo lo que sea importante y elaborando las oportunas recetas. Los turnos en el centro de salud son debidamente gestionados para que no se produzcan los típicos problemas al aglutinar a una elevada cifra de pacientes (Pediatría Kids 2018).

Pediatría es una aplicación de escritorio, propietaria y de pago, razones por las cuales no es compatible con la tecnología que utiliza XAVIA HIS y por tanto nunca sería posible su integración al sistema.

ExpDX

ExpDX es una aplicación web que permite guardar y consultar expedientes de los pacientes de forma segura, desde cualquier computadora, tableta electrónica, o teléfono con internet. Ofrece la oportunidad de ahorrar tiempo al buscar el historial clínico para agilizar la consulta y dar una mejor atención médica. ExpDX genera automáticamente las curvas de crecimiento para sus pacientes: Peso – Talla, Peso – Edad, Talla – Edad y Perímetro Cefálico. No requiere la instalación de ningún software o complemento para su uso. Después de preguntarle a médicos pediatras qué es lo que buscan en un expediente clínico electrónico la mayoría estuvo de acuerdo: Que ExpDX es un software para pediatras fácil de usar («ExpDX Planes» 2018).

Para el trabajo con esta aplicación resulta imprescindible la conexión a Internet, además es propietaria y de pago, esto la hace no apta para vincularla con el sistema XAVIA HIS.

SAM Pediatría V9.2

SAM Pediatría V9.2 Medical Software, forma parte del Sistema Administrativo Médico (SAM). Es un software especializado para la gestión de médicos pediatras el cual ha sido diseñado teniendo en cuenta todos los parámetros necesarios para llevar un control de sus pacientes (recién nacidos, niños, niñas y menores de edad en general) permitiéndole al especialista tener un buen funcionamiento de su clínica. SAM Pediatría permite tener un mejor soporte a la parte administrativa, pues se puede almacenar el historial y exámenes físicos de cada menor de edad y recién nacido. A través de este sistema el médico pediatra tiene la oportunidad de realizar un seguimiento exacto del crecimiento de sus pacientes mediante gráficas de peso por edad, perímetro cefálico-edad, índice de masa corporal-edad, estatura-edad, y de esta manera poder agilizar su atención a los mismos («SAM Pediatría - Sistema Administrativo Médico | Software Mas - Software para Médicos Software para Médicos» 2018).

Por sus características SAM Pediatría V9.2 representa un sistema bastante completo, pero tiene como en los casos anteriores es un software propietario, por lo que para su explotación se hace necesario comprar su licencia.

1.3.2 Tendencias actuales de los Sistemas de Gestión de la Información de Pediatría en Cuba

Galen Clínicas

Galen Clínicas es un sistema de información que permite gestionar la actividad hospitalaria del paciente e informatiza los diferentes servicios que se brindan en los centros de atención de salud,

como hospitales y clínicas. Permite mejorar la atención médica, optimizar el uso del personal, aumentar la calidad de los servicios hospitalarios y disminuir sus costos.

La solución abarca tres vistas fundamentales:

Atención al paciente: permite el registro de los datos relacionados a la atención al paciente en cada servicio, sala, consulta, etc. Para esto garantiza el acceso, con seguridad, a la información del historial de paciente y a la posibilidad de solicitud de los servicios necesarios.

Gestión de las áreas de la institución: automatiza los procesos en las áreas organizando los flujos, el registro de la información, los informes y la estadística necesaria.

Gerencia: a partir de la información relacionada con los pacientes y las áreas, emite estadística en diferentes formas (tablas, gráficas, informes) fundamentales para la toma de decisiones.

A través de entrevistas realizadas a trabajadores de la Empresa de Soluciones Informáticas SOFTEL, donde se implementó Galen Clínicas, se pudo concluir que, aunque al sistema se le puede configurar una Consulta Externa con algunos de los datos de pediatría, no contiene forma de gestionar información de específica de la especialidad de manera completa con los datos que resultan de la atención de un niño.

Para resumir el estudio del estudio del arte de estas diferentes aplicaciones de software se muestra una gráfica con los principales temas a tener en cuenta a la hora de su realización:

Tabla 1 Analisis de algunos sistemas que gestionan informacion de Pediatría

Nombre del software	Tipo de software	Tipo de licencia	De pago
Pediatría	Escritorio	Propietario	Si
SAM Pediatría	Web	Propietario	Si
ExpDX	Web	Propietario	Si
Galen Clínicas	Web	Propietario	Sí

1.4 Ambiente de desarrollo usado para la solución del problema

Para el desarrollo de la hoja de consulta de Pediatría del módulo Consulta Externa para el Sistema de Información Hospitalaria XAVIA HIS, se hace necesaria la utilización de la metodología, tecnologías y herramientas definidas por la dirección del centro para lograr la estandarización de todos sus productos.

1.4.1 Metodología de desarrollo de software: Agil Unified Process

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas. Como toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable y al no existir una metodología de software universal, se decide hacer una variación de la metodología del Proceso Unificado Ágil (AUP por sus siglas en inglés), de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Con esta adaptación de AUP que se propone se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define la Integración de modelos de madurez de capacidades para el desarrollo (CMMI-DEV por sus siglas en inglés) en su versión 1.3. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a uno la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a once («Metodología AUP-UCI.pdf» 2018).

Con el fin de lograr una mejor integración entre sus procesos, la UCI definió la utilización de la metodología AUP-UCI, por lo que para el desarrollo de la solución propuesta esta fue la que se adaptó. Específicamente XAVIA HIS está desarrollado sobre el escenario tres de esta metodología, por lo tanto, se decide utilizar este mismo en la solución de este trabajo.

1.4.2 Lenguaje de modelado: Unified Modeling Language

Para llevar a cabo el diseño de la solución se empleará el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) que se define por un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativos al desarrollo de programas informáticos. UML es una herramienta propia de personas que tienen conocimientos relativamente avanzados de programación y es frecuentemente usada por analistas funcionales (aquellos que definen qué debe hacer un programa sin entrar a escribir el código) y analistas-programadores (aquellos que, dado un problema, lo estudian y escriben el código informático para resolverlo en un lenguaje como Java, C#, Python o cualquier otro).

Asimismo se escoge el Modelo y Notación de Procesos de Negocio (BPMN, Business Process Management Notation), el cual es una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujos de negocio (workflow), puesto que el escenario seleccionado trabajo con modelado de procesos de negocio (Hitpass 2017).

1.4.4 Tecnologías utilizadas en la implementación

Lenguaje de Programación JAVA

JAVA es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados (Barnes, Kölling y Brenta 2007).

Java Server Faces (JSF)

La tecnología Java™ Server Faces (JSF) es un popular marco de Java™ para componentes de la interfaz de usuario Web. IBM® Rational® Application Developer ha incluido potentes herramientas para desarrollar aplicaciones Web JSF en las últimas versiones. La versión 7.5 ofrece a los desarrolladores una nueva funcionalidad que permite importar y usar componentes JSF arbitrarios de terceros con un mínimo esfuerzo. Este artículo describe un proceso que incluye el consumo de una biblioteca de componentes de terceros, la configuración de metadatos para la biblioteca y el uso compartido de la configuración resultante para su reutilización por parte del equipo de desarrollo («Definiciones de biblioteca Faces para controles JavaServer Faces de terceros» 2009).

Librería de componentes JSF RichFaces v3.3.1

RichFaces es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF (Atiaga et al. 2014).

Son características de RichFaces las siguientes:

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

Ajax4JSF

Es una biblioteca de código abierto que se integra totalmente en la arquitectura de JSF y hereda las funcionalidades de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir

código Javascript. Mediante este framework podemos variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, etcétera. Ajax4jsf y Richfaces permiten dotar a nuestra aplicación JSF de contenidos mucho más profesionales con muy poco esfuerzo(Bazán 2008).

Facelets v1.1

Facelets es un framework de templating que nos facilita la gestión de la estructura y el estilo de las Java Server Pages(JSP), lo cual tiene beneficios en cuanto a reutilización de código en la capa de presentación y por consecuencia eleva la mantenibilidad de esta capa al usar facelets nuestro proyecto se basará en una plantilla, que contendrá la estructura de la presentación (jsp) y las definiciones de estilo, en esta plantilla definimos zonas o áreas (como el header, menú, espacios comunes) y el motor de facelets se encargará de llenar esas áreas con el contenido apropiado, así por ejemplo no tendremos el header en todas las JSP y si después de un tiempo algo tiene que cambiar se hará solo en la plantilla en lugar de hacerlo en cada página («JavaServer Facelets» 2018).

XHTML

El Extensive Markup Language (XML) es un lenguaje de programación que deriva del lenguaje de marcado generalizado estándar o SGML (por sus siglas en inglés de Standard Generalized Markup Lenguaje) y se usa para el almacenamiento de datos. Este lenguaje (o metalenguaje) posibilita la definición de la gramática de diversos lenguajes, contribuyendo a la estructuración de la información y brindando soporte a las bases de datos.

Lo que hace el XHTML (Extensive Hypertext Markup Language) es volver compatibles ambos lenguajes (el HTML y el XML). De esta forma, mantiene la mayoría de las características del HTML, aunque con elementos del XML. El XHTML, por lo tanto, otorga mayor robustez y capacidad de adaptación, resultando importante para el desarrollo de la Web 3.0 gracias a su tipo de codificación. Cascading Style Sheets (CSS).

Las hojas de estilo en cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Lo que se persigue con el desarrollo de CSS es separar la estructura de un documento de su presentación(Cordero 2014).

JBoss Seam v2.1.1

Es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB), funcionando, según versa su significado en español, como una “costura” entre estos componentes. Seam también se integra perfectamente con otros frameworks como:

RichFaces, ICE Faces, MyFaces, Hibernate y Spring. Seam es una potente plataforma de desarrollo de código abierto para construir aplicaciones ricas de Internet en Java. Seam integra tecnologías como JavaScript asíncrono y XML (AJAX), JSF, JPA, Enterprise Java Beans y BPM. Seam elimina la capa artificial que existe entre Enterprise Java Beans v3.0 y JSF y provee un consistente sistema de anotaciones para integrar estos dos frameworks. Comparada con aplicaciones desarrolladas en otros frameworks, las aplicaciones Seam son conceptualmente simples y requieren significativamente menos código (en Java y en XML) para obtener las mismas funcionalidades («Seam - Contextual Components» 2018).

Hibernate v3.3

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones. Proporciona un lenguaje de consulta para acceder a objetos, proporcionar almacenamiento en caché y el apoyo JMX (Java Management Extensions). Es destinado a proporcionar la persistencia de alto rendimiento con bajos recursos (Halder, Jana y Cortesi 2016).

Enterprise Java Beans (EJB3) v3.0

Enterprise JavaBeans (EJB) es una arquitectura que permite la creación de componentes de aplicaciones distribuidas y orientadas a transacciones. Las aplicaciones escritas utilizando EJB son escalables, transaccionales y multiusuarios.

Las características esenciales de EJB son:

- Contiene la lógica del negocio que opera con el Enterprise Information System (EIS).
- Las instancias son creadas y manejadas por el container EJB.
- Puede ser configurado editando sus parámetros de entorno vía archivos XML.
- Las características de seguridad y transacciones se encuentran separadas de las clases EJB, lo que permite la operación de aplicaciones externas y middlewares.

Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue su diseño es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sucedía con EJB2, y permitir usar objetos regulares conocidos como POJOs (Plain Old Java Object)(«Enterprise JavaBeans» 2018).

Java Persistence API (JPA)

Proporciona un mecanismo para gestionar la persistencia y la correlación y funciones relacionales de objetos para las especificaciones EJB 3.0 y EJB 3.1.

La especificación JPA define la correlación relacional de objetos internamente, en lugar de basarse en implementaciones de correlación específicas del proveedor. JPA se basa en el modelo

de programación Java que se aplica a los entornos Java EE, pero JPA puede funcionar dentro de un entorno Java SE para probar funciones de la aplicación.

JPA representa una simplificación del modelo de programación de persistencia. La especificación JPA define explícitamente la correlación relacional de objetos, en lugar de basarse en implementaciones de correlación específicas del proveedor. JPA crea un estándar para la importante tarea de la correlación relacional de objetos mediante la utilización de anotaciones o XML para correlacionar objetos con una o más tablas de una base de datos. Para simplificar aún más el modelo de programación de persistencia:

- La API EntityManager puede persistir, actualizar, recuperar o eliminar objetos de una base de datos
- La API EntityManager y los metadatos de correlación relacional de objetos manejan la mayoría de las operaciones de base de datos sin que sea necesario que se escriba código JDBC o SQL para mantener la persistencia

JPA proporciona un lenguaje de consulta, ampliando el lenguaje de consulta EJB independiente (también denominado JPQL), que se puede utilizar para recuperar objetos sin escribir consultas SQL específicas de la base de datos con la que está trabajando («IBM Knowledge Center - Arquitectura JPA (Java Persistence API)» 2018).

Java Platform Enterprise Edition v5.0

Java Platform, Enterprise Edition (Java EE) se basa en la especificación Java SE. Representa una colaboración entre diversos proveedores y líderes del sector y proporciona el soporte de infraestructura para las aplicaciones.

En la infraestructura Java EE, añade reglas:

- En la capa de la aplicación, para gestionar lógica empresarial dinámica y el flujo de tareas.
- En la capa de la presentación, para personalizar el flujo de páginas y el flujo de trabajo, y para construir páginas personalizadas basadas en estado de sesión.

Java EE es portable y escalable, y da soporte a la integración con versiones anteriores y componentes basados en arquitectura EJB. Java EE simplifica las aplicaciones empresariales definiendo y especificando un complejo conjunto de servicios estándar comunes, como denominación, gestión de transacciones, simultaneidad, seguridad y acceso a base de datos.

Java EE define un modelo de contenedor, que aloja y gestiona instancias de componentes de aplicaciones Java EE. Los contenedores están a su vez alojados en servidores Java EE («IBM Knowledge Center - Aplicaciones Java SE y Java EE» 2018).

Java Runtime Environment (JRE) v1.6

El JRE (Java Runtime Environment) es una máquina virtual de Java y su función es hacer de intermediario entre una aplicación programada en Java y el sistema operativo que se esté usando. De este modo, cualquier aplicación puede funcionar en cualquier sistema operativo que disponga del JRE. Es lo que se obtiene al descargar el software de Java, está formado por Java Virtual Machine (JVM), clases del núcleo de la plataforma Java y bibliotecas de la plataforma Java de soporte. JRE es la parte de tiempo de ejecución del software de Java, que es todo lo que necesita para ejecutarlo en el explorador web («JRE (Java Runtime Environment)» 2018).

JBoss Application Server v4.2.2

Es un proyecto de código abierto, basado en J2EE, e implementado al 100% en Java. Características de JBOSS AS(DE et al. 2018):

- Kernel completamente construido como una aplicación J2EE e implementado sobre Java Management Extensions (JMX)
- Despliegue dinámico de servicios Mbeans
- Gestión de dependencias
- Cargadores unificados de clases
- Proporciona la base para los componentes middleware de JBoss.

1.3.6 Herramientas empleadas

Red Hat JBoss DevStudio 8.0

Red Hat® JBoss® Developer Studio proporciona un soporte superior para todo el ciclo de vida de desarrollo en una sola herramienta. Entorno de desarrollo integrado (IDE) certificado y basado en Eclipse para desarrollar, probar e implementar aplicaciones web avanzadas, aplicaciones web móviles, aplicaciones empresariales transaccionales y aplicaciones y servicios de integración basados en la Arquitectura Orientada a Servicios (SOA) y Arquitectura Cliente-Servidor(SCA) («JBoss Developer Studio» 2018).

PostgreSQL Server 9.4.1

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES. Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglase integridad transaccional. PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos(Narváez Coello 2014).

Visual Paradigm for UML 8.0

Visual Paradigm for UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los

pasos del ciclo de vida del desarrollo de un software. La herramienta tiene un buen ambiente de trabajo, lo que facilita la visualización y manipulación del proyecto de modelado. También es compatible con cambios específicos en el código fuente de algunos lenguajes de programación como C ++ y Java (Mendoza Peña et al. 2016).

Conclusiones Parciales Capítulo 1

Una vez realizado el análisis de los sistemas existentes de gestión de información médica pediátrica, a nivel nacional e internacional, se llega a la conclusión de que los mismos no establecen una solución factible para aplicarlos en XAVIA HIS, pues son herramientas propietarias y cuentan con costos elevados para la adquisición de sus licencias. Otras, son dependientes a plataformas de trabajo predeterminadas o presentan características propias para una institución determinada. Teniendo en cuenta lo expuesto durante este capítulo, se decide desarrollar la hoja de consulta de la especialidad de Pediatría del módulo Consulta Externa para el sistema XAVIA HIS. Para ello se determinó utilizar la metodología, tecnologías y herramientas definidas por la dirección del centro para lograr la estandarización de todos sus productos.

Capítulo 2. Características del sistema

2.1 Flujo actual de los procesos involucrados en el campo de acción

La hoja de consulta médica de los niños y adolescentes, a través de la cual se realiza el proceso de gestión de información de la especialidad de Pediatría, es uno de los componentes más importantes del ciclo de atención médica de cada paciente. La misma registra el histórico de cada uno de ellos durante estas etapas de su vida, incluyendo el registro de sus padecimientos, alergias, enfermedades crónicas, por solo mencionar algunos, todos estos datos son elementos esenciales al momento de emitir un tratamiento acertado por el doctor responsable en ese momento.

Proceso atender paciente de Pediatría

El proceso se realiza de la siguiente manera, primero el doctor chequea que el paciente tiene una hoja de consulta médica, en caso negativo se comienza en ese mismo momento a crearle una, de lo contrario se revisa el documento en búsqueda de información relevante que pudiese ayudar en la toma de la decisión médica y así mejorar su posible efectividad. Después de terminados estos pasos se procede a la actualización de la misma con los resultados observados en el paciente y, de acuerdo a conducta a seguir que estipule el doctor, se le hacen análisis, exámenes físicos, se remite a una interconsulta o se le aplica un tratamiento. En caso de que fuese necesario el médico le da una orden de reconsulta.

2.2 Objeto de automatización

Durante los procesos de gestión de la información que se llevan a cabo en el momento de la atención de los pacientes pediátricos, se identificaron un conjunto de actividades y registros que serán los principales objetos a automatizar. Como resultado de esta actividad, estos procesos que antes se registraban de forma manual, ahora podrán contar con las ventajas que ofrece la informatización de procesos y que en capítulos anteriores fueron explicadas.

El proceso que se va a automatizar es Atender al paciente de Pediatría, y las actividades relacionadas con el proceso son Crear hoja de consulta de pediatría, Modificar hoja de consulta de Pediatría y Ver hoja de consulta de pediatría.

2.3 Modelo de Negocio

EL modelo de negocio es fundamental a la hora de comprender el problema a resolver, así como también logra proveer una visión general de los pasos necesarios para dar respuesta al mismo y todo esto se logra definiendo los procesos y roles que posee.

EL proceso de modelado se llevará a cabo mediante el uso de la herramienta Business Process Management Notation (BPMN), más específicamente el Visual Paradigm (VP), que cuenta con un conjunto de herramientas que facilitan el entendimiento del diagrama a realizar, así como los actores del negocio y las tareas que cada uno de ellos realiza en los determinados departamentos.

2.3.1 Actores y Trabajadores del Negocio

Un actor involucrado representa a un ser humano, software o hardware, institución o entidad, que desempeña un rol dentro de los procesos del negocio. Este interactúa con entidades y otros involucrados para que el negocio funcione. Son roles y no posiciones organizacionales, ya que una persona puede desempeñar varios roles, pero sólo tiene una posición en la organización. Ellos realizan las actividades y son propietarios de elementos.

Tabla 2 Roles y funciones del Negocio

Rol	Funciones
Paciente(Niño)	Recibir atención médica.
Médico	Interrogar, examinar, diagnosticar, indicar complementarios y tratamiento a pacientes que solicitan asistencia médica. Entregar al (a) técnico (a) de registros y estadísticas de salud la relación de pacientes.
Técnico (a) de registros de salud del área de consulta externa	Entregar al médico las historias clínicas de los pacientes citados y la hoja de Relación de pacientes antes del horario de consulta.

2.3.2 Diagrama de procesos del negocio

BPMN define diagramas de procesos de negocio basados en la técnica de diagramas de flujo, adaptados para graficar las operaciones de los procesos de la organización. Se compone de un conjunto de elementos gráficos que facilitan un diagrama entendible, por ejemplo: objetos de flujo, de conexión, de carriles y artefactos. Para un mejor entendimiento a continuación se explican los elementos utilizados en el modelo.

Objetos de flujos:

Eventos: dirigen el proceso uno de los elementos que componen a los diagramas con el objetivo de lograr una mayor comprensión.

Actividades: es un término genérico para el trabajo que una organización realiza. Pueden ser tareas o subprocesos.

Decisiones: se usa para controlar la divergencia de la secuencia de un flujo. Determina las decisiones, tanto bifurcaciones, como uniones y acoplamientos de flujos.

Objetos de conexión:

Flujo de secuencia: es usado para mostrar el orden (secuencia) de la actividad dentro del proceso.

Flujo de mensajes: es usado para mostrar el flujo de mensajes entre dos participantes de procesos separados (entidades del negocio o roles del negocio).

Asociaciones: Se usa para asociar datos, textos u otros artefactos con flujos de objetos.

Objetos de carriles:

Piscinas (Pool): representa un Participante en un proceso. También actúa como contenedor gráfico para separar al grupo de actividades realizadas por un participante de otros pools.

Calles (Lanes): es una partición dentro de un pool y se extiende a lo largo de todo el pool, tanto vertical como horizontalmente. Los lanes son usados para organizar y categorizar actividades.

Artefactos:

Objetos de datos: Los Data Objects son un mecanismo para mostrar como las actividades requieren o producen objetos. Ellos se conectan a las actividades a través de asociaciones.

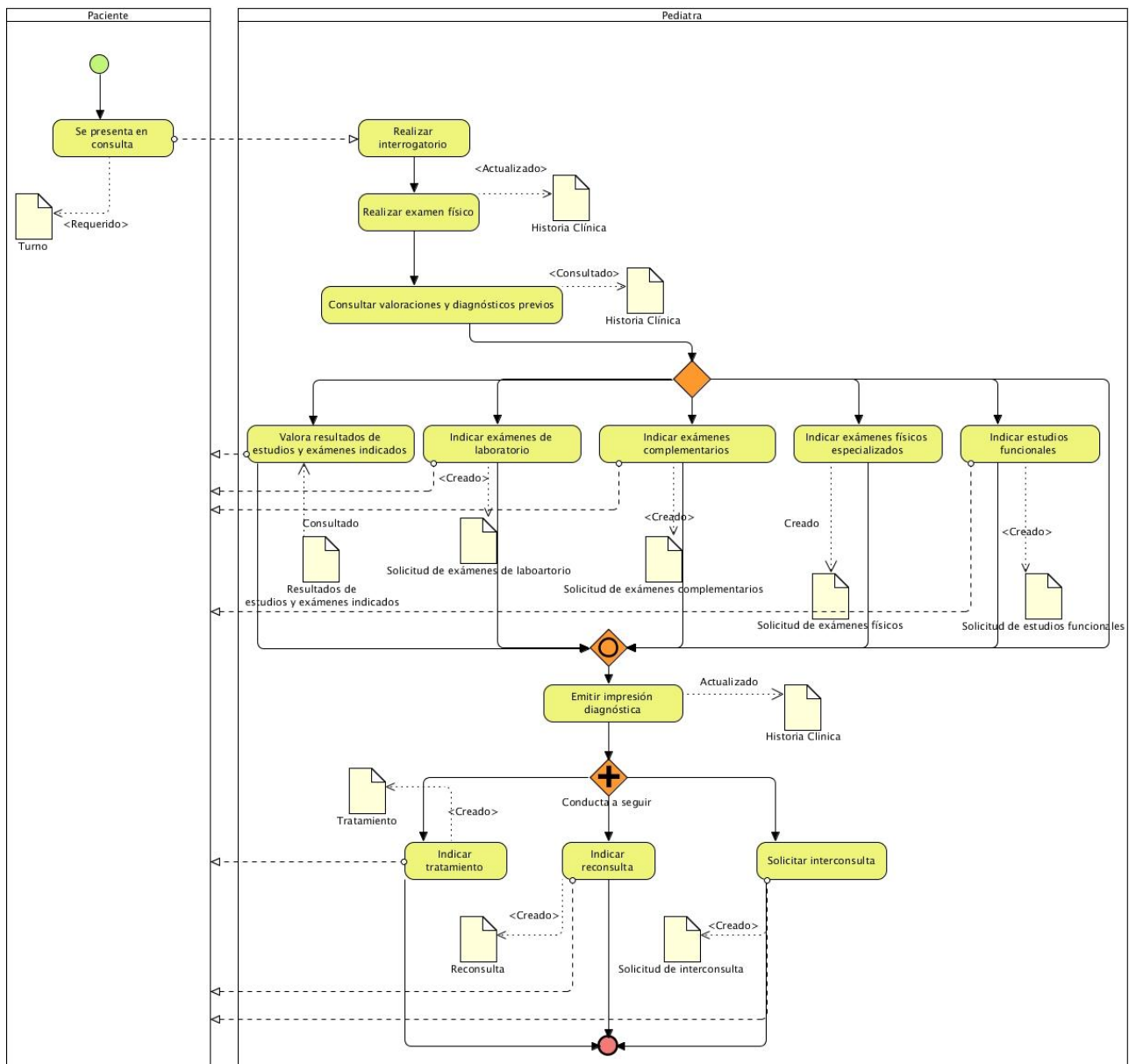


Ilustración 1 Diagrama de Procesos del Negocio: Atender Paciente de Pediatría

2.4 Propuesta del sistema

El objetivo principal es desarrollar la hoja de consulta de la especialidad de Pediatría del módulo Consulta Externa para el sistema XAVIA HIS para facilitar la gestión de la información de los procesos relacionados con esta especialidad en el área de Consulta Externa en instituciones de salud cubanas. Las características básicas estarán centradas en:

- Mostrar a los usuarios las opciones a las que tiene acceso.
- Crear, modificar y ver las distintas Hojas de Consulta de Pediatría dependiendo de los permisos.

2.5 Especificación de los requerimientos de software

2.5.1 Requerimientos funcionales

Los requerimientos funcionales especifican una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas, definiendo un comportamiento de entrada y salida.

Dentro de las acciones identificadas como requisitos funcionales para el desarrollo del sistema se pueden mencionar las que se listan a continuación:

1. Crear hoja de consulta de pediatría
2. Ver hoja de consulta de pediatría
3. Modificar hoja de consulta de pediatría

En la siguiente tabla se hace un pequeño resumen de los requisitos funcionales del sistema con una descripción breve de cada uno.

Tabla 3 Descripción de los requisitos funcionales

No.	Nombre	Descripción	Prioridad para el cliente	Complejidad
1.	Crear hoja de consulta de Pediatría	Permite crear la hoja de consulta de Pediatría con los datos generales, entrevista y datos específicos del paciente.	Alta	Alta
2.	Ver datos de hoja de consulta de Pediatría	Permite observar los datos de la hoja de consulta de Pediatría de un paciente específico.	Media	Baja
3.	Modificar hoja de consulta de Pediatría	Permite modificar, en caso necesario la hoja de consulta de Pediatría de un paciente específico.	Alta	Alta

2.5.1.1 Descripción detallada de los requisitos funcionales asociados al proceso atender al paciente de pediatría

RF 1. Crear hoja de consulta de pediatría

Tabla 4 Crear hoja de consulta de pediatría

Descripción textual	El requisito inicia cuando el actor accede a la opción Seleccionar un paciente de la lista de pacientes programados o no programados para un Pediatra y el sistema brinda la posibilidad de introducir y seleccionar los datos asociados a las diferentes secciones que componen la hoja de consulta, el actor introduce y selecciona los datos, el sistema adiciona la hoja de consulta, el requisito termina.
Actores	Médico de consulta externa de Pediatría
Precondiciones	El paciente debe estar citado o incluido en la lista de pacientes no programados.
Flujo de eventos	
Flujo básico Crear hoja de consulta de Pediatría	
1.	El requisito inicia cuando el actor accede a la opción seleccionar un paciente de la lista de pacientes programados o no programados para un Pediatra
2.	<p>El sistema muestra por defecto la información asociada a la pestaña:</p> <ul style="list-style-type: none"> ➤ Datos Generales. ➤ Brinda la posibilidad de seleccionar las pestañas: ➤ Antecedentes ➤ Examen Físico ➤ Historia de la Enfermedad Actual ➤ Hospitalización <p>Además, permite seleccionar las opciones:</p> <ul style="list-style-type: none"> ➤ Diagnóstico. Se ejecuta el requisito, ver requisito Elementos Comunes: Seleccionar enfermedad ➤ Conducta seguida. ➤ Ver opciones. Ver Flujo alternativo 1: “Ver opciones” ➤ Consultar acciones realizadas hasta el momento. Ver Flujo alternativo 2: “Consultar acciones realizadas hasta el momento” <p>Y permite:</p> <ul style="list-style-type: none"> ➤ Aceptar crear hoja general de consulta.

	<ul style="list-style-type: none"> ➤ Cancelar operación. Ver Flujo alternativo 3: “Cancelar operación”
3.	<p>El actor selecciona los datos relacionados con la conducta a seguir:</p> <ul style="list-style-type: none"> ➤ Conducta seguida (Examen Físico , Interconsulta, Referencia)
4.	<ul style="list-style-type: none"> ➤ El actor selecciona la opción Aceptar.
5.	<p>El sistema muestra un mensaje de información “¿Está seguro que desea crear la hoja de la consulta?”.</p> <p>Y permite:</p> <ul style="list-style-type: none"> ➤ Aceptar (Sí) ➤ Cancelar (No). Ver Flujo alternativo 3: “Cancelar operación”
6.	El actor selecciona Sí.
7.	El sistema valida los datos. Si hay datos incompletos. Ver Flujo alternativo 4: “Algunos de los datos entrados están incompletos o son incorrectos”. Si hay datos incorrectos. Ver Flujo alternativo 5: “Algunos de los datos entrados están incompletos o son incorrectos”.
8.	El sistema adiciona los datos de la hoja de consulta, crea el CDA Hoja general de consulta, y guarda las solicitudes de exámenes físicos o de laboratorio si se crearon alguno.
9.	El requisito termina.

Secciones

Sección 1 Datos Generales

1.	<p>El sistema muestra los datos predeterminados:</p> <ul style="list-style-type: none"> ➤ Datos generales: ➤ No. H.C. ➤ Foto ➤ Nombre ➤ Primer apellido ➤ Segundo apellido ➤ No. de identidad ➤ Fecha de nacimiento
----	---

- Sexo
- Edad

Datos escolares:

- Nombre del centro escolar
- Tipo del centro escolar
- Dirección

Además, visualiza la información registrada en los paneles:

- Antecedentes personales (Antecedentes, Fecha, Crónico, Descripción)
- Antecedentes familiares (Antecedentes, Parentesco, Descripción)
- Antecedentes quirúrgicos (Procedimiento/Categoría, Fecha intervención, Descripción)
- Transfusiones previas (Fecha, Preparado, Hora, No. frasco, Volumen, ABO Frasco, Técnico)
- Historia Socio Ambiental (Animales en la vivienda, Higiene en la Vivienda, Convivencia, Nivel de Escolaridad)
- Inmunizaciones (Listado de vacunas recibida)

Brinda la posibilidad de registrar los datos:

- Registro de Intervenciones Quirúrgicas, Tratamientos y Secuelas

Y permite:

- Gestionar antecedentes familiares. Ver Flujo alternativo 7: “Gestionar antecedentes
- Gestionar Registro de Intervenciones Quirúrgicas, Tratamientos y Secuelas. Ver Flujo alternativo 9: “Gestionar antecedentes quirúrgicos”

El actor registra los datos asociados a cada una.

El sistema regresa al paso 2 del Flujo básico.

Sección 2 Interrogatorio

El sistema muestra las siguientes secciones:

- Embarazo
- Parto

- Post-Natales
- Desarrollo Psicomotor y Adolescencia
- Alimentación

Y permite seleccionar e introducir los datos asociados a cada una.

El actor selecciona e introduce los datos asociados a las secciones seleccionadas.

El sistema regresa al paso 2 del Flujo básico.

Sección 3 Examen Físico

El sistema muestra las siguientes secciones:

- General
- Regional
- Por Sistemas

Y permite seleccionar e introducir los datos asociados a cada una.

El actor selecciona e introduce los datos asociados a las secciones seleccionadas.

El sistema regresa al paso 2 del Flujo básico.

Sección 4 Historia de la Enfermedad Actual

El sistema muestra las siguientes secciones:

- Historia de la enfermedad
- Antecedentes Patológicos familiares y personales
- Intervenciones Quirúrgicas

Y permite seleccionar e introducir los datos asociados a cada una.

El actor selecciona e introduce los datos asociados a las secciones seleccionadas.

El sistema regresa al paso 2 del Flujo básico.

Sección 5 Hospitalización

El sistema muestra las siguientes secciones:

- Datos de Hospitalización
- Discusión Diagnostica de la Evolución

	<ul style="list-style-type: none"> ➤ Resumen de Egreso ➤ Indicaciones y Tratamiento ➤ Estado al Egreso ➤ Historial de Temperaturas <p>Y permite seleccionar e introducir los datos asociados a cada una.</p>
	El actor selecciona e introduce los datos asociados a las secciones seleccionadas.
	El sistema regresa al paso 2 del Flujo básico.
Flujos alternativos	
Flujo alternativo 1 “Ver opciones”	
1.	<p>El sistema permite:</p> <ul style="list-style-type: none"> ➤ Crear orden de ingreso. Ver Flujo alternativo 17: “Crear orden de ingreso” ➤ Indicar reconsulta. Ver Flujo alternativo 22: “Indicar reconsulta” ➤ Crear solicitud de interconsulta. Ver Flujo alternativo 11: “Crear solicitud de interconsulta” ➤ Crear referencia médica. Ver Flujo alternativo 12: “Crear referencia médica” ➤ Si es un paciente citado para una interconsulta. Consultar solicitud de interconsulta. Ver Flujo alternativo 13: “Consultar solicitud de interconsulta” ➤ Si es un paciente referido de otro servicio u hospital: Consultar referencia médica. Ver Flujo alternativo 14: “Consultar referencia médica” ➤ Si se le indicó al paciente interconsulta(s): Buscar resultado de interconsulta. Ver Flujo alternativo 15: “Buscar resultado de interconsulta” ➤ Buscar hojas de consulta. Ver Flujo alternativo 16: “Buscar hojas de consulta”.
2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 2 “Consultar acciones realizadas hasta el momento”	
1.	El actor selecciona la opción Consultar acciones realizadas hasta el momento. Se ejecuta el requisito, ver requisito Elementos Comunes: Consultar acciones realizadas hasta el momento.
2.	El sistema regresa al paso 2 del Flujo básico.

Flujo alternativo 3 “Cancelar operación”

1. El actor selecciona la opción Cancelar.
2. El sistema verifica el origen del cancelar si se seleccionó “No” en el mensaje de confirmación “Está seguro que desea crear la hoja de consulta” regresa al crear hoja de consulta, si se selecciona el cancelar de la hoja consulta regresa al Consultar relación de pacientes programados o Consultar relación de pacientes no programados.

Flujo alternativo 4 “Existen datos incompletos”

1. El sistema valida si se seleccionó el diagnóstico, si no se seleccionó muestra el mensaje “Debe seleccionar al menos una enfermedad para conformar el diagnóstico”.

Si en la sección asociada a la Conducta seguida se seleccionaron los valores Referencia, Interconsulta o Trabajo Social y no se generaron las solicitudes asociadas, se muestra el siguiente mensaje en dependencia del valor seleccionado “Debe registrar <una referencia médica, una solicitud de interconsulta, una indicación médica o un anuncio de operaciones respectivamente>”. Se crea un mensaje por cada valor.

Si en la sección asociada a la Conducta seguida no se seleccionó ningún elemento se muestra el mensaje “Debe seleccionar al menos una conducta a seguir”.

Si en la hoja no se registraron todos los elementos requeridos se muestra un indicador (asterisco rojo) al lado de los campos incompletos, en la sección asociada y la pestaña.
2. El sistema regresa al paso 2 del Flujo básico.

Flujo alternativo 5 “Existen datos incorrectos”

1. El sistema muestra un indicador (asterisco rojo) al lado de los campos incorrectos con un mensaje en dependencia del error cometido.
2. El sistema regresa al paso 2 del Flujo básico.

Flujo alternativo 6 “Gestionar antecedentes personales”

1. El actor selecciona la opción Gestionar antecedentes personales. Se ejecuta el requisito, ver requisito Elementos Comunes: Gestionar antecedentes personales.
2. El sistema regresa al paso 2 del Flujo básico.

Flujo alternativo 7 “Gestionar antecedentes familiares”

1. El actor selecciona la opción Gestionar antecedentes familiares. Se ejecuta el requisito, ver requisito Elementos Comunes: Gestionar antecedentes familiares.

2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 8 “Gestionar inmunizaciones ”	
1.	El actor selecciona la opción Gestionar inmunizaciones. Se ejecuta el requisito, ver requisito Elementos Comunes: Gestionar inmunizaciones.
2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 9 “Gestionar intervenciones quirúrgicas ,traumatismos y secuelas”	
1.	El actor selecciona la opción Gestionar antecedentes quirúrgicos. Se ejecuta el requisito, ver requisito Elementos Comunes: Gestionar intervenciones quirúrgicas, traumatismos y secuelas.
2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 10 “Gestionar examen físico ”	
1.	El actor selecciona la opción Gestionar transfusiones previas. Se ejecuta el requisito, ver requisito Elementos Comunes: Gestionar examen físico.
2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 11 “Crear solicitud de interconsulta”	
1.	El actor selecciona la opción Crear solicitud de interconsulta. Se ejecuta el requisito, ver requisito Elementos Comunes: Crear solicitud de interconsulta.
2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 12 “Crear referencia médica”	
1.	El actor selecciona la opción Crear referencia médica. Se ejecuta el requisito, ver requisito Elementos Comunes: Crear referencia médica.
2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 13 “Consultar solicitud de interconsulta”	
1.	El actor selecciona la opción Consultar solicitud de interconsulta. Se ejecuta el requisito, ver requisito Consultar solicitud de interconsulta.
2.	El sistema regresa al paso 2 del Flujo básico.
Flujo alternativo 14 “Consultar referencia médica”	
1.	El actor selecciona la opción Consultar referencia médica. Se ejecuta el requisito, ver

	requisito Consultar referencia médica.	
2.	El sistema regresa al paso 2 del Flujo básico.	
Flujo alternativo 15 “Buscar resultado de interconsulta”		
1.	El actor selecciona la opción Buscar resultado de interconsulta. Se ejecuta el requisito, ver requisito Buscar resultado de interconsulta.	
2.	El sistema regresa al paso 2 del Flujo básico.	
Flujo alternativo 16 “Buscar hojas de consulta”		
1.	El actor selecciona la opción Buscar hojas de consulta. Se ejecuta el requisito, ver requisito Buscar hojas de consulta.	
2.	El sistema regresa al paso 2 del Flujo básico.	
Flujo alternativo 17 “Crear orden de ingreso”		
1.	El actor selecciona la opción Crear orden de ingreso. Se ejecuta el requisito, ver requisito Elementos Comunes: Crear orden de ingreso.	
2.	El sistema regresa al paso 2 del Flujo básico.	
Flujo alternativo 18 “Indicar reconsulta”		
1.	El actor selecciona la opción Asignar cita sucesiva en la atención. Se ejecuta el requisito, ver requisito Elementos Comunes: Indicar reconsulta.	
2.	El sistema regresa al paso 2 del Flujo básico.	
Pos-condiciones		
1.	Quedan creadas la hoja de consulta, el documento CDA Hoja general de consulta y las solicitudes asociadas a la atención.	
Validaciones		
1.	N/A	
Conceptos	Hoja de consulta	N/A
	N/A	N/A
Restricciones del sistema	La opción Consultar acciones realizadas hasta el momento se muestra deshabilitada mientras no se haya creado una solicitud.	
	Cuando el actor selecciona la opción No refiere en las secciones	

correspondientes a la pestaña Interrogatorio, el sistema debe seleccionar todas las opciones No asociadas a estas secciones.

Cuando el actor selecciona la opción Nada a señalar en las secciones correspondientes a la pestaña Examen físico, el sistema debe seleccionar todas las opciones Normal (N) asociadas a estas secciones.

El campo Observaciones de cada una de las secciones se habilita si se selecciona al menos una de las siguientes opciones asociadas a las secciones (Sí, Anormal o No examinado).

En la pestaña Interrogatorio, sección Historia de la Enfermedad Actual el elemento Observaciones se habilita si se selecciona el elemento Otros.

En el ver opciones, los vínculos Crear solicitud de interconsulta, Crear referencia médica, Crear indicaciones médicas y Crear anuncio de operaciones se deben mostrar deshabilitados y solo se deben habilitar si en la sección Conducta seguida se seleccionan los elementos Interconsulta, Referencia, Tratamiento médico y Tratamiento quirúrgico respectivamente, en el caso específico del crear anuncio de operaciones además debe cumplirse que el usuario sea un cirujano y el servicio sea quirúrgico.

Una vez que se crea una solicitud de interconsulta, una referencia médica, las indicaciones médicas y un anuncio de operaciones, en la sección Conducta seguida los elementos Interconsulta, Referencia, Tratamiento médico y Tratamiento quirúrgico respectivamente deben mostrarse marcados y deshabilitados. En caso de que se elimine una de estas solicitudes el elemento correspondiente en la sección Conducta seguida debe mostrarse desmarcado y habilitado.

En el ver opciones los elementos Consultar solicitud de interconsulta y Consultar referencia médica se deben mostrar deshabilitados y solo se habilitan si el origen de la consulta parte de una solicitud de interconsulta o una referencia médica respectivamente.

En el Ver opciones la opción Registro de Intervenciones Quirúrgicas, Tratamientos y Secuelas debe mostrarse cuando se haya registrado al menos una intervención quirúrgica o tratamiento médico.

En Ver opciones una vez que se cree un elemento asociado a:

- Solicitud de análisis de laboratorio




- Indicación médica
- Referencia médica
- Registro de reacciones adversas
- Certificado médico
- Informe médico
- Solicitud de examen físico
- Indicar reconsulta
- Registro de intervenciones Quirúrgicas, Traumatismos y Secuelas
- Orden de ingreso

Se deben deshabilitar las opciones, solo se habilitan en caso de que se eliminen.

La pestaña Seguimiento solo se muestra si el paciente es citado para una consulta para seguimiento o una interconsulta.

Dependencias	Obligatoria	Elementos Comunes: Seleccionar conducta a seguir.
---------------------	-------------	---

	Opcional	<p>Elementos Comunes: Consultar acciones realizadas hasta el momento.</p> <p>Elementos Comunes: Gestionar antecedentes personales.</p> <p>Elementos Comunes: Gestionar antecedentes familiares.</p> <p>Elementos Comunes: Gestionar registro de Intervenciones, Traumatismos y Secuelas.</p> <p>Elementos Comunes: Crear solicitud de interconsulta.</p> <p>Elementos Comunes: Crear referencia médica.</p> <p>Consultar solicitud de interconsulta.</p> <p>Consultar referencia médica.</p> <p>Buscar resultado de interconsulta.</p> <p>Buscar hojas de consulta.</p> <p>Crear solicitud de examen físico.</p> <p>Elementos Comunes: Indicar reconsulta.</p> <p>Elementos Comunes: Crear orden de ingreso.</p>
Requisitos especiales	N/A	
Asuntos pendientes	N/A	
Prototipo elemental de interfaz gráfica de usuario		

Datos personales	Interrogatorio	Historia de la enfermedad actual	Examen físico																
1Datos generales del paciente		1Datos Escolares																	
<table border="1"> <thead> <tr> <th colspan="3">1Datos generales del paciente</th> <th>1No.H.C:</th> </tr> </thead> <tbody> <tr> <td></td> <td>1Nombre: -</td> <td>1Cédula: -</td> <td>1Tipo de paciente: -</td> </tr> <tr> <td></td> <td>1Primer apellido: -</td> <td>1Fecha de nacimiento: -</td> <td>1Edad: -</td> </tr> <tr> <td></td> <td>1Segundo apellido: -</td> <td>1Sexo: -</td> <td>1ABO/Rh: -</td> </tr> </tbody> </table>				1Datos generales del paciente			1No.H.C:		1Nombre: -	1Cédula: -	1Tipo de paciente: -		1Primer apellido: -	1Fecha de nacimiento: -	1Edad: -		1Segundo apellido: -	1Sexo: -	1ABO/Rh: -
1Datos generales del paciente			1No.H.C:																
	1Nombre: -	1Cédula: -	1Tipo de paciente: -																
	1Primer apellido: -	1Fecha de nacimiento: -	1Edad: -																
	1Segundo apellido: -	1Sexo: -	1ABO/Rh: -																
Motivo de consulta:		Historia enfermedad actual:																	
<table border="1"> <tbody> <tr> <td>Antecedentes personales</td> <td><input type="checkbox"/> No refiere »</td> </tr> <tr> <td>Antecedentes familiares</td> <td><input type="checkbox"/> No refiere »</td> </tr> <tr> <td>Hábitos psicobiológicos</td> <td><input type="checkbox"/> No refiere »</td> </tr> <tr> <td>Antecedentes quirúrgicos</td> <td><input type="checkbox"/> No refiere »</td> </tr> <tr> <td>Transfusiones previas</td> <td>»</td> </tr> <tr> <td>Diagnóstico</td> <td>»</td> </tr> <tr> <td>Conducta seguida</td> <td>»</td> </tr> </tbody> </table>				Antecedentes personales	<input type="checkbox"/> No refiere »	Antecedentes familiares	<input type="checkbox"/> No refiere »	Hábitos psicobiológicos	<input type="checkbox"/> No refiere »	Antecedentes quirúrgicos	<input type="checkbox"/> No refiere »	Transfusiones previas	»	Diagnóstico	»	Conducta seguida	»		
Antecedentes personales	<input type="checkbox"/> No refiere »																		
Antecedentes familiares	<input type="checkbox"/> No refiere »																		
Hábitos psicobiológicos	<input type="checkbox"/> No refiere »																		
Antecedentes quirúrgicos	<input type="checkbox"/> No refiere »																		
Transfusiones previas	»																		
Diagnóstico	»																		
Conducta seguida	»																		
		<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>																	
Formatos de entrada/salida																			
N/A																			
Entradas																			
N/A																			
Salidas																			
N/A																			

2.5.2 Requerimientos no funcionales

Los requerimientos no funcionales especifican propiedades del sistema, como restricciones del entorno o implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad; de manera que definen las restricciones físicas sobre los requisitos funcionales.

Los requisitos no funcionales van a ser adoptados de los requisitos definidos para el proyecto XAVIA HIS del Centro de Soluciones Médicas, los cuales se encuentran especificados en la documentación de la misma incluida con el producto (Centro de Informática Médica. 2018).

Conclusiones Parciales Capítulo 2

En este capítulo se especificaron los principales procesos asociados a la hoja de consulta de Pediatría del módulo Consulta Externa. Con el apoyo de diagramas y fichas se logró hacer una representación detallada de cada proceso, que permitió un mejor entendimiento del problema y una rápida identificación de las funcionalidades y propiedades de la solución a desarrollar. A partir

de estas funcionalidades se definió el modelo del sistema, con sus diagramas y especificaciones de los requisitos funcionales y no funcionales.

Capítulo 3. Diseño del Sistema, Implementación y Pruebas

En el presente capítulo se describe la arquitectura especificada por la dirección del proyecto para el desarrollo de la solución, así como los artefactos concernientes al modelo de diseño: diagramas de clases y de paquetes, que son las entradas del modelo de implementación y se describen las clases asociadas a cada diagrama. Además, se muestra, de acuerdo al escenario número tres de la metodología AUP-UCI, los diagramas de paquetes y de clases asociados al módulo descrito.

3.1 Descripción de la Arquitectura

3.1.1 Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. El sistema XAVIA HIS se basa en la arquitectura cliente-servidor pues así garantiza estar accesible desde cualquier lugar y a cualquier hora siempre que se posea una conexión a la red y un navegador web.

3.2.1 Patrones de Arquitectura

Un patrón de arquitectura es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Para la realización de la solución propuesta se decidió utilizar el patrón de arquitectura del software Modelo-Vista-Controlador (MVC).

El patrón modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos. («IBM Knowledge Center - Patrón de diseño de modelo-vista-controlador» 2018).

Uno de los patrones más conocidos en el desarrollo web es el patrón MVC (Modelo Vista Controlador). Este patrón obliga a separar la lógica de control (sabe qué cosas hay que hacer pero no cómo), la lógica de negocio (sabe cómo se hacen las cosas) y la lógica de presentación (sabe cómo interactuar con el usuario) (DesarrolloWeb.com 2018).

El **modelo** contiene únicamente los datos puros de aplicación; no contiene lógica que describe cómo pueden presentarse los datos a un usuario.

La **vista** (por ejemplo, la información de presentación) presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos.

Por último, el **controlador** (por ejemplo, la información de control) está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista.

La mayoría de las aplicaciones hoy en día siguen este patrón, muchas con ligeras variaciones. Por ejemplo, algunas aplicaciones combinan la vista y el controlador en una clase porque ya están estrechamente unidos. Todas las variaciones recomiendan enérgicamente la separación de los datos de su presentación. Esto no sólo simplifica la estructura de una aplicación, sino que también permite reutilizar el código.

A partir de este patrón los componentes del sistema se dividen en tres capas (Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas «MÓDULO CONSULTA EXTERNA DEL SISTEMA DE INFORMACIÓN HOSPITALARIA alas HIS.pdf» Junio 2009):

La **capa de presentación** está formada por páginas XHTML que contienen formularios. Se desarrolla básicamente con JSF, haciendo uso de la librería de componentes Richfaces, la cual se integra fácilmente con el framework Seam. Además, incorpora controles out-of-the-box AJAX-ready y el framework de extensión AJAX para los controles JSF básicos Ajax4Jsf.

La **capa de negocio** está compuesta por clases controladoras y contiene reglas de gestión de eventos. Usa como framework a Seam, el cual integra todas las tecnologías estándares JSF, EJB3, JPA, además de BPM (Business Process Management) garantizando crear una capa consistente que impide todo tipo de errores.

Para el **acceso a datos** se usa la implementación de JPA de Hibernate 3.3, minimizando las configuraciones en XML sin chequeo de tipos y usando los servicios del contenedor de EJB3 y/o los contextos de persistencias administrados por Seam. Además, se pueden establecer validaciones end-to-end gracias a los Hibernate Validators.

El framework Seam es quien rige la comunicación entre los elementos de cada una de las capas, pues permite a través de anotaciones, que las páginas referencien las funcionalidades especificadas en las clases controladoras, y que estas a su vez puedan usar componentes tanto de la capa de acceso a datos como de la capa de negocio (Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas «MÓDULO CONSULTA EXTERNA DEL SISTEMA DE INFORMACIÓN HOSPITALARIA alas HIS.pdf» Junio 2009).

3.2 Modelo del Diseño del sistema

Un patrón es una descripción de un problema bien conocido que suele incluir (Tabares 2011):

- Descripción
- Escenario de Uso

- Solución concreta
- Las consecuencias de utilizar este patrón
- Ejemplos de implementación
- Lista de patrones relacionados

Patrones de Asignación de Responsabilidades (GRASP) Es un sistema orientado a objetos se compone de objetos que envían mensajes a otros objetos para que lleven a cabo las operaciones requeridas. Los diagramas de interacción describen gráficamente estas operaciones, a partir de los objetos en interacción, que se responsabilizan de una actividad determinada.

Estos patrones constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos, un patrón es una descripción de un problema y la solución a la que se le da un nombre, y que además se puede aplicar a nuevos contextos, los patrones GRASP se dividen en 5 ellos son: experto, creador, controlador, alta cohesión y bajo acoplamiento(Larman 2004).

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

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

Bajo Acoplamiento: El acoplamiento es una medida de fuerza con que un elemento está a, tiene conocimiento de, confía en, otros elementos. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

Alta Cohesión: La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento, Cada elemento de nuestro diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Controlador: Es un evento generado por actores externos. Se asocian con operaciones del sistema, operaciones del sistema como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer; coordina o controla la actividad. No realiza mucho trabajo por sí mismo.

Como antes mencionado acerca de la metodología utilizada en el desarrollo de la solución AUP-UCI, este propone en su escenario 3 dentro del Modelo de diseño el Diagrama de paquetes y el Diagrama de clases.

3.2.1 Diagrama de paquetes

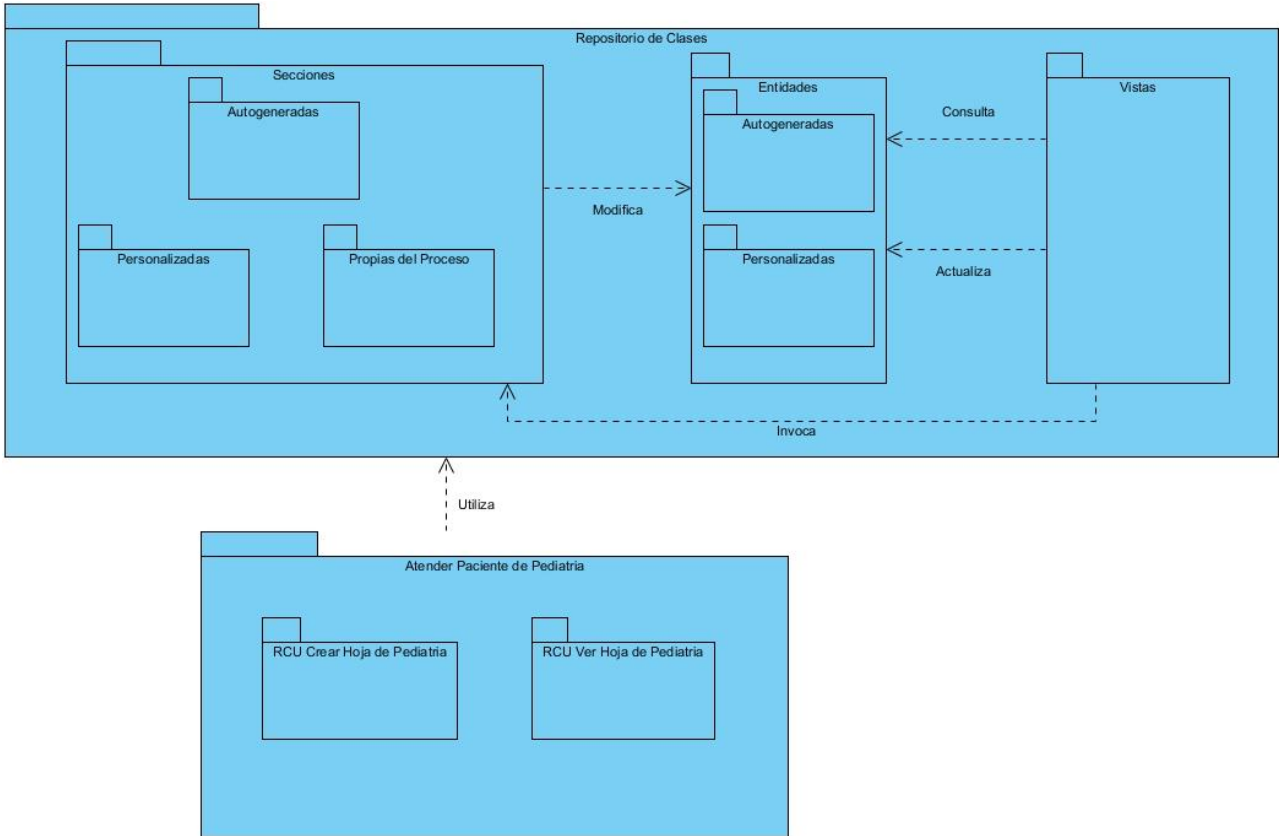


Ilustración 2 Diagrama de Paquetes

3.2.2 Diagrama de clases del diseño

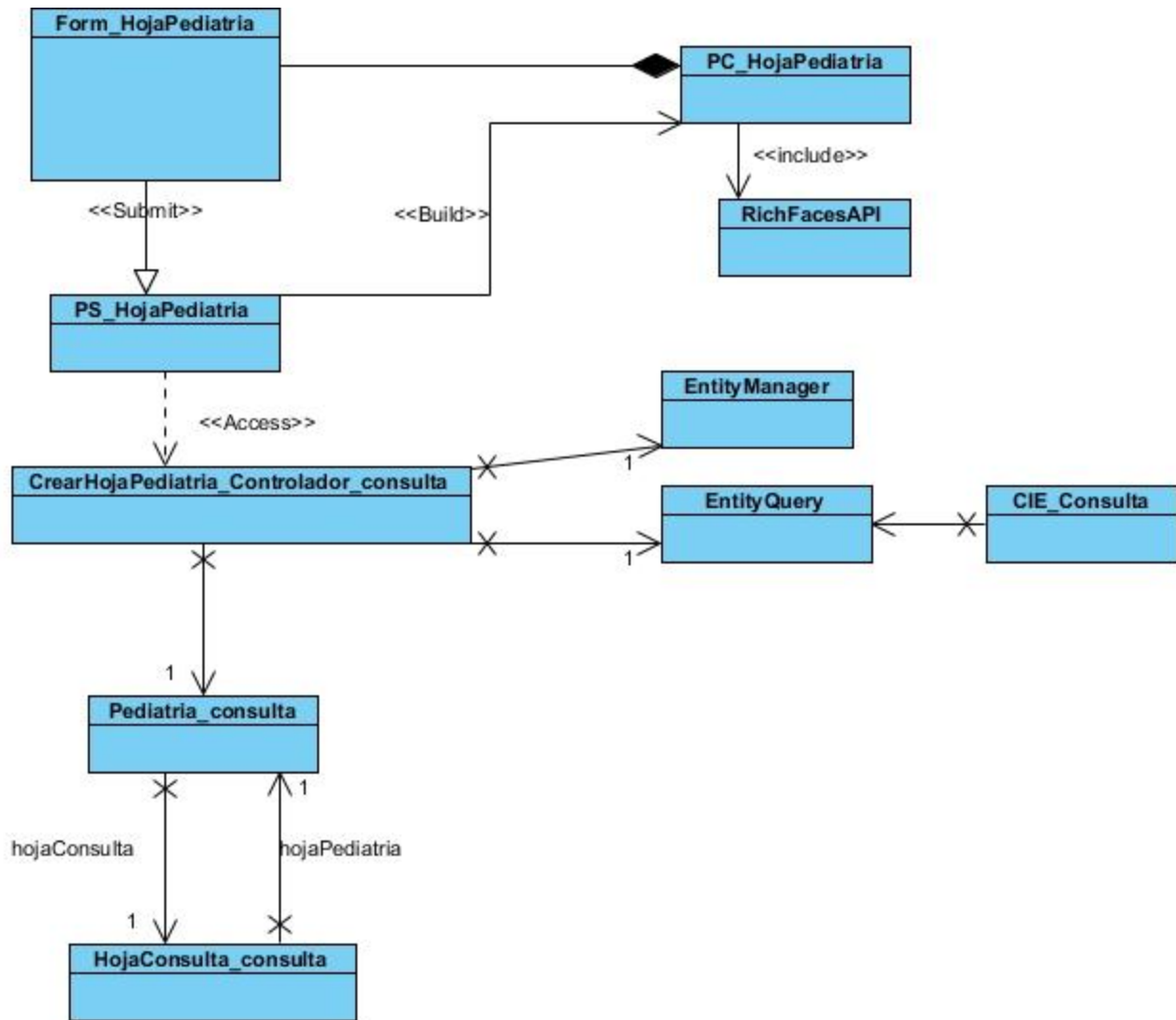


Ilustración 3 Diagrama de clases del diseño: Crear Hoja de Pediatría

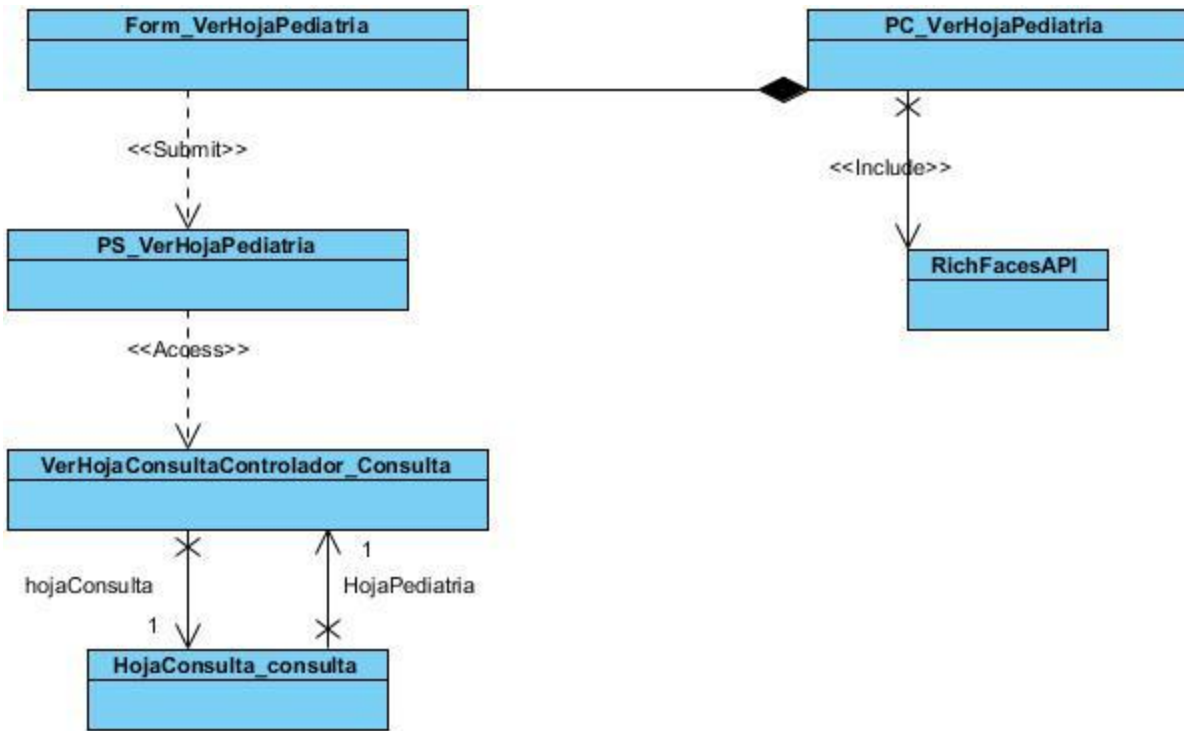


Ilustración 4 Diagrama de clases del diseño: Ver Hoja de Pediatría

A partir del resultado del diseño se obtiene el modelo de datos, diagrama de componente y despliegue; estos últimos, con la idea de mostrar las dependencias entre las partes del código y la estructura física del sistema en ejecución. Además, se especifica la estrategia de codificación a seguir, así como, elementos de seguridad y captura de errores.

3.2.3 Descripción de las clases del diseño

Descripción de las páginas controladoras en la hoja de Pediatría

Tabla 5 Descripción de la clase controladora: CrearHojaPediatríaControlador_consulta

Nombre: CrearHojaPediatríaControlador_consulta	
Tipo de Clase: Controladora	
Atributo	Tipo
pediatria	Pediatria_consulta
datosPaciente	PediatriaDatosPaciente_consulta
entrevista	PediatriaEntrevista_consulta
hojaConsulta	HojaConsulta_consulta
cita	Cita_consulta

result_exam_fis	ExamFis_consulta
registro_temperaturas	List<Temperatura>_consulta
alimentación	Alimentacion_consulta
desarrollo	Desarrollo_consulta
diagnostico	Diagnostico_consulta
Para cada responsabilidad	
Nombre:	String inicializarHoja()
Descripción:	Recibe el id de la cita y se cargan los antecedentes del paciente, el diagnóstico anterior, en caso de tener y se inicializan todos los campos de la hoja de pediatría.
Nombre:	String salvarHoja()
Descripción:	Valida y guarda los campos llenados de Pediatría_consulta (hoja de pediatría).
Nombre:	void Cancelar()
Descripción:	Cancela el proceso.

Tabla 6 Descripción de la clase controladora: VerHojaConsultaControlador_consulta

Nombre: VerHojaConsultaControlador_consulta	
Tipo de Clase: Controladora	
Para cada responsabilidad	
Nombre:	void setId(int id)
Descripción:	Recibe el valor del identificador de la hoja de consulta seleccionada. Busca esta hoja para que la misma sea mostrada con todos sus datos cargados.

3.3 Modelo de datos

Seguidamente se muestra el Diagrama Entidad-Relación de la base de datos utilizada en la solución.

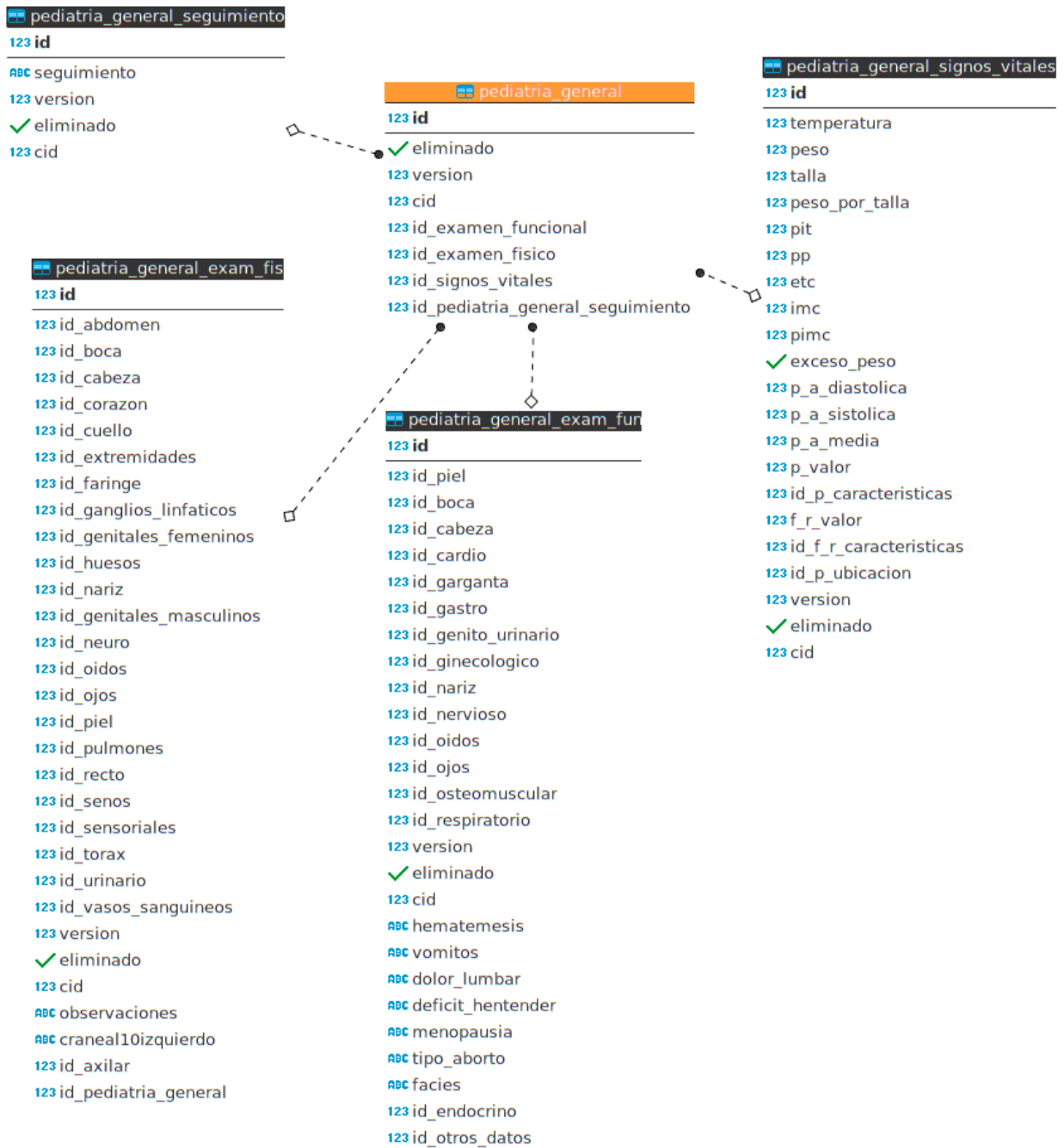


Ilustración 5 Diagrama Entidad-Relacion de la Base de datos

En la próxima tabla se muestra una descripción de atributos comunes entre todas las entidades:

Tabla 7 Descripción de atributos comunes entre todas las entidades

Atributo	Tipo	Descripción
id.	int4.	Id necesario en cada entidad para las referencias en las relaciones entre tablas.

version.	int4.	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.	bool.	Permite la eliminación lógica con que cuenta el sistema, cuando está en verdadero indica que la entidad está eliminada.
cid.	int4.	Permite identificar quién realiza alguna acción sobre la entidad.

Tabla 8 Descripción de la tabla *Pediatría_consulta*

A continuación, una descripción de la tabla principal de la base de datos *Pediatría_General*

Nombre: <i>Pediatría_General</i>		
Descripción: Tabla que contiene los datos pertenecientes a la hoja de consulta de pediatría, los identificadores de las tablas que almacenan los datos referentes a la consulta médica, con el fin de acceder a los resultados de las diferentes consultas que se le pueden realizar al paciente.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
eliminado	boolean	Permite verificar si la información ha sido eliminada
cid	integer	Identificador único de las acciones realizadas por el usuario.
id_indicaciones_medicas	integer	Identificador de la tabla indicacion_medica.
id_orden_admision	integer	Identificador de la tabla orden_admision.
id_solicitud_analisis_lab	integer	Identificador de la tabla solicitud_analisis_lab.
id_solicitud_intervencion_quirurgica	integer	Identificador de la tabla solicitud_intervencion_quirurgica.
id_cita_consulta	integer	Identificador de la tabla cita.
fecha	date	Fecha en que es creada la hoja de consulta.

id_impresion_diagnostica	integer	Identificador de la tabla impresion_diagnostica.
id_diagnostico_final	integer	Identificador de la tabla diagnostico_final.
id_observaciones	integer	Identificador de la tabla observaciones, donde se almacenan las observaciones realizadas por el médico en la consulta.
id_conclusiones	integer	Identificador de la tabla conclusiones, donde se almacena las conclusiones de la consulta médica.
id_diagnostico_princ_final	integer	Identificador de la tabla diagnostico_princ_final.
id_hoja_general	integer	Identificador de la tabla general, que es donde se almacenan los datos referentes a una consulta general.
motivo	text	Motivo por la que el paciente acude a la consulta médica.
enfermedad_actual	text	Enfermedad que presenta el paciente.
hora	time	Hora en la que se crea la hoja de consulta.
id_cita_origen	integer	Identificador de la tabla cita, la cual almacena los datos referentes a la cita reservada al paciente para la consulta.

3.4 Estrategias de integración

Debido al inmenso campo que ocupa la Pediatría como rama de la medicina, y para evitar duplicidad y reiteraciones innecesarias a la hora de programar el modulo, se hizo uso en este proceso de desarrollo de la reutilización de código previamente implementado para otros módulos del área de Consulta Externa del sistema XAVIA HIS de manera común, para así explotar los beneficios de la reutilización de código, dándole estabilidad mantenibilidad y modularidad al software que se desarrolló.

Entre ellos se pueden destacar, pero no limitándose solamente a estos, los siguientes elementos:

- Clase HojaGeneral_consulta
- ClaseExamenFisicoGeneral

➤ Clase CrearSolicitudInterconsulta

Y todas las tablas de la base de datos que fueran necesarias para el correcto funcionamiento de dichos componentes.

3.5 Implementación de las hojas de consulta

A continuación, se muestra el diagrama de despliegue de la solución propuesta:

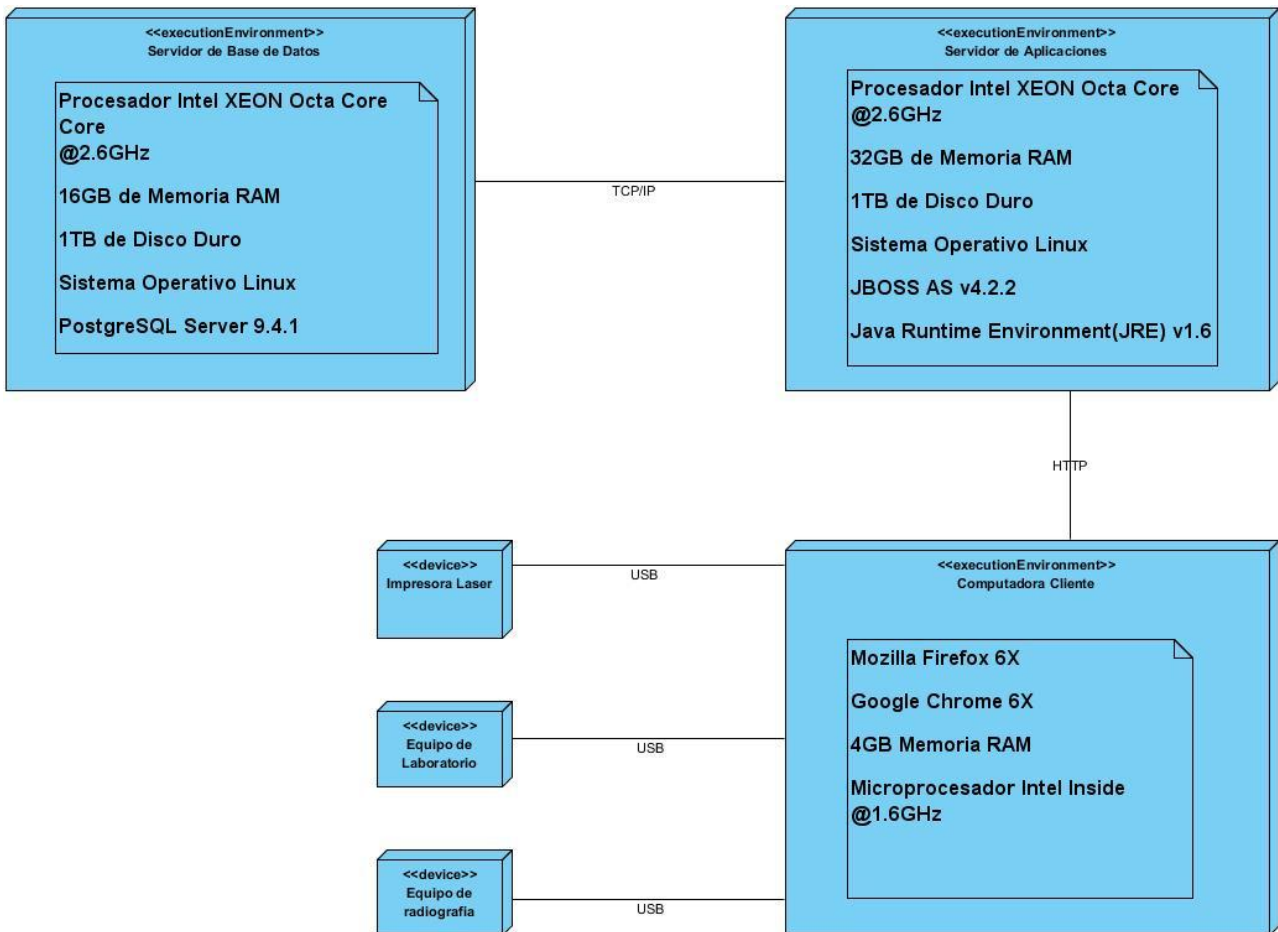


Ilustración 6 Diagrama de Despliegue

3.5.1 Estructura del modelo de implementación

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

Un componente es un módulo de software que puede ser código fuente, código binario, un ejecutable, o una librería con una interfaz definida. Una interfaz establece las operaciones externas de un componente, las cuales determinan una parte del comportamiento del mismo. Además, se representan las dependencias entre componentes o entre un componente y la interfaz de otro, es decir uno de ellos usa los servicios o facilidades del otro.

Estos diagramas pueden incluir paquetes que permiten organizar la construcción del sistema de información en subsistemas y que recogen aspectos prácticos relacionados con la secuencia de compilación entre componentes y la agrupación de elementos en librerías («Diagramas de Componentes | LENGUAJE DE MODELADO UNIFICADO - UML» 2018).

De acuerdo con el patrón de diseño mencionado en capítulos anteriores se realizó, dividiendo en tres componentes principales Modelo-Vista-Controlador, el diagrama de componentes para la hoja de Pediatría del módulo Consulta Externa del sistema XAVIA HIS.

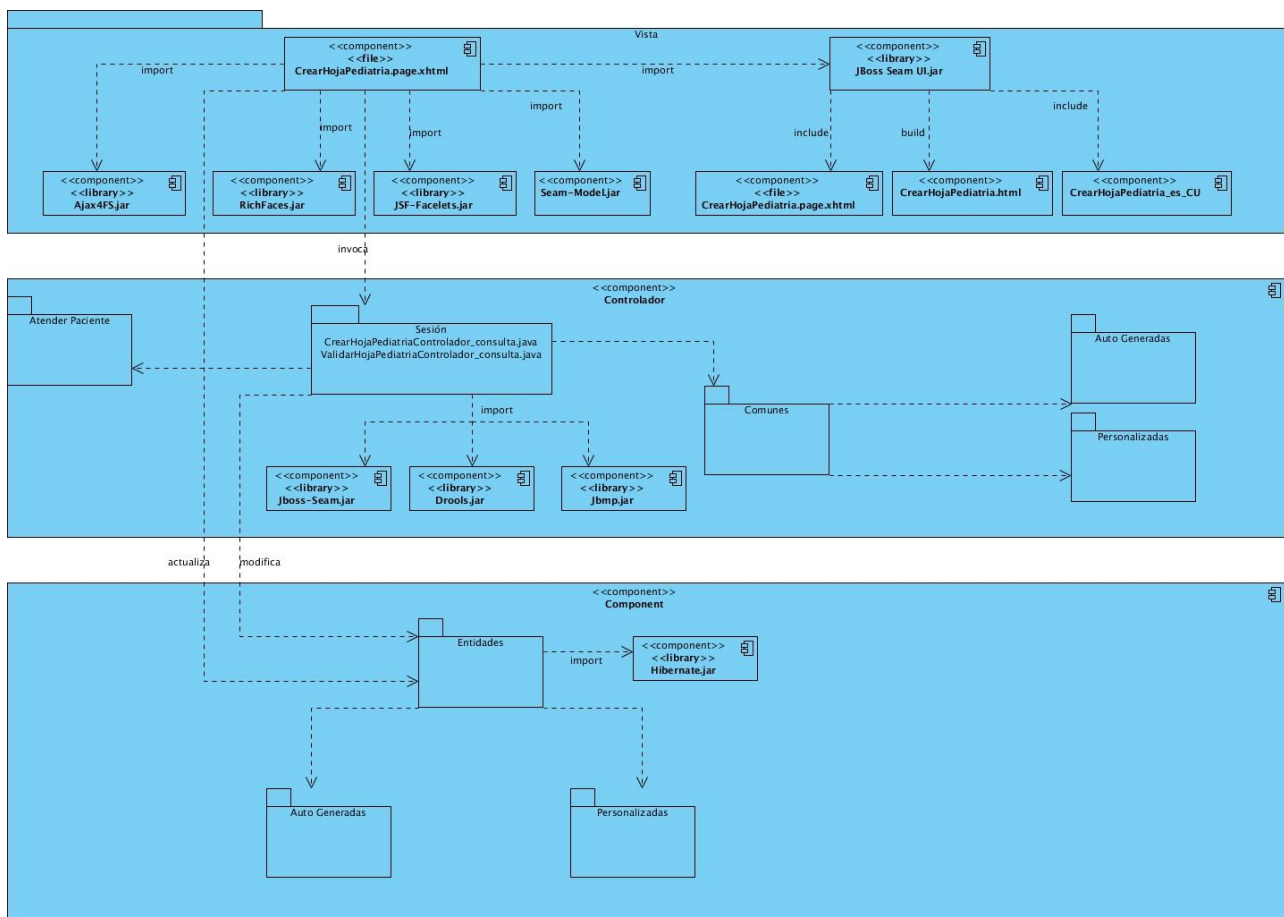


Ilustración 7 Diagrama de componentes de la hoja de Pediatría

3.5.2 Tratamiento de errores

Para el asegurar el buen funcionamiento de la solución propuesta, así como el del sistema en general, en todas las clases que fuese necesario se utilizaron las facilidades del lenguaje para el tratamiento de errores y excepciones. Dicho tratamiento se llevó a cabo a través de las palabras

reservadas del lenguaje Java: try y catch. Estas encierran un bloque de código en Java y se aseguran que cualquier error o excepción que ocurra dentro de esta zona sea debidamente tratada dependiendo de lo que se le programe hacer. En cada uno de los casos se le envía un mensaje al usuario a través de la interfaz gráfica del XAVIA HIS, gracias a las ventajas que posee el framework Seam y su componente FacesMessages. Todos los diferentes mensajes se guardan en un fichero page.xml con el tipo de error y la información que se debe mostrar.

3.5.3 Seguridad

En todo sistema la seguridad es uno de los factores más importantes a tener en cuenta durante el desarrollo e implementación. En el caso de la información que gestiona el modulo hoja de consulta de Pediatría, estos datos son confidenciales y solo para verse por el personal autorizado. Para cumplir con estas restricciones de confidencialidad, el sistema XAVIA HIS cuenta con varias funcionalidades entre las que se encuentra por solo mencionar dos de las más importantes: iniciar y cerrar sesión de usuario. La primera se encarga de asegurarse que solo el personal con los permisos requeridos puede iniciar sesión y ver los datos de los pacientes de acuerdo a su nivel de permisos y la segunda se asegura de cerrar la sesión anterior para que otra persona no puede utilizarla.

Además de lo antes planteado también se cuenta con la posibilidad de asignar y denegar permisos, de acuerdo a las ocupaciones de cada usuario o roll registrado en el sistema, permitiendo solo el acceso a las diversas funcionalidades al personal calificado para las mismas.

3.5.4 Estrategias de codificación. Estándares y estilos a utilizar

Para la solución del problema tratado se utilizó el estándar de programación de la Sun Microsystems para el lenguaje de programación Java. Está basado en los estándares recomendados por Sun Microsystems, que han sido difundidos y aceptados ampliamente por toda la comunidad, y que han terminado por consolidarse como un modelo estándar de programación de facto. El cual se caracteriza por (Flower 2010):

- El uso de normas de programación es muy útil a la hora del desarrollo de aplicaciones por muchas razones, entre las que destacan:
- Facilitan el mantenimiento de una aplicación. Dicho mantenimiento constituye el 80% del coste del ciclo de vida de la aplicación.
- Permiten que cualquier programador entienda y pueda mantener la aplicación. En muy raras ocasiones una misma aplicación es mantenida por su autor original.
- Los estándares de programación mejoran la legibilidad del código, al mismo tiempo que permiten su comprensión rápida.

3.6 Pruebas a la hoja de consulta de Pediatría

Las pruebas de software comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, por medio de pruebas sobre el comportamiento del mismo.

Los sistemas informáticos, programas y aplicaciones han crecido a niveles inimaginables en complejidad e interoperabilidad, con lo cual también se han incrementado las posibilidades de defectos, a imple vistas insignificantes, pero que pudieran adquirir proporciones catastróficas.

Tareas a realizar para probar un software (PRESSMAN Roger S «Ingeniería.de.software.enfoque.practico.pdf» 2004):

1. Diseño de las pruebas

Esto es, identificación de la técnica o técnicas de pruebas que se utilizarán para probar el software. Distintas técnicas de prueba ejercitan diferentes criterios como guía para realizar las pruebas.

2. Generación de los casos de prueba

Los casos de prueba representan los datos que se utilizarán como entrada para ejecutar el software a probar. Más concretamente los casos de prueba determinan un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Como veremos posteriormente, cada técnica de pruebas proporciona unos criterios distintos para generar estos casos o datos de prueba. Por lo tanto, durante la tarea de generación de casos de prueba, se han de confeccionar los distintos casos de prueba según la técnica o técnicas identificadas previamente. La generación de cada caso de prueba debe ir acompañada del resultado que ha de producir el software al ejecutar dicho caso (como se verá más adelante, esto es necesario para detectar un posible fallo en el programa).

3. Definición de los procedimientos de la prueba

Esto es, especificación de cómo se va a llevar a cabo el proceso, quién lo va a realizar y cuándo.

4. Ejecución de la prueba

Aplicando los casos de prueba generados previamente e identificando los posibles fallos producidos al comparar los resultados esperados con los resultados obtenidos.

5. Realización de un informe de la prueba

Con la realización de las pruebas se realiza un informe de los resultados que se evidenciaron a la hora de la ejecución.

Existen diversos tipos y estrategias de pruebas de software, pero las realizadas en este capítulo son las que están definidas en la metodología utilizada en la solución del problema, AUP-UCI.

3.6.1 Niveles de pruebas

A continuación, una breve descripción de cada uno de los niveles de las pruebas (Sánchez 2013):

Pruebas Unitarias o de Componente: este tipo de pruebas son ejecutadas normalmente por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada. Adicionalmente, Las pruebas sobre componentes unitarios, suelen denominarse pruebas de módulos o pruebas de clases, siendo la convención definida por el lenguaje de programación la que influye en el término a utilizar. Por último, es importante que todas las funcionalidades de cada componente unitario sean cubiertas, por al menos, dos casos de prueba, los cuales deben centrarse en probar al menos una funcionalidad positiva y una negativa.

Pruebas de Integración: este tipo de pruebas son ejecutadas por el equipo de desarrollo y consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta.

Pruebas de Sistema: este tipo de pruebas deben ser ejecutadas idealmente por un equipo de pruebas ajeno al equipo de desarrollo, una buena práctica en este punto corresponde a la tercerización de esta responsabilidad. La obligación de este equipo, consiste en la ejecución de actividades de prueba en donde se debe verificar que la funcionalidad total de un sistema fue implementada de acuerdo a los documentos de especificación definidos en el proyecto. Los casos de prueba a diseñar en este nivel de pruebas, deben cubrir los aspectos funcionales y no funcionales del sistema. Para el diseño de los casos de prueba en este nivel, el equipo debe utilizar como bases de prueba entregables tales como: requerimientos iniciales, casos de uso, historias de usuario, diseños, manuales técnicos y de usuario final, etc.

Pruebas de Aceptación: Independientemente de que se haya tercerizado el proceso de pruebas y así la firma responsable de estas actividades haya emitido un certificado de calidad sobre el sistema objeto de prueba, es indispensable, que el cliente designe a personal que haga parte de los procesos de negocio para la ejecución de pruebas de aceptación, es incluso recomendable, que los usuarios finales que participen en este proceso, sean independientes al personal que apoyó el proceso de desarrollo. Cuando las pruebas de aceptación son ejecutadas en instalaciones o ambientes proporcionados por la firma desarrolladora se les denominan pruebas Alpha, cuando son ejecutadas desde la infraestructura del cliente se les denomina pruebas Beta.

Pruebas unitarias

El nivel utilizado para las pruebas fue el de las pruebas unitarias, a continuación, se muestra una gráfica que resume los métodos escogidos y un breve resumen de lo que realiza cada uno.

Tabla 9 Métodos a los que se les realizaron las pruebas

Nombre del método	Hoja de Consulta	Descripción
guardarHoja()	Pediatría	Método que se encarga de realizar todo el proceso de guardar la información de la hoja en la base de datos y registrarla en el sistema como una hoja de consulta disponible para su uso.
crearHoja()	Pediatría	Método que se encarga de mostrar al usuario la interfaz de la hoja de consulta en la cual puede ingresar todos los datos de la atención al paciente para que queden registrados, en caso de que el paciente no tenga una creada ya.
verHoja()	Pediatría	Método que permite al usuario del sistema seleccionar una hoja de consulta de la lista de pacientes para así poder realizar las acciones pertinentes sobre esta.

3.6.2 Métodos o modelos de pruebas aplicados

Las pruebas basadas en modelos requieren la derivación sistemática y posiblemente automática de las pruebas a partir de modelos. Los métodos de derivación de pruebas conocidos de las pruebas de software convencionales, pueden ser aplicados a los modelos UML. Las dos técnicas básicas en las pruebas de software son las pruebas de caja negra y las pruebas de caja blanca. Los principios de las pruebas de caja negra y caja blanca están bien detallados, por ejemplo, en los libros de Beizer (Beizer 1995) y Myers (Myers 2004).

Además de estas técnicas, existen algunos métodos para la generación automática de casos de prueba a partir de descripciones formales (Nebut, Fleurey, Traon, Jezequel 2006). Estos métodos pueden ser aplicados en modelos UML si los modelos son ejecutables y si existe una semántica formal para el subconjunto UML usado.

Pruebas de caja negra

También suelen ser llamadas funcionales y basadas en especificaciones. En ellas se pretende examinar el programa en busca de que cuente con las funcionalidades que debe tener y como lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos.

Resultados de las pruebas de caja negra

Las pruebas de caja negra fueron realizadas para asegurar el buen funcionamiento de la interfaz. Se realizaron en 3 iteraciones de las cuales en la primera se comprobó el funcionamiento general del software donde se produjeron 10 no conformidades de las cuales todas fueron corregidas en la segunda iteración y posteriormente en la última iteración se comprobó que se hayan resuelto todas las no conformidades y que las modificaciones hechas no hayan afectado los demás segmentos de la aplicación.

Pruebas de caja blanca

También suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones y bloques. que han sido ejecutados por los casos de prueba.

Resultados de las pruebas de caja blanca

Para la realización de las mismas fueron implementados una cantidad aproximada de 20 casos de pruebas de los cuales 10 aportaron no conformidades los cuales fueron corregidos instantáneamente. Después de concluidas las correcciones adecuadas se volvieron a ejecutar las pruebas donde quedó plasmado el funcionamiento correcto de la solución desarrollada.

Conclusiones Parciales Capítulo 3

En este capítulo se especificó el modelo de datos de la solución. Se diseñaron los diagramas de componente y despliegue, se identificaron las principales clases que deben ser definidas para el buen funcionamiento del sistema, además se realizó el diagrama de paquetes donde quedaron plasmadas todas las clases que se utilizaron durante el proceso de desarrollo de la aplicación. Se

realizaron las pruebas de software pertinentes a través de las cuales se pudo corregir errores existentes en la solución desarrollada.

Conclusiones

Una vez finalizada la presente investigación, se ha cumplido con el objetivo y las tareas propuestas, por lo que se concluye:

- Los sistemas estudiados a pesar de brindar soluciones avanzadas no son los más factibles a utilizar por concepto de costos, requerimientos y flexibilidad.
- El estudio de los procesos del negocio permitió definir las funcionalidades del sistema donde las de mayor relevancia están asociadas a la gestión de las hojas de consulta.
- Se realizó el diseño de la solución en correspondencia con la arquitectura propuesta, obteniéndose un sistema robusto y flexible capaz de gestionar la información relacionada con la atención del paciente en el área de consulta pediatría.
- Se desarrolló la hoja de consulta de Pediatría del módulo Consulta Externa para el Sistema de Información Hospitalaria XAVIA HIS, que facilitará la gestión de información en esta área de las instituciones hospitalarias.
- Con el desarrollo de este módulo el sistema XAVIA HIS ganará en valor añadido lo que influirá positivamente en el desarrollo del centro, la Universidad y del país en general.

Recomendaciones

Se recomienda:

- Realizar un despliegue inmediato en instituciones hospitalarias con el objetivo de incluir estas nuevas funcionalidades.
- Agregar una funcionalidad que permita exportar la hoja de consulta de Pediatría a una hoja de consulta general cuando el paciente cumpla la mayoría de edad que deja de ser atendido por pediatras y pasa a la medicina general.

Referencias Bibliográficas

- ATIAGA, E., PAOLA, C., QUEZADA, R. y DAVID, S., 2014. Artículo Científico.-Análisis comparativo de Frameworks JSF 2.0: Icefaces, Primefaces y Richfaces; para la implementación en el desarrollo del sistema de gestión de proyectos ambientales de la empresa KAYMANTA. ,
- BARNES, D.J., KÖLLING, M. y BRENTA, B.I., 2007. *Programación orientada a objetos con Java*. S.l.: Pearson Educación.
- BAZÁN, P., 2008. AJAX: un análisis tecnológico y posibilidades metodológicas. *X Workshop de Investigadores en Ciencias de la Computación*. S.l.: s.n.,
- CABELLO MORALES, E., 2001. Calidad de la Atención Médica: ¿ Paciente o cliente? *Revista médica heredia*, vol. 12, no. 3, pp. 96–99.
- CAÑEDO ANDALIA, R., NODARSE RODRÍGUEZ, M., CRUZ FONT, J., GERMÁN HECHAVARRÍA, I., CELORRIO ZARAGOZA, I. y GUERRERO PUPO, J.C., 2016. Papel de la colaboración científica en el impacto de la investigación en salud de Cuba en el contexto de América Latina. *Revista Cubana de Información en Ciencias de la Salud*, vol. 27, no. 1, pp. 56–74.
- CENTRO DE INFORMÁTICA MÉDICA., 2018. *Especificación de requisitos de software*. 5 febrero 2018. S.l.: s.n.
- CERRITOS, A., FERNÁNDEZ, F. y GÁTICA, F., 2003. Sistema de Información Hospitalaria. *México, DF*,
- Citas | Fidel soldado de las ideas. [en línea], 2018. [Consulta: 27 junio 2018]. Disponible en: <http://www.fidelcastro.cu/es/citas-sobre/Ni%C3%B1ez>.
- CORDERO, J.J.T., 2014. *La guía definitiva del diseño web: Html, xhtml, css y herramientas de diseño*. S.l.: Juan Jesús Tortajada Cordero.
- DE, P.A.A.T., QUIROZ, C.A.H., ANAYA, I.D.P. y LOZANO, L.D., 2018. EXPERIMENTACIÓN Y EVALUACIÓN DEL FRAMEWORK JBOSS AOP. ,
- Definiciones de biblioteca Faces para controles JavaServer Faces de terceros. [en línea], 2009. [Consulta: 20 junio 2018]. Disponible en: <http://www.ibm.com/developerworks/ssa/rational/library/09/faceslibrarydefinitionrationalapplicationdeveloper/index.html>.
- DESARROLLOWEB.COM, 2018. JSF - Java Server Faces. *DesarrolloWeb.com* [en línea]. [Consulta: 20 junio 2018]. Disponible en: <http://www.desarrolloweb.com/articulos/2380.php>.
- Diagramas de Componentes | LENGUAJE DE MODELADO UNIFICADO - UML. [en línea], 2018. [Consulta: 26 junio 2018]. Disponible en: http://stadium.unad.edu.co/ovas/10596_9836/diagramas_de_componentes.html.
- Enterprise JavaBeans. [en línea], 2018. [Consulta: 20 junio 2018]. Disponible en: <https://users.dcc.uchile.cl/~jbarrios/J2EE/node44.html>.
- ExpDX Planes. *ExpDX*[en línea], 2018. [Consulta: 24 junio 2018]. Disponible en: <https://www.expdx.com/planes/>.
- FLOWER, 2010. Java Foundations: Java - Estándares de programación. *Java Foundations* [en línea], [Consulta: 26 junio 2018]. Disponible en: <http://javafoundations.spot.com/2010/07/java-estandares-de-programacion.html>.

- GENTA MESA, G., 2006. Anotaciones para una historia de la pediatría y la puericultura. *latreia*, vol. 19, no. 3, pp. 296–304.
- GILARDON, A., DEL PINO, M., DI CANDIA, A., FANO, V., KRUPITZKY, S., FERNÁNDEZ, M.I. y ORAZI, V., 2004. El desarrollo del niño: Una definición para la reflexión y la acción. *Archivos argentinos de pediatría*, vol. 102, no. 4, pp. 312–313.
- HALDER, R., JANA, A. y CORTESI, A., 2016. Data leakage analysis of the hibernate query language on a propositional formulae domain. *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXIII*. S.l.: Springer, pp. 23-44.
- HERNÁNDEZ PÉREZ, R.R., ROMERO RUÍZ, M.J. y ROSALES HERNÁNDEZ, C.E., 2015. *Cumplimiento de llenado de las hojas de Historia Clínica Perinatal (HCP) de las pacientes atendidas en el Hospital Bertha Calderón Roque en el I Semestre 2012 y 2013* [en línea]. other. S.l.: Universidad Nacional Autónoma de Nicaragua, Managua. [Consulta: 27 junio 2018]. Disponible en: <http://repositorio.unan.edu.ni/1354/>.
- HITPASS, B., 2017. *BPM: Business Process Management: Fundamentos y Conceptos de Implementación 4a Edición actualizada y ampliada*. S.l.: Dr. Bernhard Hitpass.
- IBM Knowledge Center - Aplicaciones Java SE y Java EE. [en línea], 2018. [Consulta: 20 junio 2018]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SSQP76_7.5.0/com.ibm.dserver.res/Content/Business_Rules/_pubskel/Infocenter_Primary/ps_DS_Rule_Execution_Server1444.html.
- IBM Knowledge Center - Arquitectura JPA (Java Persistence API). [en línea], 2018. [Consulta: 20 junio 2018]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/cejb_persistence.html.
- Ingenieria.de.software.enfoque.practico..pdf*, 2004. S.l.: s.n.
- JavaServer Facelets. *Scribd* [en línea], 2018. [Consulta: 20 junio 2018]. Disponible en: <https://es.scribd.com/document/112815385/JavaServer-Facelets>.
- JBoss Developer Studio. [en línea], 2018. [Consulta: 21 junio 2018]. Disponible en: <https://www.redhat.com/es/technologies/jboss-middleware/developer-studio>.
- JRE (Java Runtime Environment). [en línea], 2018. [Consulta: 20 junio 2018]. Disponible en: <http://www.ayudabitorrent.com/jre>.
- LAÍN ENTRALGO, P., 1978. Historia de la medicina. ,
- LARMAN, C., 2004. GRASP: Más patrones para asignar responsabilidades. *L. Hernández MONTENEGRO, RODRÍGUEZ Y SALAZAR: Uso de patrones de diseño de software*, vol. 59.
- Metodologia UCI.pdf*, 2018. S.l.: s.n.
- MÓDULO CONSULTA EXTERNA DEL SISTEMA DE INFORMACIÓN HOSPITALARIA alas HIS.pdf*, 2018. S.l.: s.n.
- NARVÁEZ COELLO, J.I., 2014. *Guía de las mejores prácticas administrativas, seguridad y alta disponibilidad, caso de estudio: PostgreSQL*. B.S. thesis. S.l.: Pontificia Universidad Católica del Ecuador.

- OTERO, P., 2011. Beneficios y riesgos relacionados con el uso de la historia clínica electrónica. *Archivos argentinos de pediatría*, vol. 109, no. 6, pp. 476–477.
- PEDIATRIA KIDS, 2018. Pediatría. *Pediatría* [en línea]. [Consulta: 24 junio 2018]. Disponible en: <http://www.pediatriakids.com/about-us/questions/>.
- RUEDA, E.P. y QUEVEDO, F.J.L., 2002. *El pediatra eficiente*. S.I.: Ed. Médica Panamericana.
- SAM Pediatría - Sistema Administrativo Médico | Software Mas - Software para Médicos Software para Médicos. [en línea], 2018. [Consulta: 24 junio 2018]. Disponible en: <https://softwaremas.com/index.php/sam-pediatria/>.
- SÁNCHEZ, P.J.Z., 2013. Niveles de Prueba del Software. *PRUEBAS DE SOFTWARE* [en línea]. [Consulta: 27 junio 2018]. Disponible en: <https://pruebasdelsoftware.wordpress.com/2013/01/21/niveles-de-prueba-del-software/>.
- Seam - Contextual Components. [en línea], 2018. [Consulta: 20 junio 2018]. Disponible en: https://docs.jboss.org/seam/2.2.2.Final/reference/en-US/html_single/.
- TABARES, R.B., 2011. Patrones Grasp y Anti-Patrones: un Enfoque Orientado a Objetos desde Lógica de Programación. *Entre Ciencia e Ingeniería*, no. 8, pp. 161–173.
- TURABIÁN FERNÁNDEZ, J.L. y PÉREZ FRANCO, B., 2004. La historia clínica electrónica: ¿comer sopa con tenedor? *Cuad. gest. prof. aten. prim.* (Ed. impr.), pp. 175–188.

Glosario de Términos

Opensource: Fuente abierta o código abierto (open source) se refiere a cualquier programa cuyo código fuente se pone a disposición para su uso o modificación, conforme los usuarios u otros desarrolladores lo consideren conveniente.

Framework: En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado.

Anexos