



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

EXPEDIENTE CLÍNICO DE ENFERMERÍA

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTAN:

DIEGO ALEJANDRO BENITEZ LÓPEZ

GABRIEL OCTAVIO HERNÁNDEZ TORRES

LUIS FERNANDO JR PRADO FLORES

HUGO ENRIQUE SANTIAGO VELASCO

DIRECTOR: M.I. JUAN CARLOS ROA BEIZA

Ciudad Universitaria, México D.F., Noviembre 2015





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

AGRADECIMIENTOS.

A mis padres Alejandra y Francisco, por estar siempre conmigo, por todo lo que me han enseñado y por brindarme toda la confianza y el apoyo necesario para terminar este proyecto.

A mis hermanos Metztlí y Miguel por acompañarme en todos los momentos difíciles, por nunca dudar de mí y apoyarme en todas las decisiones que he tomado.

A mis amig@s por todos los años de amistad y por los buenos momentos que hemos pasado juntos.

A Luis, por enseñarme el lado más bonito de la vida, por estar día a día junto a mí en las buenas y malas. Y gracias a tus padres y hermanos por aceptarme como un miembro más de su familia.

Al Programa de Apoyo a la Titulación, por brindarme el apoyo y la ayuda necesaria para la elaboración de este trabajo.

A mis amigos del PAT, Luis Prado, Hugo Santiago y Gabriel Hernández por acompañarme en este viaje y hacer más divertido este proyecto.

Al M.I. Juan Carlos Roa Beiza, por su apoyo y dedicación en la elaboración de este trabajo.

Diego Alejandro Benítez López

AGRADECIMIENTOS

A mis padres, por su amor y apoyo continuo,

A mis hermanos, por su ejemplo y compañía,

A Dios por sus infinitas bendiciones,

A mis maestros, compañeros y amigos por sus enseñanzas,

Al PAT, a mis compañeros de tesis Diego, Hugo y Gabriel,

A la UNAM por su formación y liderazgo.

A la vida por esta oportunidad.

Luis Fernando Jr Prado Flores

Agradecimientos

A Dios, que me ha dado vida, herramientas para vencer obstáculos, sabiduría para la toma de decisiones y bendiciones para tener salud y alcanzar mis metas.

A mi Mamá Marina Santiago Velasco, que toda la vida ha estado conmigo, me ha criado, me ha cuidado. Que ha estado conmigo en las malas, en las peores y siempre me ha apoyado. Que me ha amado desde que me trajo al mundo y trato que nunca me hiciera falta nada. Que ha sufrido conmigo todos los pesares, me alentó para acabar con este sueño de ser ingeniero y ahora que he ejercido la profesión se la dedico en gran parte a ella. Te amo mamá.

A mi amada prometida Alejandra Olivares, que desde decidió ser mi novia, me ha motivado a ser mejor persona y alcanzar todas mis metas. Que ha estado conmigo cuando las cosas no andan bien y siempre escuche de sus labios un "si se puede. Y sobre todo que me aceptó para compartir su vida conmigo.

A mi tío Irineo que con mi mamá me crio, me ayudó a formarme y a respetar a las personas. Que a pesar de que se haya adelantado al cielo, lo tengo muy presente en mi vida diaria y es una de mis motivaciones para nunca dejarme vencer.

A mi hermana Lidia y sobrina ahijada Sarabi que estuvieron conmigo en momentos difíciles, y fueron de gran motivación para terminar la carrera y obtener un buen empleo.

A mis compañeros de Tesis Diego, Luis y Gabriel, que desde el principio de este proyecto estuvieron conmigo cumpliendo en tiempo y trabajo, y compartir conmigo experiencias personales y de amistad.

Al Programa de Apoyo a la Titulación PAT del palacio de minería, que me aceptaron en el equipo de trabajo para realizar la tesis y nos llenó de facilidades académicas.

A mi director de Tesis M. I. Juan Carlos Roa que nos asesoró y guio en todas las sesiones para perfeccionar el trabajo de Tesis.

A mi amada UNAM que me dio la oportunidad de estudiar, aprender lo mejor, crecer profesional y humanamente, y gritar con orgullo que gracias a ella soy un Puma.

Hugo Enrique Santiago Velasco

Agradecimientos

Principalmente quiero agradecer a mis padres y a mi abuelita por el apoyo que me brindaron a lo largo de mi educación, que gracias a sus comentarios me ayudaron a motivarme para continuar con mis estudios y poder seguir superando los retos de la vida.

También quiero agradecer a mis compañeros de tesis y amigos que son, que sin su trabajo, dedicación y compromiso esto no hubiera podido ser posible.

Me gustaría agradecer sinceramente al PAT (programa de apoyo a la titulación) y en especial al Ingeniero Roa Beiza Juan Carlos que gracias a su apoyo, dedicación y conocimiento pudimos realizar este trabajo de tesis de una manera más rápida y eficaz.

Quiero expresar mi profundo agradecimiento a la facultad de ingeniería por proporcionarme el conocimiento y habilidades para poder desarrollarme como profesional.

Por último quiero agradecer a todas las personas que me aportaron algo en esta formación, como fueron familiares, maestros, amigos y conocidos. Quiero compartir con todos ustedes este logro.

Hernández Torres Gabriel Octavio

ÍNDICE TEMÁTICO

Capítulo 1. Antecedentes

1.1 Introducción.....	1
1.2 Técnicas y procedimientos del manejo de información usados en hospitales.....	5
1.3 Descripción de las áreas de un hospital.....	12
1.4 Normatividad para elaboración de registros clínicos.....	21
1.5 Acotamiento del problema a resolver.....	30

Capítulo 2. Marco Teórico

2.1 Características, ventajas y desventajas de las bases de datos orientadas a objetos.....	31
2.2 Características, ventajas y desventajas del lenguaje de programación Groovy...	33
2.3 Características, ventajas y desventajas del gestor de base de datos Postgresql.	39
2.4 Características, ventajas y desventajas de la arquitectura cliente-servidor.....	45
2.5 Características, ventajas y desventajas de la metodología de desarrollo iterativo creciente y Extreme Programming-XP.....	52

Capítulo 3. Análisis y planteamiento del problema

3.1 Análisis del problema.....	58
3.2 Requerimientos generales del sistema.....	63
3.3 Planteamiento de la solución y posibles módulos.....	69
3.4 Posicionamiento del software a utilizar.....	75
3.5 Aplicación de la metodología Extreme Programming-XP al sistema.....	84

Capítulo 4. Diseño y construcción del sistema

4.1 Modelos del sistema.....	92
4.1.1 Ciclos de desarrollo.....	92
4.1.2 Diagrama de contexto.....	94
4.1.2 Diagramas de caso de usos.....	95
4.1.3 Diagrama Entidad-Relación.....	99
4.1.4 Diagrama de Clase.....	101
4.1.5 Diccionario de datos.....	107

4.2 Implementación de la base de datos.....	113
4.3 Diseño de la interfaz de usuario.....	133
4.4 Generación de pruebas y Mantenimiento.....	149
4.5 Generación de Reportes y análisis de resultados.....	163
Conclusiones	170
Bibliografía	173

CAPÍTULO 1.

INTRODUCCIÓN



1.1 Introducción

El expediente clínico es un documento normalizado de carácter obligatorio para el personal del área de la salud, sector público, social y privado que integra el Sistema Nacional de Salud¹. Esta herramienta es demasiado útil tanto para el paciente como para el médico que lo atiende. Se trata de un conjunto de datos personales e información, integrado por documentos escritos, gráficos, imagenológicos, electrónicos, magnéticos, electromagnéticos, ópticos, magneto-ópticos y de otras tecnologías que son registrados en diferentes momentos del proceso de atención médica y describe el estado actual de salud del paciente.

Por medio de la revisión de notas médicas y otras de carácter diverso con motivo de la atención médica, el personal del área de la salud, puede realizar diagnóstico, tratamiento y rehabilitación. Además se brinda la protección de los datos personales a los pacientes y se les otorga la confidencialidad del documento.

Los expedientes clínicos son propiedad de la institución o del prestador de servicios y deben ser conservados por un periodo mínimo de cinco años. También se usa como una herramienta de protección en el ámbito legal para el personal de la salud.

En la actualidad, el crecimiento en número y volumen de las historias clínicas ha llevado a la falta de espacio para su correcta conservación. Además se ha elevado considerablemente el costo de la gestión de los archivos² y se están deteriorando los procedimientos y cuidados de los pacientes. Se tienen incontables cajas con archivos impresos que contienen registro de la actividad médica, de cuidados especiales e indicaciones para cada uno de los pacientes que han sido tratados. Esto conlleva a que con frecuencia se pierdan los registros o se traslapen entre algunos pacientes. A la

¹ NORMA Oficial Mexicana NOM-004-SSA3-2012, Del expediente clínico

² Flor, Julio. La polémica gestión de las historias clínicas. 2015, julio 6, de El médico interactivo Sitio web:

<http://www.elmedicointeractivo.com/ap1/emiold/informes/gestion/historias.htm>



Capítulo 1. Introducción



hora de hacer los trámites clínicos, las personas esperan varios minutos mientras se realiza la búsqueda del expediente y se valida que se trate del paciente correcto.

Por lo anterior, nace la necesidad de desarrollar un sistema que ayude al personal administrativo del área de la salud a ser más ágiles en la búsqueda, archivamiento y control de expedientes, para evitar retrasos en procedimientos y en trámites de regla. Al mismo tiempo, el sistema ayudará a los médicos y enfermeros para fungir como una herramienta de respaldo legal, ya que se registrarán todos los cuidados especiales y de rutina en todos los turnos, así como las justificaciones del diagnóstico. Para los pacientes, el sistema de registro clínico será de gran utilidad ya que los tiempos de espera serán mucho menores, además estarán cargados una gran cantidad de catálogos para evitar el error humano.

En resumen, la principal motivación para el desarrollo del sistema del registro clínico, es mejorar los servicios de salud, proporcionar al personal de la salud herramientas para la mejora de procedimientos que dicta la normatividad, usar el registro clínico como respaldo legal, mejorar los procedimientos administrativos y principalmente ahorrar tiempo y costos de papelería así como de almacenamiento.

El presente trabajo de tesis se divide en cuatro capítulos, los cuales presentan de forma global el estudio de los métodos actuales de llenado y gestión de un expediente clínico. Así como presentar mejoras en el llenado, manipulación, almacenamiento y control del expediente clínico de forma electrónica en línea. El primer capítulo describe los antecedentes y la forma en que se trabaja actualmente.

El segundo capítulo describe de forma general algunos de los conceptos básicos que se usaron para realizar la presente tesis, estos conceptos están basados en definiciones del lenguaje de programación **Groovy** (nombre de lenguaje de programación), arquitectura Cliente-Servidor, bases de datos **PostgreSQL** (Nombre de manejador de base de datos) y las metodologías de desarrollo iterativo creciente y **Extreme Programming X-P** (Programación Extrema).



Capítulo 1. Introducción



El tercer capítulo describe el análisis del problema, los requerimientos para la construcción del sistema y la solución usando las tecnologías descritas en el capítulo anterior.

En el cuarto capítulo se implementaron las tecnologías para la construcción del sistema del expediente clínico. Se detalló de manera puntual el diseño y la documentación del sistema por medio de diagramas. Para optimizar el funcionamiento de dicho sistema se programaron pruebas y mantenimiento, además de la generación de reporte y análisis de resultados. Por último, se presentan las conclusiones finales del presente trabajo de tesis.

El principal objetivo que se pretende alcanzar al final del presente trabajo de tesis es tener un sistema vía web de expediente clínico que auxilie al personal de enfermería, en sus diferentes turnos y registrar la hoja de expediente clínico en los siguientes rubros:

- Agilizar la consulta de la información por parte del personal administrativo. En lugar de buscar los expedientes en los archiveros de forma manual, el sistema hará búsquedas mucho más rápidas sobre la base de datos.
- Centralizar la información de todos los pacientes del hospital en una base de datos. Siguiendo las normas de almacenamiento temporal de los expedientes clínicos, se implementará una base de datos en PostgreSQL donde estarán todos los datos tanto del personal de la salud como de los pacientes, restringiendo el acceso a ella dependiendo del puesto.
- Dar una mejor imagen hacia el cliente que usa de los servicios del hospital. Se buscará agilizar los procesos administrativos usando las tecnologías actuales y mejorar los servicios en las áreas médicas de un hospital para comodidad de los pacientes.
- Proveer de catálogos al usuario para evitar en lo posible el error humano. La base de datos, tendrá una serie de catálogos basados en los estándares de



Capítulo 1. Introducción



salud para que el personal administrativo y médico evite escribir y que se enfoque a llenar campos predefinidos.

- Generar reportes de forma automatizada de todas las actividades que realiza el departamento de enfermería y área médica. Mediante el llenado de los formularios y de los **Checklist (Lista de comprobación)**, se tendrá la opción de imprimir o guardar en documentos digitales la información capturada para su administración.
- Generar expediente clínico en caso de ser necesario un seguimiento de un paciente en particular que tenga algún reclamo judicial contra el hospital o médico. En caso de un proceso Judicial, se podrá imprimir el expediente clínico del paciente, empleando las normas oficiales, y entregarlo a las autoridades competentes.
- Llevar un control sobre la medicación de cada paciente que requiera algún medicamento controlado. Para el caso de los medicamentos controlados, se tendrá un registro e inventario especial basado en las normas y estándares nacionales.
- Llevar una bitácora del ingreso de los usuarios al sistema para asentar el tratamiento y medicamentos suministrados al paciente. Se utilizará una bitácora para llevar el control del personal que tiene la facultad de suministrar medicamentos a los pacientes, así como registrar quién es la persona que realiza esta actividad y usarla para aclaraciones tanto internas como legales.
- Ayudar al ahorro en papelería y reducir el archivo de documentos que usa normalmente el hospital. Como el registro clínico lo realizará un sistema electrónico, sólo se imprimirá lo necesario para el ahorro de papel y almacenamiento.
- Generar reportes de las áreas solicitadas o de personal que requiera ser evaluado para su promoción. Algunos reportes serán llenados por el sistema para informar de forma estadística el comportamiento del personal médico para su evaluación.



1.2 Técnicas y procedimientos del manejo de información usados en hospitales.

Un procedimiento de atención médica³ es un plan de acción destinado a lograr un resultado en el cuidado de las personas con problemas de salud, tiene la intención de determinar, medir o inferir una condición o parámetro de los pacientes, empleando para ello normas y técnicas que permitirán llevar un registro de las actividades realizadas por el personal médico o de enfermería que ayudarán a determinar el diagnóstico de los usuarios de una unidad médica.

El procedimiento de atención médica⁴ es dividido en varias etapas, como se muestra en la figura 1.2.1, las cuales van desde la admisión del paciente en la unidad hospitalaria, hasta el alta médica del mismo y su seguimiento y evaluaciones posteriores.

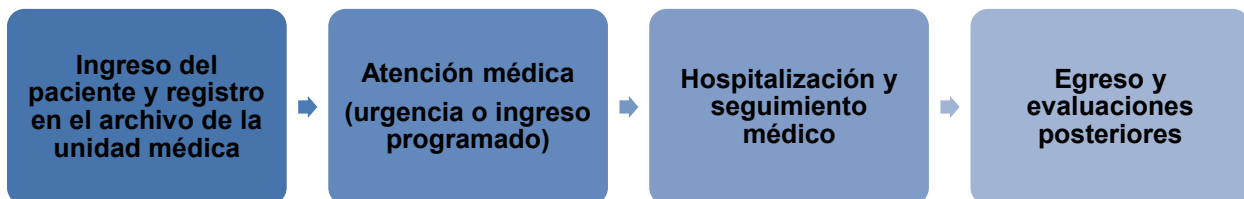


Figura 1.2.1 Etapas de la atención médica.

En cada una de las etapas, el personal administrativo, médico o de enfermería realiza, el registro y actualización de los datos e historia clínica del paciente, con el objetivo de mantener un control de la información personal, signos vitales, diagnóstico, control de medicamentos, estudios de laboratorio y gabinete, etc. creando así un registro documental con la información detallada de una persona concreta, para que sea consultada por todos los miembros que participan en el proceso de atención médica.

³ Selman-Housein Abdo, Guía de acción para la excelencia en la atención médica, Científico Técnica, 2002.

⁴ Manual de procedimientos del servicio de urgencias para hospitales generales, instituto de salud del estado de México,



Ingreso del paciente y registro en la unidad médica.

Un usuario que puede estar sano o enfermo ingresa al hospital, en el área de vigencia y archivo clínico, se le proporciona una solicitud que deberá llenar con su información personal, para ser registrado como paciente del hospital y brindarle posteriormente el servicio médico solicitado.

En esta área, mediante la presentación de documentos de identificación, como son acta de nacimiento, identificación oficial, carnet de servicio médico, etc. el personal administrativo obtendrá mediante el llenado de la solicitud, la información correspondiente al usuario, que podremos clasificar de la siguiente forma:

- Información personal (nombre completo, edad, sexo, estado civil, dirección, teléfono, número de seguridad social, etc.)
- Información complementaria (Información sobre familiares, Contactos de emergencia, etc.)
- Breve historia médica (alergias, medicamentos consumidos, cirugías previas, etc.)
- Patologías (presión arterial, diabetes, asma, problemas cardiovasculares, convulsiones, cáncer, etc.)
- Historial social (fumador, ingesta de alcohol, uso de drogas, etc.)
- Otra información (creencias religiosas, información complementaria)
- Padecimientos (motivos de visita al hospital, dolores, lesiones, etc.)

Esta información es almacenada de forma impresa en los archivos del hospital y es el personal de vigencia y archivo clínico, el encargado de almacenar, buscar, organizar y transportar la información proporcionada por el usuario (Expediente), a los diferentes servicios médicos y de especialidad del hospital para que el personal médico y de enfermería realicen las valoraciones necesarias y seguimientos necesarios de los pacientes.



Atención Médica

En esta etapa, el personal médico y de enfermería realiza el diagnóstico del paciente, registrando para ello en el expediente proporcionado por el área de vigencia y archivo clínico, la información obtenida durante este proceso.

La información que puede obtenerse en esta etapa es la siguiente:

- Signos vitales (temperatura corporal, frecuencia cardiaca, frecuencia respiratoria, presión arterial)
- Parámetros del paciente (peso, estatura y talla)
- Parámetros adicionales (saturación de oxígeno, niveles de glucosa, etc.)
- Diagnóstico (información sobre la evaluación del médico, tratamiento, estudios solicitados, cirugías o intervenciones solicitadas, etc.)

Esta información será actualizada por el personal médico y de enfermería en el expediente del paciente y almacenada en el archivo clínico del hospital, para futuras actualizaciones y consultas.

Hospitalización y seguimiento médico

En este proceso, los médicos en turno y el personal de enfermería, realizan un control y seguimiento de los pacientes que en la etapa de atención médica fueron diagnosticados con algún padecimiento que requería hospitalización e intervención quirúrgica.

Para ello el personal en turno utilizará un formato impreso conocido como expediente clínico de enfermería⁵, los cuales son la evidencia escrita de los cuidados otorgados al paciente, son un medio de comunicación y coordinación entre el personal que labora en el hospital y su realización correcta permite la continuidad de los cuidados y la

⁵ Norma oficial mexicana nom-004-ssa3-2012, del expediente clínico, Secretaría de Salud, México, 2012



Capítulo 1. Introducción



seguridad del paciente, estos expedientes están disponible en las camas de los pacientes, de esta manera el personal de enfermería de diferentes turnos puede realizar la consulta, actualización y seguimiento de los pacientes hospitalizados.

El personal médico, al momento de realizar su recorrido de un extremo de las camas de la sala al otro extremo, puede verificar la evolución de los pacientes, revisando la información disponible en el expediente clínico de enfermería, para posteriormente emitir alguna observación referente al cuidado de los pacientes.

Estos expedientes clínicos de enfermería deben ser actualizados con la información proporcionada por los médicos, personal de enfermería u otros y deben estar disponibles para su consulta, pues estos expedientes, son usados para el control de calidad de los hospitales, así como para realizar un seguimiento de pacientes que tienen algún reclamo de tipo judicial contra el hospital o el médico.

En el expediente clínico de enfermería, se registra los siguientes rubros; como se aprecia en la figura 1.2.2.

- Datos generales del paciente
- Alergias y comorbilidad
- Diagnósticos e intervenciones
- Estudios de laboratorio y gabinete
- Control de líquidos
- Evacuaciones
- Control de medicamentos
- Dietas
- Valoración de enfermería
- Tratamiento e indicaciones médicas
- Escalas de dolor
- Patrones de sueño



Capítulo 1. Introducción



PLAN INTEGRAL DE ENFERMERÍA

NOMBRE _____
 N° EXP. _____ FECHA DE NAC: _____
 EDAD _____ GENERO _____
 SERVICIO _____ N° CAMA _____
 DIAGNOSTICO MEDICO _____

DATOS COMPLEMENTARIOS
 FECHA _____ DIAS DE HOSPITALIZACION _____ PESO _____ TALLA _____ GRUPO Y Rh _____
 MEDICO RESPONSABLE _____ DIABETICO () HIPERTENSO () OTRO _____
 ALERGICO () ESPECIFICAR _____ PERIM. PC _____ PT _____ PA _____

PERCEPCION Y MANEJO DE SALUD	MATUTINO	VESPERTINO	NOCTURNO
CONOCIMIENTO DE SU ENFERMEDAD			
CONOCIMIENTO PARA EL AUTOCUIDADO			
ADHERENCIA AL TRATAMIENTO 1 H			

REGISTROS CLINICOS

HORA	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6
FRECUENCIA CARDIACA																								
FRECUENCIA RESPIRATORIA																								
PRESION ARTERIAL (mmHg)	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
TEMPERATURA (°C)																								
PIEL: (NL) (C) (P) (I) (D) (S) (H) (R)																								
LLENADO CAPILAR (seg)																								
OXIMETRIA DE PULSO (% SAT.)																								
PRESION VENOSA CENTRAL (PVC)																								
GLICEMIA CAPILAR (mg/dl)																								
TIPO DE RESPIRACION: (E), (M)																								
DISPOSITIVO: (MNI),(PN), (N), (MNB)																								
CAMBIOS DE POSICIÓN (LI) (LD)(DD) (DV) (S) (SF) (F) (T) (P)																								
PUPILA DER.	TAMAÑO (dm pupilar)																							
	RESPUESTA																							
PUPILA IZQ.	TAMAÑO (dm pupilar)																							
	RESPUESTA																							
ESCALA DE GLASGOW																								
CRISIS CONVULSIVAS (T), (C), (TC) (F), (G)																								

COGNITIVO - PERCEPTUAL

TURNO	MATUTINO	VESPERTINO	NOCTURNO
LA ESCALA DEL DOLOR			
REFIERE DOLOR (REGION ANATOMICA)			
TIPO			
DURACIÓN (C) , (IT)			
FACTORES			
REPERCUSIÓN EN AVD			

NUTRICIONAL - METABOLICO

Tubo de dieta:	Tubo de dieta:	Tubo de dieta:
Alimentos que ingiere	Alimentos que ingiere	Alimentos que ingiere

VALORACIÓN DEL RIESGO: ÚLCERAS POR PRESION	MATUTINO			PUNTAJE	VESPERTINO			PUNTAJE	NOCTURNO			PUNTAJE
	RIESGO	RIESGO	RIESGO		RIESGO	RIESGO	RIESGO		RIESGO	RIESGO	RIESGO	
ESCALA DE BRADEN BERGSTROM	ALTO	MEDIANO	BAJO		ALTO	MEDIANO	BAJO		ALTO	MEDIANO	BAJO	

Figura 1.2.2 Hoja de control de enfermería.

Estos formatos incluyen rubros para que el personal de tres turnos diferentes de enfermería, así como el personal médico registre la información necesaria para llevar el control, para ello el personal debe acudir a la cama de cada paciente a revisar el expediente y en ese lugar, realizar la consulta, actualización y seguimiento anotando la



Capítulo 1. Introducción



información en los formatos y devolviéndolos de nueva cuenta en la cama del paciente para futuras consultas y actualizaciones, como se muestra en la figura 1.2.3.

PATRON ACTIVIDAD EJERCICIO	MATUTINO				VESPERTINO				NOCTURNO			
	RIESGO		PUNTAJE		RIESGO		PUNTAJE		RIESGO		PUNTAJE	
VALORACION RIESGO DE CAIDA	ALTO	MED.	BAJO		ALTO	MED.	BAJO		ALTO	MED.	BAJO	
PATRON SEXUALIDAD Y REPRODUCCION	EMBARAZO SDG		FLUJO TRANSVAG.		HEMORRAGIA TRANSVAGINAL		ALTERACION EN MAMAS					
	FUM		ANTICONCEPTIVOS		INFECC. REL. CON EL SEXO		ALTERACION EN TESTICULOS					
PATRON ROL RELACIONES	M	V	N	PATRON SUEÑO Y REPOSO			M	V	N	PATRON AFRONTAMIENTO Y TOLERANCIA AL ESTRÉS		
Dificultad para oír				Sueño continuo						Agresivo		
Dificultad para hablar				Sueño discontinuo						Inquieto		
Dificultad para entender				Dificultad para dormir						Indiferente		
Dificultad visual				Deambulación						Ansioso		
Otro				Reposo relativo						Depresivo		
PATRON VALORES Y CREENCIAS				Reposo absoluto						Angustiado		
Manifiesta necesidad de apoyo espiritual				Otro						Otro		

DISPOSITIVOS	SITIO		N° DIAS	NOMBRE				CARACTERISTICAS/ OBSERVACIONES
	D	I		INSTALACION	CURACION	RETIRO	CAMBIO	
	Catéter corto periférico							
Catéter corto periférico								
Catéter largo								
Catéter subclavio								
Catéter bilumen								
Catéter trilumen								
Catéter Mahukar								
Catéter femoral (hemodiálisis)								
Catéter puerto								
Catéter diálisis (R). (T)								
Sonda nasogástrica / nasoyeyunal								
Sonda orogástrica								
Sonda endotraqueal								
Sonda vesical								
Sonda pleural								
Drenajes (especificar)								
Traqueostomía								
Ostomía (especificar)								
Vendajes (especificar)								
Aparato de yeso (especificar sitio)								
Tracción: (E.) (C)								
Otro:								

N° DIAS	MEDICAMENTOS		DOSIS	VIA	FRECUENCIA	HORARIO	OBSERVACIONES

Figura 1.2.3 Hoja de control de enfermería

Los expedientes clínicos de enfermería son elaborados bajo estrictas normas de calidad, lo que permite realizar un proceso eficiente y certero sobre la evolución de un paciente.

Dichos formatos, resultan ser complejos y pueden incurrir en errores humanos al momento de ser requisitados, además de dificultar la difusión entre los diferentes pisos



Capítulo 1. Introducción



a los cuales es trasladado el paciente, o entre las diferentes áreas del hospital, además provocan un volumen elevado de hojas impresas, provocando que los expedientes físicos de los pacientes sean de gran tamaño y de difícil acceso.

Es por ello que nace la necesidad de elaborar, un sistema de registro clínico de enfermería que permita centralizar en una base de datos la información de todos los pacientes del hospital, agilizando la consulta de la información y optimizando el archivo del hospital, en todas las etapas del proceso de atención médica

Egreso y evaluaciones posteriores

Una vez que el paciente es dado de alta, las hojas del expediente clínico de enfermería serán almacenadas en el expediente del paciente, registrando en formatos adicionales la información de:

- Fecha de alta médica
- Indicaciones médicas
- Futuras citas y seguimiento médico
- Recomendaciones

La información obtenida en todas las etapas de la atención médica, son resguardadas por el hospital en un archivo físico, por un lapso de 5 años, donde se encuentran disponibles para su consulta o actualización en caso de seguimiento médico una vez que el paciente deja la unidad médica.



1.3 Uso y seguridad de datos personales

Durante su proceso de atención, se obtienen datos personales (datos de identificación), esta información se utiliza para la identificación del paciente, prestación de servicios, trámites administrativos, análisis estadísticos etc. También se obtienen datos sensibles (datos de salud, creencias ideológicas, preferencia sexual), con la finalidad de apoyar el diagnóstico y tratamiento adecuado, creación, estudio, análisis, actualización y conservación del expediente clínico y trámites administrativos.

Mantener íntegra y segura la información de los pacientes es indispensable para las instituciones de salud, es por ello que nacen normas que permitan llevar un control y regulación sobre el uso y seguridad de los datos del paciente.

Carta de consentimiento informado⁶

Todos los pacientes tienen derecho a la libertad de decisión, la equidad, la dignidad, la seguridad jurídica, la salud y a la vida. Es por esto que la Asociación Médica Mundial a través de la Declaración de Helsinki publicó una serie de normas éticas para el uso de los datos personales de los pacientes, en las que destaca la protección y la confidencialidad, tomando las medidas necesarias para el resguardo de esta información. Esta declaración está orientada a médicos para que atiendan de manera correcta a sus pacientes.

El consentimiento informado, se define como la *"Facultad del enfermo válidamente informado y libre de coacción, para aceptar o no la atención médica que se le ofrezca o la participación en proyectos de investigación que se le propongan"*.

De acuerdo a la definición anterior, el paciente tiene el derecho de legitimar su posición ante el médico o prestador de servicios de salud. Entre los derechos de los pacientes está: recibir una atención médica conforme a los principios éticos vigentes.

⁶ http://www.facmed.unam.mx/sms/seam2k1/2002/ponencia_nov_2k2.html Consultada el 22-11-2015



Capítulo 1. Introducción



Dentro del consentimiento informado se tiene el derecho a la información que incluye los siguientes conceptos:

- Debe ser informado quién es el médico responsable de su atención y conocer la reglamentación y normatividad de los establecimientos de salud y en particular de los hospitales donde va a recibir atención, aplicables a su conducta como paciente.
- Ser informado objetivamente y con toda claridad, de todo lo relativo al proceso de su atención y particularmente lo relacionado con el programa de estudio y tratamiento, con el diagnóstico definitivo o el pronóstico de su padecimiento. Para esto, el médico debe verificar que el paciente haya comprendido la información que se le proporcionó.
- Para todo procedimiento o tratamiento, el paciente debe ser previamente informado, incluyendo los beneficios que se espera lograr, las molestias que le ocasionará, los riesgos significativos asociados, posibilidad de complicaciones y consecuencias adversas, así como la duración de las discapacidades que pudieran ocurrir.
- Conocer las expectativas de curación, control, mejoría o paliación de su enfermedad.
- Cuando solicite más información, asesorarlo con dicha información.
- Conocer los riesgos de tipo físico, emocional o de otra índole, así como y los beneficios esperados, de los proyectos de investigación en que se le proponga participar.
- Que se dé respuesta a sus preguntas y se aclaren sus dudas.

Las condiciones que requieren consentimiento informado son las siguientes:

- La hospitalización.
- La realización de una intervención quirúrgica.
- La práctica de cualquier procedimiento con fines de control de la fertilidad.
- La participación en protocolos de investigación.



Capítulo 1. Introducción



- La realización de procedimientos diagnósticos o terapéuticos que impliquen riesgo de producir secuelas, discapacidades o complicaciones.
- La práctica de procedimientos invasivos.
- El compromiso del paciente de cumplir con las normas hospitalarias.
- Procedimientos que produzcan dolor no justificado o sufrimiento moral.
- En las urgencias el consentimiento está implícito, a menos que el paciente haya expresado lo contrario.
- El paciente tiene derecho a rechazar tratamientos dirigidos a prolongar artificialmente su vida.

En la mayoría de los casos, el consentimiento informado debe otorgarse por escrito, en formatos oficiales, diseñados, con la firma de dos testigos, familiares y no familiares, con el propósito de prevenir quejas, demandas o conflictos legales.

El paciente también tiene derecho a que se tomen en cuenta sus preferencias en el ejercicio de su autonomía para tomar una decisión, particularmente si está consiente o si las dejó por escrito cuando lo estaba (testamento vital).

Cuando el paciente no esté en condiciones de otorgar su consentimiento, cualquier indicación relacionada con la abstención o suspensión de apoyo vital, debe ser informada y consensada con la familia o con el familiar legalmente responsable. Si existe desacuerdo con la familia, se puede recurrir a la intervención de un médico de su confianza, al Comité de Ética Hospitalario o al poder judicial, y en caso necesario trasladar al paciente a otra institución. Todo deberá quedar escrupulosamente registrado en el expediente clínico.

Para la investigación médica en que se utilice material o datos humanos identificables, como la investigación sobre material o datos contenidos en biobancos o depósitos similares, el médico debe pedir el consentimiento informado para la recolección, almacenamiento y reutilización. Podrá haber situaciones excepcionales en las que será imposible o impracticable obtener el consentimiento para dicha investigación. En esta



situación, la investigación sólo puede ser realizada después de ser considerada y aprobada por un comité de ética de investigación.

Debido a las características sobre el uso de datos personales de los usuarios de sistemas de salud, deben tomarse las siguientes consideraciones al momento de desarrollar el sistema del expediente clínico a medida de asegurar la información de los pacientes, la cual será propiedad de la institución de salud.

Seguridad de la información

Seguridad de la información se refiere a los procesos y metodologías que han sido diseñados implementados para proteger la información impresa, electrónica, o cualquier otra forma de información o datos confidenciales, privados y sensibles frente al acceso, uso, mal uso, divulgación, destrucción, modificación o alteración.

Amenazas

No sólo las amenazas que surgen de la programación y el funcionamiento de un dispositivo de almacenamiento, transmisión o proceso deben ser consideradas, también hay otras circunstancias que deben ser tomadas en cuenta.

Las amenazas pueden ser causadas por:

- **Usuarios:** causa del mayor problema ligado a la seguridad de un sistema informático. En algunos casos sus acciones causan problemas de seguridad, si bien en la mayoría de los casos es porque tienen permisos sobre dimensionados, no se les han restringido acciones innecesarias, etc.
- **Programas maliciosos:** programas destinados a perjudicar o a hacer un uso ilícito de los recursos del sistema. Es instalado (por inatención o maldad) en el ordenador, abriendo una puerta a intrusos o bien modificando los datos. Estos programas pueden ser un virus informático, un gusano informático, un troyano, una bomba lógica, un programa espía o spyware, en general conocidos como malware.



Capítulo 1. Introducción



- **Errores de programación:** La mayoría de los errores de programación que se pueden considerar como una amenaza informática es por su condición de poder ser usados como exploits por los crackers, aunque se dan casos donde el mal desarrollo es, en sí mismo, una amenaza. La actualización de parches de los sistemas operativos y aplicaciones permite evitar este tipo de amenazas.
- **Intrusos:** personas que consiguen acceder a los datos o programas a los cuales no están autorizados (crackers, defacers, hackers, script kiddie o script boy, viruxers, etc.).
- **Un siniestro** (robo, incendio, inundación): una mala manipulación o una mala intención derivan a la pérdida del material o de los archivos.
- **Personal técnico interno:** técnicos de sistemas, administradores de bases de datos, técnicos de desarrollo, etc. Los motivos que se encuentran entre los habituales son: disputas internas, problemas laborales, despidos, fines lucrativos, espionaje, etc.
- **Fallos electrónicos o lógicos de los sistemas informáticos en general.**
- **Catástrofes naturales:** rayos, terremotos, inundaciones, rayos cósmicos, etc.

A continuación se enumeran una serie de medidas que se consideran básicas para asegurar un sistema tipo, si bien para necesidades específicas se requieren medidas extraordinarias y de mayor profundidad:

- Utilizar técnicas de desarrollo que cumplan con los criterios de seguridad al uso para todo el software que se implante en los sistemas, partiendo de estándares y de personal suficientemente formado y concienciado con la seguridad.



Capítulo 1. Introducción



- Implantar medidas de seguridad físicas: sistemas anti incendios, vigilancia de los centros de proceso de datos, sistemas de protección contra inundaciones, protecciones eléctricas contra apagones y sobretensiones, sistemas de control de accesos, etc.
- Codificar la información: criptología, criptografía y criptociencia. Esto se debe realizar en todos aquellos trayectos por los que circule la información que se quiere proteger, no solo en aquellos más vulnerables. Por ejemplo, si los datos de una base muy confidencial se han protegido con dos niveles de firewall, se ha cifrado todo el trayecto entre los clientes y los servidores y entre los propios servidores, se utilizan certificados y sin embargo se dejan sin cifrar las impresiones enviadas a la impresora de red, tendríamos un punto de vulnerabilidad.
- Contraseñas difíciles de averiguar que, por ejemplo, no puedan ser deducidas a partir de los datos personales del individuo o por comparación con un diccionario, y que se cambien con la suficiente periodicidad. Las contraseñas, además, deben tener la suficiente complejidad como para que un atacante no pueda deducirla por medio de programas informáticos. El uso de certificados digitales mejora la seguridad frente al simple uso de contraseñas.
- Vigilancia de red. Las redes transportan toda la información, por lo que además de ser el medio habitual de acceso de los atacantes, también son un buen lugar para obtener la información sin tener que acceder a las fuentes de la misma. Por la red no solo circula la información de ficheros informáticos como tal, también se transportan por ella: correo electrónico, conversaciones telefónicas (VoIP), mensajería instantánea, navegación Internet, lecturas y escrituras a bases de datos, etc. Por todo ello, proteger la red es una de las principales tareas para evitar robo de información. Existen medidas que abarcan desde la seguridad física de los puntos de entrada hasta el control de equipos conectados, por



Capítulo 1. Introducción



ejemplo 802.1x. En el caso de redes inalámbricas la posibilidad de vulnerar la seguridad es mayor y deben adoptarse medidas adicionales.

- Redes perimetrales de seguridad, o DMZ, permiten generar reglas de acceso fuertes entre los usuarios y servidores no públicos y los equipos publicados. De esta forma, las reglas más débiles solo permiten el acceso a ciertos equipos y nunca a los datos, que quedarán tras dos niveles de seguridad.
- Tecnologías repelentes o protectoras: sistema de detección de intrusos - antispyware, antivirus, llaves para protección de software, etc.
- Mantener los sistemas de información con las actualizaciones que más impacten en la seguridad.
- Copias de seguridad e, incluso, sistemas de respaldo remoto que permiten mantener la información en dos ubicaciones de forma asíncrona.
- Controlar el acceso a la información por medio de permisos centralizados y mantenidos (tipo Active Directory, LDAP, listas de control de acceso, etc.). Los medios para conseguirlo son:
 - Restringir el acceso (de personas de la organización y de las que no lo son) a los programas y archivos.
 - Asegurar que los operadores puedan trabajar pero que no puedan modificar los programas ni los archivos que no correspondan (sin una supervisión minuciosa).
 - Asegurar que se utilicen los datos, archivos y programas correctos en/y/por el procedimiento elegido.



Capítulo 1. Introducción



- Asegurar que la información transmitida sea la misma que reciba el destinatario al cual se ha enviado y que no le llegue a otro y que existan sistemas y pasos de emergencia alternativos de transmisión entre diferentes puntos.
- Organizar a cada uno de los empleados por jerarquía informática, con claves distintas y permisos bien establecidos, en todos y cada uno de los sistemas o aplicaciones empleadas.
- Actualizar constantemente las contraseñas de accesos a los sistemas de cómputo, como se ha indicado más arriba, e incluso utilizando programa que ayuden a los usuarios a la gestión de la gran cantidad de contraseñas que tienen gestionar en los entornos actuales, conocidos habitualmente como gestores de identidad.
- Redundancia y descentralización.

Seguridad en el Sistemas

El plugin Spring Security⁷ simplifica la integración de Spring Security en aplicaciones Grails. El plug-in ofrece parámetros por defecto con muchas opciones de configuración de personalización. Casi todo es configurable o sustituibles en el complemento y en la propia Spring Security, lo que hace un amplio uso de las interfaces.

Cualquier aplicación basada en la web debe tener un mecanismo para la autenticación de usuarios y autoriza que hagan sus actividades definidas en el sistema. También en entornos corporativos generalmente autenticación y autorización se realiza contra una base de datos LDAP.

En el centro de Spring Security se encuentra Spring Security Core que es el usado en el sistema.

⁷ <http://grails.org/plugin/spring-security-core> consultada el 23/11/2015



Seguridad en la base de datos

La seguridad de la base de datos esta implementada en varios niveles:

- Protección de los ficheros de la base de datos. Todos los ficheros almacenados en la base de datos están protegidos contra escritura por cualquier cuenta que no sea la del súper usuario de Postgres.
- Las conexiones de los clientes al servidor de la base de datos están permitidas, por defecto, únicamente mediante sockets Unix locales y no mediante sockets TCP/IP.
- Las conexiones de los clientes se pueden restringir por dirección IP y/o por nombre de usuario mediante el fichero `pg_hba.conf` situado en `PG_DATA`.
- Las conexiones de los clientes pueden ser autenticadas mediante otros paquetes externos.
- A cada usuario de Postgres se le asigna un nombre de usuario y (opcionalmente) una contraseña. Por defecto, los usuarios no tienen permiso de escritura a bases de datos que no hayan creado.
- Los usuarios pueden ser incluidos en *grupos*, y el acceso a las tablas puede restringirse en base a esos grupos.

La seguridad del expediente clínico de enfermería a nivel sistema, será implementada con el plugin de grails Spring Security que ayuda al control de acceso, este consiste en la autorización, autenticación, autorización de acceso y auditoria, por medio de este plugin no se podrá acceder a ningún recurso del sistema sino ha iniciado sesión o esta no se encuentra vigente, tampoco se podrá acceder a módulos que no se les fueron asignados al dar de alta su cuenta, en el sistema serán implementados dos roles



Capítulo 1. Introducción



principales ROLE_ENFERMERIA(enfermero que realiza registros en el sistema) y ROLE_ENFERMERIA_JEFE_SUPERVISOR(enfermero jefe supervisor que firma los expedientes de conformidad).

El expediente clínico de enfermería, será implementado en Amazon Web Services el cual es un servicio de cómputo en la nube, se utilizará el servicio Elastic Beanstalk el cual proporciona todo lo necesario para la aplicación como deployar, monitorear, escalar y garantizar la seguridad por medio de las capas adicionales que implementa el mismo servicio como servidor proxy o detectar ataques comunes como la denegación de servicio, también se permite el firewall, es decir, los puertos de entrada y salida que estarán abiertos para la aplicación lo que da como resultado solo enfocarnos prácticamente en la lógica del sistema, para la base de datos se usó el servicio RDS que al igual que Elastic Beanstalk proporcionarán las características antes mencionadas.

1.4 Normatividad para la elaboración de registros clínicos.

El 7 de diciembre de 1998 se publicó en el Diario Oficial de la Federación la Norma Oficial Mexicana del expediente clínico NOM-168-SSA1-1998. Aunque posteriormente tuvo algunas modificaciones, las más recientes en el año 2012 dando así la norma NOM-004-SSA3-2012.

Uno de los principales objetivos en materia de salud fue fortalecer y complementar los servicios y sus componentes. La importancia del ordenamiento de esta norma está dirigida a sistematizar, homogeneizar y actualizar el manejo del expediente clínico que contiene los registros de los elementos técnicos esenciales para el estudio racional y la solución de los problemas de salud del usuario, involucrando acciones preventivas, curativas y rehabilitatorias que constituyen un herramienta obligatoria para los sectores público, social y privado del Sistema Nacional de Salud.



Capítulo 1. Introducción



Dicha norma es un instrumento para la regulación del expediente clínico y ayuda al desarrollo de una cultura de calidad que nos permite darle diferentes usos como lo son: médico, de investigación, enseñanza, jurídico, administrativo, estadístico y su evaluación.

Generalidades

Los prestadores de servicios médicos de carácter público, social y privado estarán obligados a integrar y conservar el expediente clínico en los términos previstos por la norma.

Todo expediente clínico, deberá tener los siguientes datos generales:

- Tipo, nombre y domicilio del establecimiento y, en su caso, nombre de la institución a la que pertenece.
- En su caso, la razón y denominación social del propietario o concesionario;
- Nombre, sexo, edad y domicilio del usuario y los demás que señalen las disposiciones sanitarias.

Del expediente en consulta

Deberá contar con: Historia Clínica.

Deberá elaborarla el médico y constará de: interrogatorio, exploración física, diagnósticos, tratamientos, en el orden siguiente:

- Interrogatorio. Deberá tener como mínimo: ficha de identificación, antecedentes heredo familiares, personales patológicos (incluido ex-fumador, ex-alcohólico y



Capítulo 1. Introducción



ex-adicto), y no patológicos, padecimiento actual (incluido tabaquismo, alcoholismo y otras adicciones) e interrogatorio por aparatos y sistemas.

- Exploración física. Deberá tener como mínimo: habitus exterior, signos vitales (pulso, temperatura, tensión arterial, frecuencia cardiaca y respiratoria), así como datos de cabeza, cuello, tórax, abdomen, miembros y genitales.
- Resultados previos y actuales de estudios de laboratorio, gabinete y otros.
- Terapéutica empleada y resultados obtenidos.
- Diagnósticos o problemas clínicos.

Nota de evolución. Deberá elaborarla el médico cada vez que proporciona atención al paciente ambulatorio, de acuerdo con el estado clínico del paciente. Describirá lo siguiente:

- Evolución y actualización del cuadro clínico (incluido tabaquismo, alcoholismo y otras adicciones).
- Signos vitales.
- Resultados de los estudios de los servicios auxiliares de diagnóstico y tratamiento.
- Diagnósticos y tratamiento e Indicaciones médicas, en el caso de medicamentos, señalando como mínimo: dosis, vía y periodicidad.



Capítulo 1. Introducción



- En el caso de control de embarazadas, niños sanos, diabéticos, hipertensos, entre otros, las notas deberán integrarse conforme a lo establecido en las Normas Oficiales Mexicanas respectivas.

Nota de Interconsulta. La solicitud deberá elaborarla el médico cuando se requiera y quedará asentada en el expediente clínico.

La elabora el médico consultado, y deberá contar con:

- Criterios diagnósticos.
- Plan de estudios.
- Sugerencias diagnósticas y tratamiento.

Nota de referencia/traslado. De requerirse, deberá elaborarla un médico del establecimiento y deberá anexarse copia del resumen con que se envía al paciente; constará de:

- Establecimiento que envía.
- Establecimiento receptor.
- Resumen clínico, que incluirá como mínimo:
 - I. Motivo de envío.
 - II. Impresión diagnóstica (incluido tabaquismo, alcoholismo y otras adicciones).
 - III. Terapéutica empleada, si la hubo.



De las Notas Médicas en Urgencias

Inicial. Deberá elaborarla el médico y deberá contener lo siguiente:

- Fecha y hora en que se otorga el servicio.
- Signos vitales.
- Motivo de la consulta.
- Resumen del interrogatorio, exploración física y estado mental en su caso.
- Diagnósticos o problemas clínicos.
- Resultados de estudios de los servicios auxiliares de diagnóstico y tratamiento;
- Tratamiento y Pronóstico.

Nota de evolución. Deberá elaborarla el médico cada vez que proporciona atención al paciente y las notas se llevarán a efecto conforme a lo previsto en el numeral 6.2 de la presente Norma.

En los casos en que el paciente requiera interconsulta por médico especialista deberá quedar por escrito, tanto la solicitud, la cual realizará el médico solicitante, como la nota de interconsulta que deberá realizar el médico especialista.

De referencia/traslado. Las notas se llevarán a efecto conforme a la nota de interconsulta.



De las notas médicas en Hospitalización.

De ingreso. Deberá elaborarla el médico que ingresa al paciente y deberá contener como mínimo los datos siguientes:

- Signos vitales.
- Resumen del interrogatorio, exploración física y estado mental, en su caso.
- Resultados de estudios en los servicios auxiliares de diagnóstico y tratamiento.
- Tratamiento y Pronóstico.

Historia clínica. Las notas se llevarán a efecto conforme a la nota de interconsulta.

Nota de evolución. Deberá elaborarla el médico que otorga la atención al paciente cuando menos una vez por día y las notas se llevarán a efecto conforme a la nota de interconsulta.

Nota de referencia/traslado. Las notas se llevarán a efecto conforme a la nota de interconsulta.

Nota Pre-operatoria. Deberá elaborarla el cirujano que va a intervenir al paciente incluyendo a los cirujanos internistas y deberá contener como mínimo:

- Fecha de la cirugía.
- Diagnóstico.
- Plan quirúrgico.



Capítulo 1. Introducción



- Tipo de intervención quirúrgica.
- Riesgo quirúrgico (incluido tabaquismo, alcoholismo y otras adicciones).
- Cuidados y plan terapéutico pre-operatorios y pronóstico.

Nota Pre-anestésica, vigilancia y registro anestésico. Se realizará bajo los lineamientos de la Normatividad Oficial Mexicana en materia de anestesiología y demás aplicables.

Nota Post-operatoria. Deberá elaborarla el cirujano que intervino al paciente, al término de la cirugía, constituye un resumen de la operación practicada, y deberá contener como mínimo:

- Diagnóstico pre-operatorio.
- Operación planeada.
- Operación realizada.
- Diagnóstico post-operatorio.
- Descripción de la técnica quirúrgica.
- Hallazgos transoperatorios.
- Reporte de gasas y compresas.
- Incidentes y accidentes.



Capítulo 1. Introducción



- Cuantificación de sangrado, si lo hubo.
- Estudios de servicios auxiliares de diagnóstico y tratamiento transoperatorios; y ayudantes, instrumentistas, anestesiólogo y circulante.
- Estado post-quirúrgico inmediato.
- Plan de manejo y tratamiento postoperatorio inmediato.
- Pronóstico.
- Envío de piezas o biopsias quirúrgicas para examen macroscópico e histopatológico.
- Otros hallazgos de importancia para el paciente relacionados con el quehacer médico; y nombre completo y firma del responsable de la cirugía.

Nota de egreso. Deberá elaborarla el médico y deberá contener como mínimo: Fecha de ingreso/egreso.

- Motivo del egreso.
- Diagnósticos finales.
- Resumen de la evolución y el estado actual.
- Manejo durante la estancia hospitalaria.
- Problemas clínicos pendientes.



Capítulo 1. Introducción



- Plan de manejo y tratamiento.
- Recomendaciones para vigilancia ambulatoria.
- Atención de factores de riesgo (incluido tabaquismo, alcoholismo y otras adicciones).
- Pronóstico y en caso de defunción, las causas de la muerte acorde al certificado de defunción y si se solicitó y obtuvo estudio de necropsia hospitalaria.

De los reportes del personal profesional, técnico y auxiliar

Hoja de enfermería. Deberá elaborarse por el personal de turno, según la frecuencia establecida por las normas del establecimiento y las órdenes del médico y deberá contener como mínimo:

- Habitus exterior.
- Gráfica de signos vitales;
- Ministración de medicamentos, fecha, hora, cantidad y vía.
- Procedimientos realizados y observaciones.

De los servicios auxiliares de diagnóstico y tratamiento. Deberá elaborarlo el personal que realizó el estudio y deberá contener como mínimo:

- Fecha y hora del estudio.



- Identificación del solicitante.
- Estudio solicitado.
- Problema clínico en estudio.
- Resultados del estudio.
- Incidentes si los hubo.
- Identificación del personal que realiza el estudio y nombre completo y firma del personal que informa.

1.5 Acotamiento del problema a resolver.

Como se ha mencionado en el capítulo 1.1 la necesidad de tener organizada en un sistema de cómputo toda la información referente al expediente clínico de enfermería del paciente es fundamental, ya que ayudará en la mejora de los cuidados del paciente y el trabajo diario del personal del hospital, tener toda la información centralizada y que pueda ser compartida por el personal médico y administrativo ayudará a reducir tiempos de atención al paciente, el sistema deseado para resolver estas necesidades sería un sistema web, lo que nos permitirá distribuir fácilmente el software por medio de una única actualización y por medio de cualquier dispositivo móvil o escritorio conectado a la red con un navegador web y que se pueda acceder a la información en cuestión de minutos, como se observa en la figura 1.5.1.



Capítulo 1. Introducción



Figura 1.5.1 Visualización del sistema en cualquier dispositivo.

Los sistemas hospitalarios son sistemas críticos que deben evitar la más mínima falla, ya que un registro erróneo puede alterar el diagnóstico o seguimiento del paciente, el sistema debe estar apegado a los más altos estándares informáticos, monitoreando todo el tiempo su funcionamiento y rendimiento, también como se ha mencionado en el capítulo 1.2 y 1.4 que los hospitales están apegados a técnicas y procedimientos regidas por normas oficiales que deben respetarse para la correcta implementación de un sistema de cómputo hospitalario, entender al usuario final (personal médico y administrativo) y rescatar toda la información y plasmarlo en requerimientos nos ayudará a dotar al sistema de las mejores características funcionales y no funcionales, y aplicando la metodología eXtreme Programming (**Programación Extrema**) se pretende acoplarse a un ciclo de vida del software dinámico, es decir, el sistema y los desarrolladores deben estar preparados para las nuevas adecuaciones que se irán presentando en el futuro como se observa en la figura 1.5.2.



Capítulo 1. Introducción



Figura 1.5.2 Ciclo de vida del software dinámico con la metodología XP

Como se ha mencionado en el capítulo 1.3 los hospitales están clasificados en diferentes áreas, el sistema de cómputo que se pretende implementar tendrá total movilidad para el llenado del expediente clínico de enfermería como se mencionó antes y es en el área de hospitalización concretamente en la central de enfermeras donde los enfermeros de campo darán seguimiento con el sistema todo lo relativo al paciente, una vez dado de alta los expedientes electrónicos cualquier personal del hospital administrativo y médico podrá acceder para su consulta.

El desarrollo ágil de aplicaciones es la tendencia en los sistemas informáticos ya que en poco tiempo y con una infraestructura robusta se tiene un sistema funcional, pudiendo dar soporte al sistema de una manera escalable y elegante, soluciones con bastantes años en el mercado como Ruby on Rails uno de los pioneros en el ramo, han hecho evolucionar nuevas propuestas basadas en lenguaje de programación Groovy como lo es Grails que nos ayudará a implementar y resolver problemas referentes con la configuración y codificación de los módulos que componen el sistema de cómputo, el producto final será un empaquetado de tipo war que podrá desplegarse en los servidores de aplicaciones Java líderes en el sector empresarial y hospitalario.

La arquitectura del sistema de cómputo que proponemos es un sistema vía web desarrollado con lenguaje de programación Groovy por medio del framework Grails, el desarrollo será en capas, es decir, se usa el ya conocido patrón de diseño



Capítulo 1. Introducción



MVC(Modelo, Vista, Controlador), y se usa mucho el principio de ingeniería de software **DRY**(No te repitas) que promueve la reducción de la duplicación de código por medio del llamado **CoC**(Convención sobre Configuración) es un paradigma de programación de software que busca decrementar el número de decisiones que un desarrollador necesita hacer, ganando así en simplicidad pero no perdiendo flexibilidad por ello, en la figura 1.5.3 se muestra la arquitectura del framework Grails y su relación con Groovy.

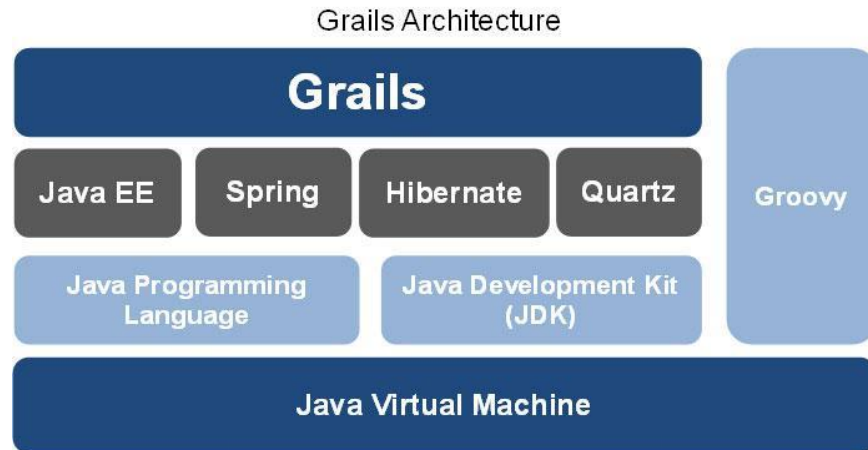


Figura 1.5.3 Groovy y el framework de desarrollo Grails

Toda la información arrojada por el sistema será guardado por el sistema gestor de base de datos de código abierto Postgresql ya que a nuestra consideración tiene un mayor número de objetos como son los tipos de datos, secuencias entre otros y su lenguaje SQL para crear procedimientos almacenados y triggers que es tan poderoso como lo es PL/SQL de Oracle, la generación de reportes se podrá extraer directamente de la base de datos por medio de herramientas de tipo **ETL**(Extracción, Transformación, Carga) para generar **Business Intelligence**(Inteligencia de Negocio), el hospital si así lo requiere podrá cambiarse a otro manejador de base de datos ya que el proceso sería muy sencillo debido a la facilidad que nos otorga usar Grails.

Para montar el sistema el hospital podrá hacer uso de su infraestructura de hardware y comunicaciones o hacer uso del cómputo en la nube para dar alojamiento y hosting al sistema y poder garantizar varios requisitos no funcionales como son:



Capítulo 1. Introducción



- Disponibilidad
- Escalabilidad
- Mantenibilidad y Portabilidad
- Operacionales y Económicos

La computación en la nube permite que los consumidores y las empresas gestionen archivos y utilicen aplicaciones sin necesidad de instalarlas en cualquier computadora con acceso a Internet. Esta tecnología ofrece un uso mucho más eficiente de recursos, como almacenamiento, memoria, procesamiento y ancho de banda, al proveer solamente los recursos necesarios en cada momento.

Actualmente en el mercado hay varias soluciones para contratar los servicios de cómputo en la nube, en este caso se recomienda Amazon Web Services por ser una de las soluciones líderes en su ramo ver la figura 1.5.4, los servicios propiamente a utilizar serán Amazon RDS para montar la base de datos, Amazon Elastic Beanstalk para desplegar el sistema en una distribución Amazon Linux con Apache Tomcat, este servicio nos ayudará a tener un control de versiones del sistema y dar escalabilidad cuando la demanda de usuarios así lo requiera y Amazon Route 53 para administrar el dominio elegido, generar los registros DNS para acceder al sistema vía internet.



Figura 1.5.4 Cómputo en la nube y el acceso por cualquier dispositivo

CAPÍTULO 2.

MARCO TEÓRICO



2.1 Características, ventajas y desventajas de las bases de datos orientadas a objetos.

Las bases de datos orientadas a objetos (BDOO) son aquellas en las que la información se representa mediante objetos. Se basa en el concepto de encapsular elementos de datos, sus características, atributos y el código que opera sobre ellos en elementos complejos llamados Objetos. El estado de un objeto se describe mediante atributos y su comportamiento se define mediante métodos. Cuando se integran las características de una base de datos con las de un lenguaje de programación orientado a objetos, obtenemos como resultado un sistema gestor de bases de datos orientada a objetos (SGBDOO).

Su origen se debe a la existencia de problemas para representar y modelar aspectos del mundo real. Las bases de datos orientadas a objetos surgen básicamente para tratar de mejorar las deficiencias de los modelos anteriores y para proporcionar eficiencia y sencillez a las aplicaciones.

Las clases que se utilizan en un determinado lenguaje de programación orientado a objetos serán las mismas clases que se usarán en una BDOO; de tal manera, que una transformación del modelo de objetos para ser utilizado por un SGBDOO no será necesaria.

Características

Proporcionan buen rendimiento en sistemas que necesitan manipulación en tipos de datos complejos. Sus costos de desarrollo son más bajos y poseen un mejor rendimiento cuando se usan objetos gracias a que almacenan objetos en disco y tienen una integración transparente con el programa escrito en un lenguaje de programación orientado a objetos, lo que reduce los costos de desarrollo y mantenimiento.



En 1989 se hizo el Manifiesto de los sistemas de base de datos orientados a objetos el cual propuso características obligatorias para un SGBDOO y un SGBD.

Características obligatorias de orientación a objetos:

- Deben soportarse objetos complejos.
- Deben soportarse mecanismos de identidad de los objetos.
- Debe soportarse la encapsulación.
- Deben soportarse los tipos o clases.
- Los tipos o clases deben ser capaces de heredar de sus ancestros.
- Debe soportarse el enlace dinámico.
- El lenguaje de manipulación de datos (DML) debe ser computacionalmente complejo.
- El conjunto de todos los tipos de datos debe ser ampliable.

Características obligatorias de SGBD:

- Debe proporcionarse persistencia a los datos.
- El SGBD debe ser capaz de gestionar bases de datos de gran tamaño.
- El SGBD debe soportar a usuarios concurrentes.
- El SGBD debe ser capaz de recuperarse de fallos hardware y software.
- El SGBD debe proporcionar una forma simple de consultar los datos.

Ventajas

- Se destaca su flexibilidad y soporte para el manejo de tipos de datos complejos.
- Manipular datos complejos en forma rápida y ágilmente.
- Reduce costos de desarrollo y mantenimiento.
- Mayor rendimiento.



- Mayor capacidad de modelado
- Ampliabilidad
- Lenguaje de consulta más expresivo.
- Adecuación a las aplicaciones avanzadas de base de datos.

Desventajas

- Carencia de un modelo de datos universal.
- Carencia de experiencia.
- Carencia de estándares.
- La optimización de consultas compromete la encapsulación.
- El modelo de objetos aún no tiene una teoría matemática coherente que le sirva de base.

2.2 Características, ventajas y desventajas del lenguaje de programación Groovy.

Los lenguajes interpretados o script están en claro auge. Son varias las razones, por una parte al incremento en la potencia y memoria de las máquinas que los ejecutan y la mejora de los motores de ejecución, lo que evita una de las mayores desventajas hasta ahora como era su menor rendimiento que un código compilado. Por otra parte, desde el punto de vista del programador son lenguajes con una sintaxis más relajada ya que son menos rígidos a la hora de establecer el tipado de datos o determinados tipos de estructuras de programación. La interoperabilidad también es uno de sus puntos fuertes, sirviendo como “pegamento” entre diferentes tecnologías. Lenguajes como Perl, Ruby o Python son una herramienta perfecta en estos días. Y ni que decir tiene del resurgimiento de JavaScript, tanto para HTML5 en los navegadores como en la parte del servidor con NodeJs.



Los lenguajes de programación interpretados requieren un motor de ejecución que va evaluando de forma secuencial sus instrucciones. Dentro de los muchos existentes llaman la atención aquellos que son capaces de correr en la **Java Virtual Machine** (Máquina Virtual de Java). Nos estamos refiriendo a Scala, Clojure o Groovy, ver figura 2.2.2

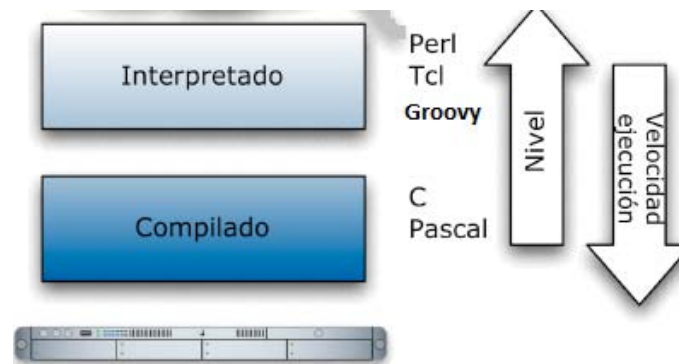


Figura 2.2.1 Lenguaje compilado e interpretado.

Groovy es un lenguaje ligero, dinámico, orientado a objetos y especialmente concebido para ser muy productivo, es decir, con pocas líneas de código se pueden hacer muchas más cosas. Y sobre todo, corre en la JVM y es compatible con Java, es decir, podemos utilizar código Java dentro de Groovy, así como las librerías de Java. Podemos hacer todo lo que puede hacer Java y además muchas otras cosas de un modo más fácil, no por algo dicen que Groovy es un Java con esteroides, Groovy se está estandarizando para hacer de él una alternativa oficial a Java.

Existe una framework de desarrollo ágil llamado Grails (como Ruby on Rails) para el desarrollo de aplicaciones web basado en Groovy, que está haciendo de Groovy un lenguaje de programación muy popular.

Características



Capítulo 2. Marco Teórico



- El lenguaje Groovy es un superconjunto del lenguaje Java. En general se puede renombrar un archivo .java en .groovy y va a funcionar, Java al igual que Groovy genera bytecodes (archivos .class) para que la JVM pueda interpretarlos, como se observa en la figura 2.2.2.

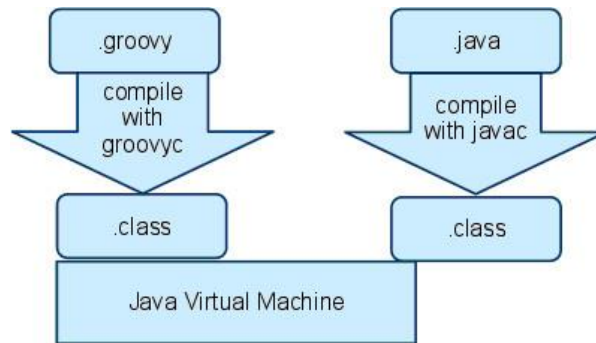


Figura 2.2.2 Groovy y Java generando bytecodes.

- Groovy tiene algunas características que no existen en Java. Esto hace que Groovy sea un paso natural para los desarrolladores Java, ya que aprender Groovy implica aprender algunos detalles extra más, la curva de aprendizaje es bastante alta.
- Las aserciones son usadas para validar una expresión que se considera como verdadero, en Groovy aceptan cualquier tipo de datos y en Java solo aceptan booleanos, ver figura 2.2.3.

```
def list = [1, 2, 'x'] // Lista de 3 elementos
assert list.size() == 3
```

Figura 2.2.3. Uso de aserciones en Groovy

- Generar un simple documento XML en Java es difícil y tardado, Groovy simplifica su creación, proveyendo clases constructoras, y haciendo que el código hable por sí mismo.
- En Groovy sobresale una característica llamada closure que es un bloque reusable de código denotado con {}, el cual puede ser asignado a una variable o



Capítulo 2. Marco Teórico



pasado como parámetro a un método, el código que está dentro `{}` es ejecutado cuando el closure es invocado, ver figura 2.2.4.

```
def name = "Luis"  
def printClosure = { println "Hola, ${name}" }  
printClosure()
```

Figura 2.2.4. Uso de closures en Groovy

- Hoy en día cualquier aplicación hace uso de **JSON**(Notación de Objeto Javascript) que es un formato ligero para el intercambio de datos es una alternativa al XML y Groovy tiene soporte nativo.
- Java no tiene soporte en su lenguaje para la declaración de colecciones pero si lo tiene para el manejo de arrays, Groovy realiza el manejo de estructuras de datos como son las listas y los mapas mejorando la declaración de sus sintaxis y agregando nuevos métodos, ver figura 2.2.5.

```
authors = [ 'Vishal', 'Chris', 'Joseph', 'Jim' ]  
println authors  
println authors[0]
```

Figura 2.2.5. Uso de lista en Groovy

- Todo es un objeto en Groovy, a diferencia de Java no hay tipos primitivos, como resultado el autoboxing no es necesario porque todo es un objeto.
- String interpolation es la habilidad para sustituir una expresión o variable con un string. Java no soporta esta característica, el desarrollador debe concatenar manualmente los valores, ver figura 2.2.6.

```
def lastName = "Layka"  
def fullname = "Vishal ${lastName}" // string interpolation  
println fullname
```

Figura 2.2.6. Uso de String Interpolation en Groovy

- En Groovy las clases sin modificador de acceso son considerados públicos, atributos sin modificador de acceso serán privados y los métodos sin modificador son públicos, esto contrasta con Java donde los métodos y atributos sin modificador son considerados con nivel package.



Capítulo 2. Marco Teórico



- Las excepciones en Groovy no necesitan ser declaradas o atrapadas, ya que Groovy envuelve todas las excepciones con el tipo `RuntimeException`, que si ocurre la excepción la dispara en tiempo de ejecución.
- Groovy introduce los `GroovyBeans`, el cual da soporte para la declaración de propiedades, las propiedades no necesitan definir explícitamente métodos `setters` y `getters`, ver figura 2.2.7.

```
class Employee { String name }  
Employee emp = new Employee()  
emp.setName("John Doe" )  
println emp.name
```

Figura 2.2.7. Uso de propiedades en Groovy

- Para ejecutar código Groovy se puede mediante la consola de Groovy. Esta consola funciona en modo gráfico y permite opciones más potentes que el shell, como guardar y cargar archivos, opciones de edición de texto, etc. Para lanzar la consola de Groovy se puede ejecutar desde la línea de comandos: `groovyconsole`.
- La metaprogramación en Groovy, al igual que en Java, se basa en la capacidad del lenguaje llamada reflexión. Mediante reflexión podemos (entre otras cosas) conocer los miembros de una clase, ver figura 2.2.8.

```
String.interfaces.each { println it }  
String.constructors.each { println it }  
String.methods.each { println it }
```

Figura 2.2.8. Uso de reflexión en Groovy

- Groovy permite la adición de nueva funcionalidad a una clase de la que, por ejemplo, no disponemos del código fuente. Esto es posible a través de una categoría, la cual implementa los métodos que deseamos añadir a dicha clase.

Ventajas



Capítulo 2. Marco Teórico



- Groovy es fácil de aprender, es decir, si ya sabes Java puede ser una buena herramienta para todo tipo de desarrollos rápidos o como complemento al código de Java más complejo que hemos realizado.
- La sintaxis es compacta evitar la verbosidad en el código, hacer mucho más cosas con menos código.
- Es un lenguaje ágil y dinámico, es decir, no se cierra con una sintaxis exacta para llevar a cabo una tarea, es decir, podemos hacer la misma tarea de muchas y elegantes formas incrementando increíblemente la productividad.
- Está implementado sobre la plataforma Java, desde él se puede acceder a todas las librerías existentes en Java por medio de sentencias import.
- Se usa como complemento, es decir, se puede utilizar los procesos críticos con Java y lo que no importe tanto en rendimiento en Groovy.
- Groovy es un buen sustituto para Java en JVM y pronto probablemente lo será en Android también. Tiene una gran cantidad de nuevas APIs que mejoran la legibilidad y la productividad y la pérdida de rendimiento, cuando la hay, es moderada.
- Groovy ya lleva utilizándose durante años para el desarrollo de software, está muy evolucionado y cuenta con una comunidad de usuarios muy amplia, empresas de gran renombre hacen uso de Groovy, ver figura 2.2.9.



Figura 2.2.9. Empresas que hacen uso de Groovy

- Es muy útil para probar nuestro código Java. El desarrollo **TDD** (Desarrollo Orientado a Pruebas) se está demostrando muy eficaz en la creación de todo



tipo de aplicaciones, nos puede servir para probar nuestro código o desarrollar pruebas de forma muy rápida. O también integrarlos con otros sistemas.

- Es una buena herramienta de scripting si estás habituado a Java. Y manejar un lenguaje de script hoy en día es fundamental para poder interconectar multitud de sistemas o simplemente ser productivos en aquellas tareas que son secundarias.
- Cuenta con un IDE oficial Groovy/Grails Tool Suite basado en eclipse para desarrollar con facilidad.

Desventajas

- Es un poco más lento Groovy que Java, pero en cosas muy puntuales, es posible hacer optimizaciones muy simples en nuestro código para incrementar la velocidad.
- Groovy es un lenguaje dinámico, por lo que hay muchos errores que deberían aparecer en tiempo de compilación y que no salen. Estos errores aparecen después, en tiempo de ejecución, a veces en producción.
- Al distribuir un aplicativo hecho con Groovy siempre será un archivo más grande en cuanto ocupación en disco duro que al hacerlo en Java.

2.3 Características, ventajas y desventajas del gestor de base de datos PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, utiliza un modelo cliente-servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema, de este modo un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Es compatible con el estándar SQL y ofrece nuevas características. En la figura 2.3.1 se muestra el diagrama básico de un sistema PostgreSQL.

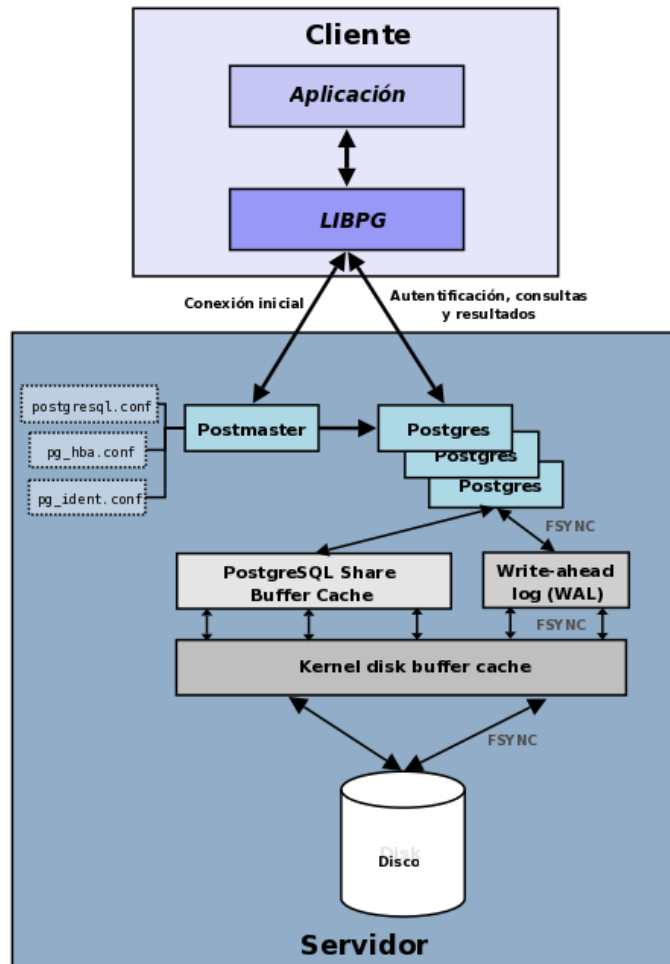


Figura 2.3.1 Diagrama básico de un sistema PostgreSQL

Aplicación cliente: Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.

Postmaster: Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.



Ficheros de configuración: Los 3 ficheros principales de configuración utilizados por PostgreSQL.

Procesos hijos postgres: Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.

PostgreSQL share buffer cache: Memoria compartida usada por PostgreSQL para almacenar datos en caché.

Write-Ahead Log (WAL): Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO)

Kernel disk buffer cache: Caché de disco del sistema operativo.

Disco: Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione

Características

- Consultas complejas
- Claves externas
- Disparadores (triggers)
- Vistas actualizables
- Integridad transaccional
- Es una base de datos con características **ACID** (Atomicidad, Consistencia, Aislamiento y Durabilidad)
- Presenta Integridad referencial
- Integra el concepto Tablespaces
- Replicación sincrónica y asincrónica



Capítulo 2. Marco Teórico



- Copias de seguridad en línea
- Juegos de caracteres internacionales
- Regionalización por columna
- Alta concurrencia **MVCC** (acceso concurrente multiversión)
- Múltiples métodos de autenticación
- Acceso encriptado vía SSL (capa de conexión segura)
- Completa documentación
- Licencia **BSD** (distribución de software Berkeley)
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP, etc.), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, etc.)
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.



Ventajas

- **Alta concurrencia:** Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de la última modificación de la tabla. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- **Multiplataforma:** Está disponible en casi cualquier plataforma Linux, UNIX y Windows.
- **Estabilidad y confiabilidad.** Al mismo nivel que gestores de bases de datos comerciales.
- **Diseñado para ambientes de alto volumen:** Utilizando una estrategia de almacenamiento de filas llamada MVCC. Que permite mejores respuestas en grandes volúmenes.
- **Herramientas gráficas de diseño y administración de bases de datos:** Como pgAdmin y pgAcces.
- **Documentación:** Información completa, documentada, pública y libre.
- **Económico:** no requiere costos de para su utilización, lo que representa un ahorro considerado en costos de instalación.
- **No es necesario tener una licencia.**



- **Extensible:** el código fuente está disponible para todos sin costo, si se necesita extender o personalizar PostgreSQL, se puede hacer de manera sencilla y sin costos adicionales.
- **Gran capacidad de almacenamiento.**

Límite	Valor
Máximo tamaño base de dato	Ilimitado
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

Desventajas

- En comparación con otros gestores de bases de datos es más lento en inserciones y actualizaciones.
- Consume más recursos que otros gestores de bases de datos, por lo que requiere mayores características de hardware para ejecutarlo.
- El software libre no tiene garantía
- Soporte en línea: aunque existen foros el soporte en línea no está disponible de forma obligatoria.
- La sintaxis de algunos de los comandos o sentencias son complejas y poco intuitivas



2.4 Características, ventajas y desventajas de la arquitectura cliente-servidor.

Características

A continuación se definirán los conceptos de Arquitectura, Cliente y Servidor en términos informáticos y del sistema presentado en este trabajo de tesis:

Arquitectura: Es el diseño operacional de la estructura de un sistema informático. Es decir, es un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para que las partes del sistema interactúen entre sí⁸.

Cliente: Es el que inicia un requerimiento de servicio, este requerimiento puede convertirse en múltiples requerimientos a través de una intranet o el internet de una red⁹. En el sistema de expediente clínico, son los procesos que permiten al usuario formular los requerimientos y pasarlos a los servidores, también conocido como **front-end** (interfaz). Estos procesos manejan todas las funcionalidades relacionadas con el despliegue de datos, la interacción con el usuario mediante **GUIs (Graphic User Interfaces)** (interfaces gráficas de usuario) y además pueden acceder a los servicios de toda la red.

Servidor: Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes **LANs (Local Área Networks)** (Redes de Área Local) o **WANs (Wide Área Networks)** (Redes de Área Amplia), para proveer de múltiples servicios a los clientes tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc. ¹⁰ En nuestro sistema, son los procesos encargados de atender múltiples clientes, también

⁸ Arquitectura Informática, ECURED, Sitio web:
http://www.ecured.cu/index.php/Arquitectura_Inform%C3%A1tica
Consulta 19-07-2015

⁹ Ídem.

¹⁰ Muñoz R., Carlos. Cliente Servidor, Universidad de Colima, Sitio web:
http://docente.ucol.mx/rcarlos/public_html/cliente-servidor.htm
Consulta 19-07-2015

son conocido como **back-end** (Motor). Estos procesos manejan todas las funciones relacionadas con las reglas de negocio, la lógica de procesamiento, el formateo y el almacenamiento de datos.

La arquitectura Cliente-Servidor se utiliza para construir sistemas de información. Su principal función es distribuir y repartir el tratamiento de la información y los datos por todo el sistema informático. Está arquitectura distribuida permite a los usuarios finales obtener acceso a la información en forma transparente. El cliente envía una petición, o envía un mensaje solicitando un determinado servicio a un servidor, y este provee el servicio al cliente enviando uno o varios mensajes con la respuesta (Figura 2.4.1).



Figura 2.4.1 Arquitectura básica Cliente-Servidor

La capacidad de proceso está repartida entre los clientes y los servidores. Usualmente el trabajo pesado se hace en el proceso servidor y el o los procesos cliente sólo se ocupan con la interacción con el usuario (aunque esto puede variar). En otras palabras la arquitectura Cliente/Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento.



Capítulo 2. Marco Teórico



En pocas palabras, distribuye los procesos y los datos en forma eficiente reduciendo significativamente el tráfico de datos de la red.

A continuación se enlistan las principales características de los sistemas que implementan la arquitectura cliente servidor¹¹:

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos. Por ejemplo, Las interfaces gráficas de formularios y **check lists** (Listas de verificación) hacen numerosas peticiones al servidor de base de datos de consulta, inserción y actualizaciones (Figura 2.4.2).
- Como los clientes son los que hacen las peticiones, los servidores tienen un comportamiento pasivo esperando las peticiones de dichos clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. Dichos mensajes se traducen en peticiones a la base de datos, al servidor de aplicaciones o al servidor web.
- A pesar de que puedan vivir en la misma máquina. Las plataformas de software y hardware entre clientes y servidores son independientes.

¹¹ Neilla Rogelio. Soluciones Transaccionales. Sitio web:
http://www.oocities.org/ar/r_niella/Document/t_marco.htm
consulta 20-07-2015

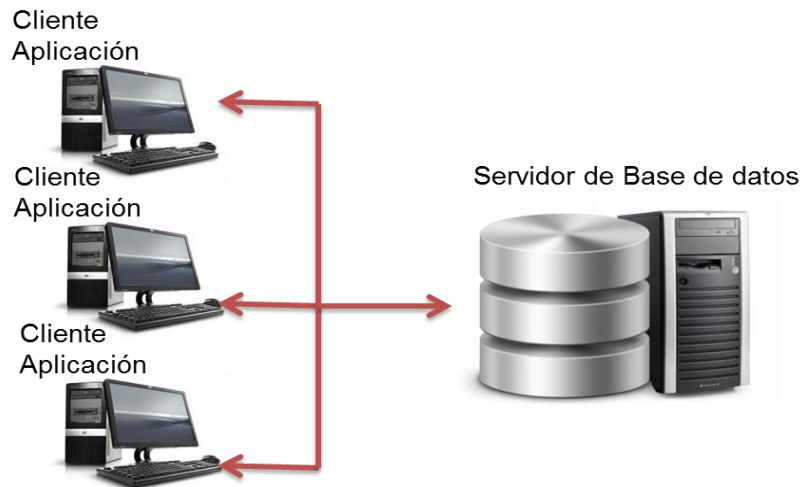


Figura 2.4.2 Arquitectura Cliente-Servidor en una base de datos

Ventajas

Las ventajas de la arquitectura Cliente-Servidor, están presentes a nivel aplicativo, funcional, comunicación, etc. A continuación se describen las más importantes:

- La simplicidad de esta arquitectura, permite el intercambio de peticiones entre el cliente y servidor para la interacción entre usuarios y poder comunicarse, tales como el chat, el correo electrónico y la navegación web.
- La arquitectura Cliente-Servidor centraliza los recursos en una red completa. El servidor administra los accesos, los recursos y su integridad. Además los clientes defectuosos no autorizados no pueden dañar el sistema.
- Al disponer de un mecanismo central de autenticación, las posibilidades de acceso indebido se reducen considerablemente.
- La red se puede alterar agregando más clientes o más servidores sin afectar el funcionamiento de los demás. Además es fácil darle mantenimiento ya que se pueden quitar componentes fácilmente.
- Un gran número de tecnologías implementan esta arquitectura, por lo que aumenta la calidad, la facilidad de empleo e interfaces más entendibles.



Desventajas

- La congestión de tráfico. Cuando una gran cantidad de clientes envían peticiones simultáneamente al mismo servidor, sobrecargan al servidor, por lo que se perjudica a los clientes y falla el sistema.
- Costos elevados. Se requiere de mayor configuración para la seguridad y estabilidad de datos. Además se requiere de personal de red experimentado para darle mantenimiento al servidor dependiendo del tamaño y complejidad del sistema.
- El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web, no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.
- Dependencia del servidor. Toda la red está construida alrededor del servidor y si éste deja de funcionar o lo hace con un rendimiento inadecuado, afectará a toda la infraestructura. Aunque se puede contrarrestar este problema con servidores redundantes o alta disponibilidad, que son inmunes a fallos.

Arquitectura de 1,2,3,...,n capas

La arquitectura de 2 capas, es la arquitectura tradicional de Cliente-servidor. Las capas que esta arquitectura presenta son las siguientes:

- Capa de aplicación. Este nivel es en el que se encuentra toda la interfaz del sistema, la lógica del negocio de un sistema y es la que el usuario puede disponer para realizar su actividad con el sistema.
- Capa de base de datos. También llamado el repositorio de datos, es la capa en donde se almacena toda la información ingresada en el sistema y que se deposita en forma permanente.



Posteriormente se añadió más complejidad a la arquitectura de dos capas y es así como surgieron los conceptos de "cliente grueso-servidor delgado" y "cliente delgado-servidor grueso".

Cliente grueso-servidor delgado

La mayor parte del procesamiento se realiza en el cliente grueso inteligente. El concepto de grueso surge a partir de que se tiene una gran cantidad de clientes conectados, entonces la interfaz gráfica de usuario emerge como un entorno predominante con él, emerge un nuevo enfoque en el planteamiento inicial de la arquitectura de dos capas. El servidor de archivos en red de propósito general se reemplaza por un servidor de bases de datos especializado. Las características más importantes de los clientes gruesos son:

- Cuanto más complicada la aplicación, más "grueso" pasa a ser el cliente y más potente debe ser el hardware que debe soportarlo.
- El coste de adecuar la tecnología del cliente pasa a ser prohibitivo y puede frustrar el funcionamiento de las aplicaciones.
- Además, la carga de la red utilizando este tipo de clientes es muy grande, de modo que el ancho efectivo de la red (y por lo tanto del número de usuarios que pueden utilizarla) se reduce¹².

Cliente delgado-servidor grueso

En este caso el cliente invoca procedimientos almacenados en el servidor de base de datos. El inconveniente de este modelo, es que el uso de procedimientos almacenados

¹² Unidad Nacional de Educación Abierta y a Distancia. Colombia. Lección 2: Tipos de arquitecturas cliente-servidor. Sitio Web: http://datateca.unad.edu.co/contenidos/208017/ContLin/leccin_2_tipos_de_arquitecturas_clienteservidor.html Consultada 21-07-2015



hace depender el desarrollo excesivamente del software del desarrollador y En grandes bases de datos distribuidas esto puede conducir a una administración tediosa.

Arquitectura de tres capas

Una generación más moderna de la arquitectura Cliente-Servidor añade una capa intermedia. En la arquitectura de tres capas (en general, en la arquitectura multicapa) se presentan las siguientes capas (Figura 2.4.3):

- Capa de presentación: Presenta el sistema al usuario, comunica la información y captura la información del usuario en un mínimo proceso. Esta capa se comunica únicamente con la capa de negocio.
- Capa de negocio: Es donde residen los programas que se ejecutan, se reciben peticiones del usuario y se envían las respuestas tras el proceso, es aquí donde se establecen todas las reglas que deben cumplirse, se comunica con la capa de presentación, para recibir solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos.
- Capa de datos: Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



Figura 2.4.3. Arquitectura de tres capas.



Arquitectura de N capas

En una aplicación distribuida en n-capas los diferentes procesos están distribuidos en diferentes capas no sólo lógicas, sino también físicas. Los procesos se ejecutan en diferentes equipos, que pueden incluso residir en plataformas o sistemas operativos completamente distintos. Cada equipo posee una configuración distinta y está optimizado para realizar el papel que le ha sido asignado dentro de la estructura de la aplicación, de modo que tanto los recursos como la eficiencia global del sistema se optimicen (Figura 2.4.4)¹³.

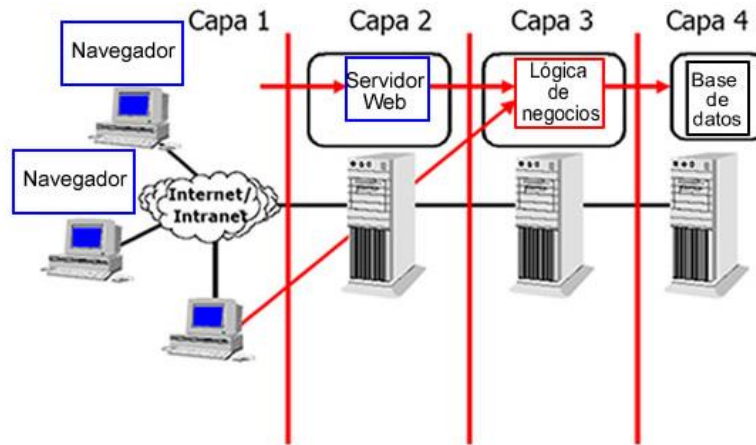


Figura 2.4.4 Arquitectura de n capas.

2.5 Características, ventajas y desventajas de la metodología de desarrollo iterativo creciente y extreme programming-xp.

Metodología de desarrollo iterativo creciente

El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias

¹³ <https://laurmolina7821.wordpress.com/1-1-3-aplicaciones-de-2-3-y-n-capas/>
Consultada 21-07-2015

lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software. Figura 2.5.1

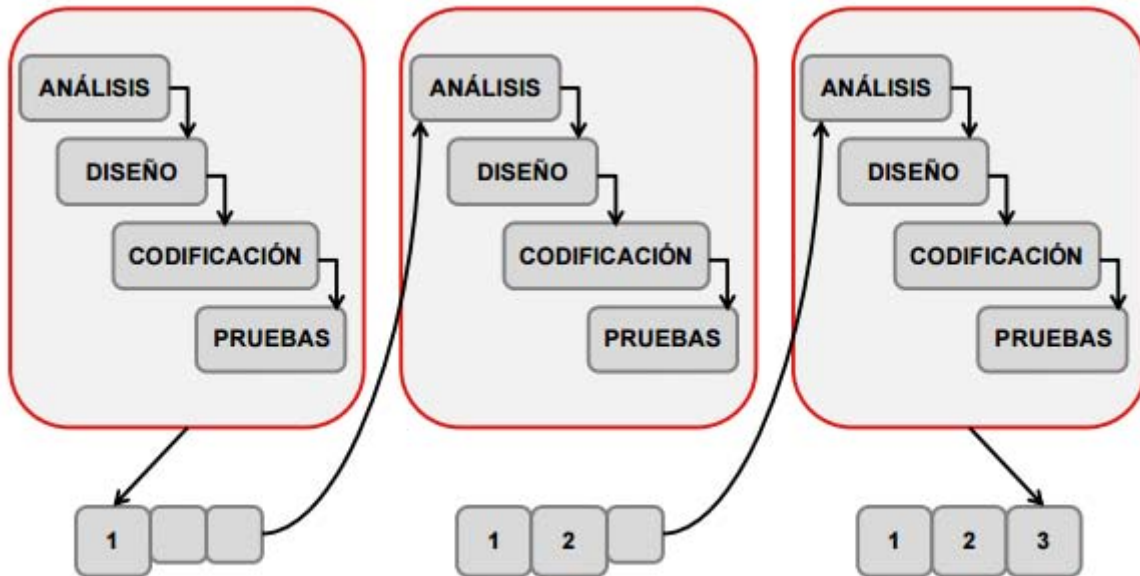


Figura 2.5.1. Desarrollo iterativo creciente

Cuando se utiliza un modelo incremental, el primer incremento es a menudo un producto esencial, sólo con los requisitos básicos. Este modelo se centra en la entrega de un producto operativo con cada incremento. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación.

Características

La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema.



Ventajas

Entre las ventajas que puede proporcionar un modelo de este tipo encontramos las siguientes:

- Mediante este modelo se genera software operativo de forma rápida y en etapas tempranas del ciclo de vida del software.
- Es un modelo más flexible, por lo que se reduce el coste en el cambio de alcance y requisitos.
- Es más fácil probar y depurar en una iteración más pequeña.
- Es más fácil gestionar riesgos.
- Cada iteración es un hito gestionado fácilmente

Desventajas

Para el uso de este modelo se requiere una experiencia importante para definir los incrementos y distribuir en ellos las tareas de forma proporcionada.

Entre los inconvenientes que aparecen en el uso de este modelo podemos destacar los siguientes:

- Cada fase de una iteración es rígida y no se superponen con otras.
- Pueden surgir problemas referidos a la arquitectura del sistema porque no todos los requisitos se han reunido, ya que se supone que todos ellos se han definido al inicio.

Extreme Programming XP

La programación extrema (XP) es un enfoque de la ingeniería del software formulado por Kent Beck. Es el más destacado de los procesos ágiles de desarrollo de software.



Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del software.

Características

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Ventajas

- Da lugar a una programación sumamente organizada.
- Ocasiona eficiencias en el proceso de planificación y pruebas.
- Cuenta con una tasa de errores muy pequeña.
- Propicia la satisfacción del programador.
- Fomenta la comunicación entre los clientes y los desarrolladores.
- Facilita los cambios.
- Permite ahorrar mucho tiempo y dinero.
- Puede ser aplicada a cualquier lenguaje de programación.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto.



Capítulo 2. Marco Teórico



- La XP es mejor utilizada en la implementación de nuevas tecnologías.

Desventajas

- Es recomendable emplearla solo en proyectos a corto plazo.
- En caso de fallar, las comisiones son muy altas.
- Requiere de un rígido ajuste a los principios de XP.
- Puede no siempre ser más fácil que el desarrollo tradicional.

CAPÍTULO 3.

ANÁLISIS Y

PLANTEAMIENTO DEL

PROBLEMA

3.1 Análisis del problema.

El objetivo de este capítulo es describir de manera gráfica los procedimientos que se deben seguir en las áreas médicas de un hospital, descritas en el capítulo 1.3 antes y después de la aplicación del sistema de registro clínico.

La forma actual que se realizan los procedimientos es la siguiente:

Previamente, para hacer uso de los servicios clínicos que ofrece un hospital o clínica, el paciente debe registrarse a sí mismo y a sus derechohabientes en el área administrativa o de archivo en la unidad médica. Debe llenar algunos formularios a mano de los involucrados y presentar una serie de documentos oficiales para acreditar su autenticidad. (Figura 3.1.1). En seguida, el personal administrativo, captura los datos a mano y saca copias de los documentos del paciente. Tanto la captura como las copias, son archivadas para su uso posterior.

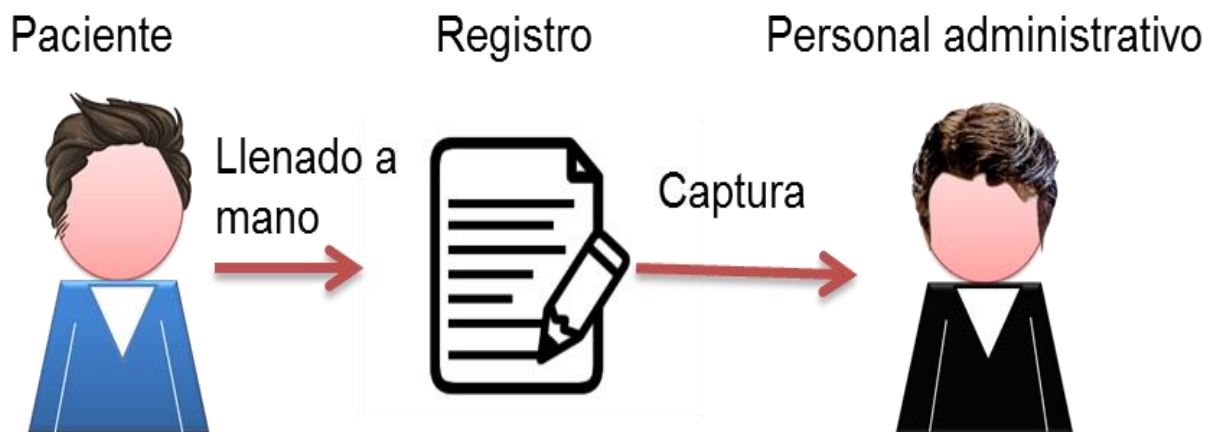


Figura 3.1.1.Registro del paciente en la unidad médica

Para agendar una cita, el paciente puede presentarse de forma personal o llamar por teléfono, después el recepcionista registra su cita en una agenda de forma manual.

Para que el paciente pueda pasar a una consulta, este deberá formarse y esperar su turno. El recepcionista deberá revisar su identificación y buscar manualmente el registro

que agendó el paciente. Deberá buscar su expediente clínico, proporcionarlo al médico y esperar a que lo llamen. Cuando el paciente ingresa al consultorio. El médico toma notas y las apunta en el formato oficial de registro clínico que se volverá a archivar. El médico a través de su experiencia, realiza el diagnóstico y le da la receta médica al paciente. Al final el paciente proporciona la receta médica al personal de farmacia y si hay medicamento se lo entrega al paciente (Figura 3.1.2).

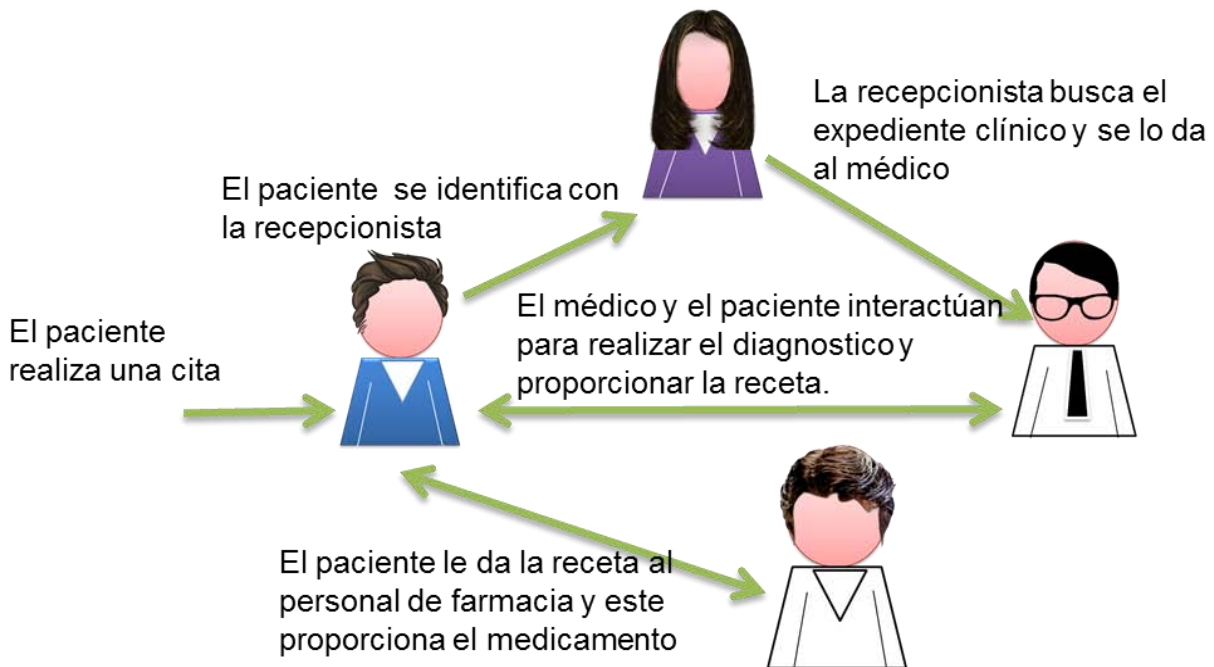


Figura 3.1.2. Procedimiento para una consulta.

El proceso para el área de consulta externa y urgencias, descrita en el capítulo 1.3 es el siguiente:

El paciente ingresa al área de control para solicitar atención, se deberá registrar en caso que no sea así y se creará su registro clínico. Al ingresar al consultorio de valoración, el médico realizará la valoración para determinar si el paciente debe ser hospitalizado para una intervención quirúrgica o únicamente se recetará medicamento o curaciones.

En caso que el paciente sea valorado por el médico para ser hospitalizado, se le deberá programar asistencia médica y enfermería permanente durante su estancia en el hospital. Para su monitoreo, cuidados y seguimiento, al personal se le aginará el expediente clínico de enfermería y se le llevará a una cama asignada en los cuartos de hospitalización, donde se depositará el expediente clínico de enfermería. Este documento debe ser llenado y actualizado de forma manual por el personal en turno para que pueda ser consultado por los médicos y el personal de otros turnos (Figura 3.1.3). De ser necesario y de acuerdo a los diagnósticos se programa una intervención quirúrgica con el grupo de médicos especializados, por cada movimiento crítico se debe registrar en el expediente clínico, para evitar problemas legales en un futuro. De acuerdo a las especificaciones puestas en el mencionado expediente clínico por parte del personal médico, se programan rehabilitaciones, control de medicamento, cuidados especiales etc., según sea el caso y se le da seguimiento, y monitoreo continuo hasta su recuperación o alta del hospital.

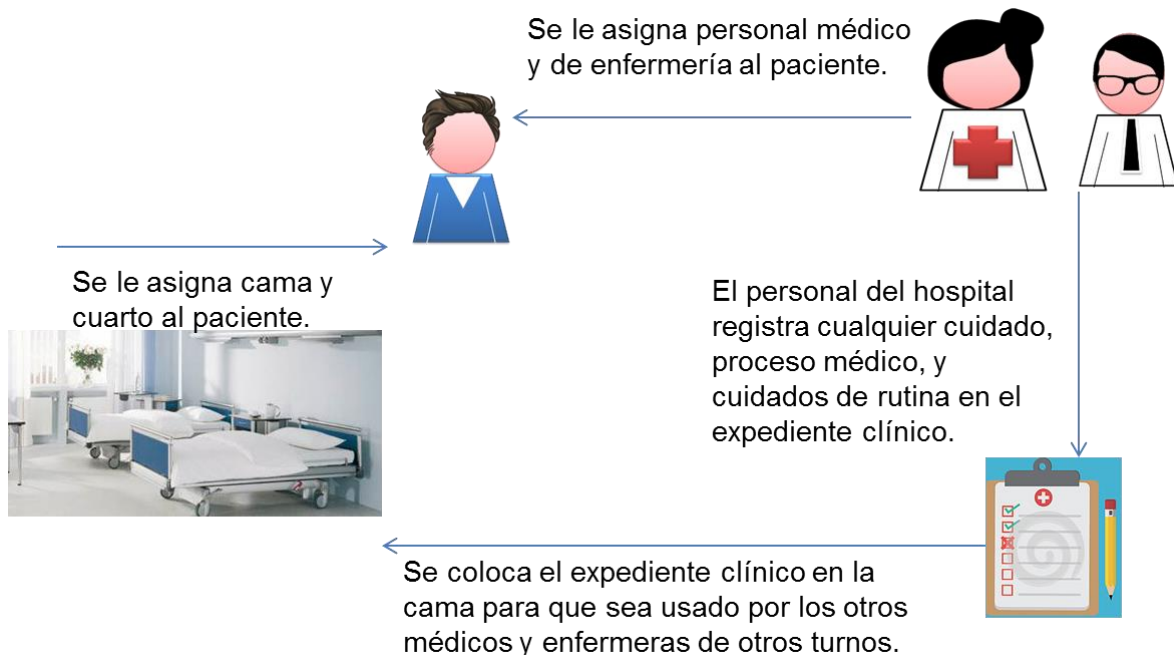


Figura 3.1.3 Hospitalización del paciente y asignación de registro clínico



Capítulo 3. Análisis y planteamiento del problema



En el proceso descrito anteriormente, se puede observar que todos los registros del paciente se realizan de forma manual, así como las tareas desempeñadas por los médicos. Cabe señalar que el nivel de error humano es alto, ya que el expediente clínico de enfermería se mantiene de forma impresa desde su creación hasta el archivamiento, lo que ocasiona que a veces se traslapen, se confundan en las tareas asignadas e incluso dar un mal diagnóstico. Aunado a esto, se desperdicia mucho espacio al archivar los expedientes de todos los pacientes en archiveros o almacenes muy grandes, ocasionando pérdidas de tiempo en búsqueda y almacenamiento.

Al implementar el sistema de registro clínico de enfermería, se mejora la mayoría de los procesos involucrados del área médica y administrativa que a continuación se describen:

Para realizar una cita, el recepcionista sólo debe capturar en la computadora, el nombre o número de identificación del paciente, la fecha y la hora que se realizará la cita. Este registro se guardará en la base de datos y mantendrá apartado el horario en esa fecha.

Cuando llegue el día de consulta del paciente, sólo deberá presentar su número de identificación al recepcionista y aparecerán sus datos de forma inmediata. Al entrar al consultorio, el médico ya tendrá su expediente clínico obtenido de la base de datos. Al terminar la revisión, el médico hará el diagnóstico a partir de un menú para disminuir el error humano, este se guardará y actualizará en la base de datos. Por último la receta se imprimirá de forma digital para que el paciente recoja los medicamentos en farmacia (Figura 3.1.4).

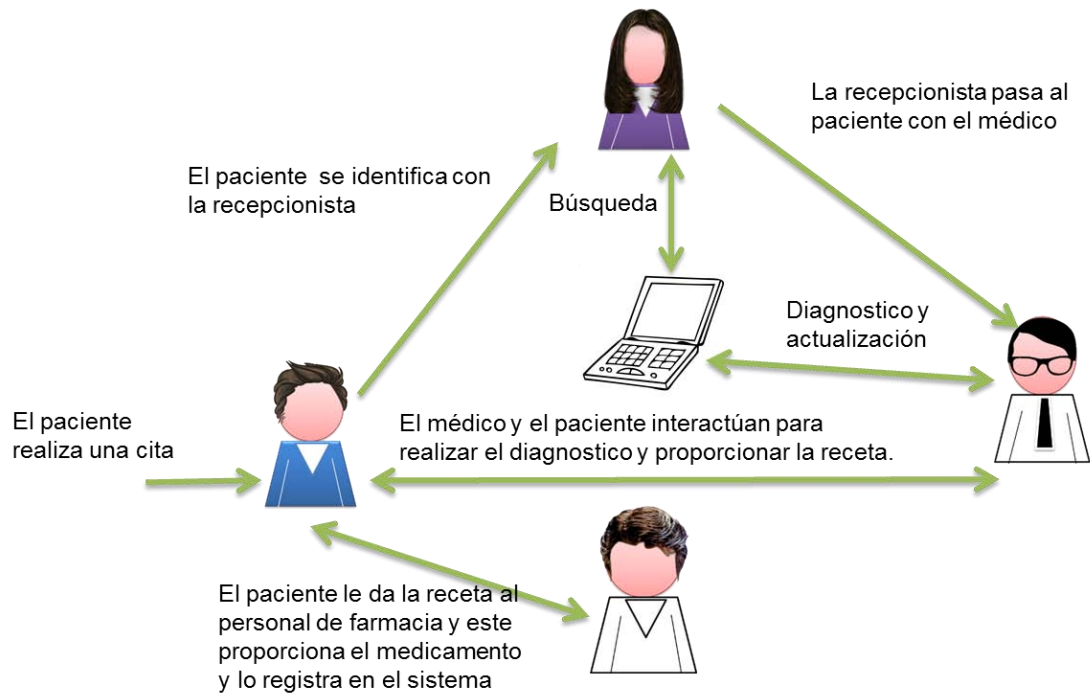


Figura 3.1.4. Procedimiento para una consulta con el sistema de expediente clínico.

En el área de urgencias y consulta externa se tendrá casi el mismo procedimiento que la forma anterior, sólo cambiará la forma de manejar la información. El registro clínico se tendrá en forma digital y será llenado desde una computadora. En caso de que el médico dictamine la hospitalización al paciente, se mantendrá el proceso normalizado de atención al paciente. Al asignarle cuarto y cama al paciente, también se le asignará un expediente clínico de enfermería. Este registro será único y se disminuirán considerablemente errores que se cometían al traspapelar los expedientes. Cualquier movimiento, cuidado, medicamento proporcionado y estadísticos que define el expediente clínico normalizado, será registrado en el expediente clínico digital y en caso de requerirse se imprimirá para trámites adicionales. A parte de disminuir errores en la atención médica, también se ahorra tiempo, papelería, generación de reportes de forma automática y el control de la medicación de los pacientes (Figura 3.1.5).

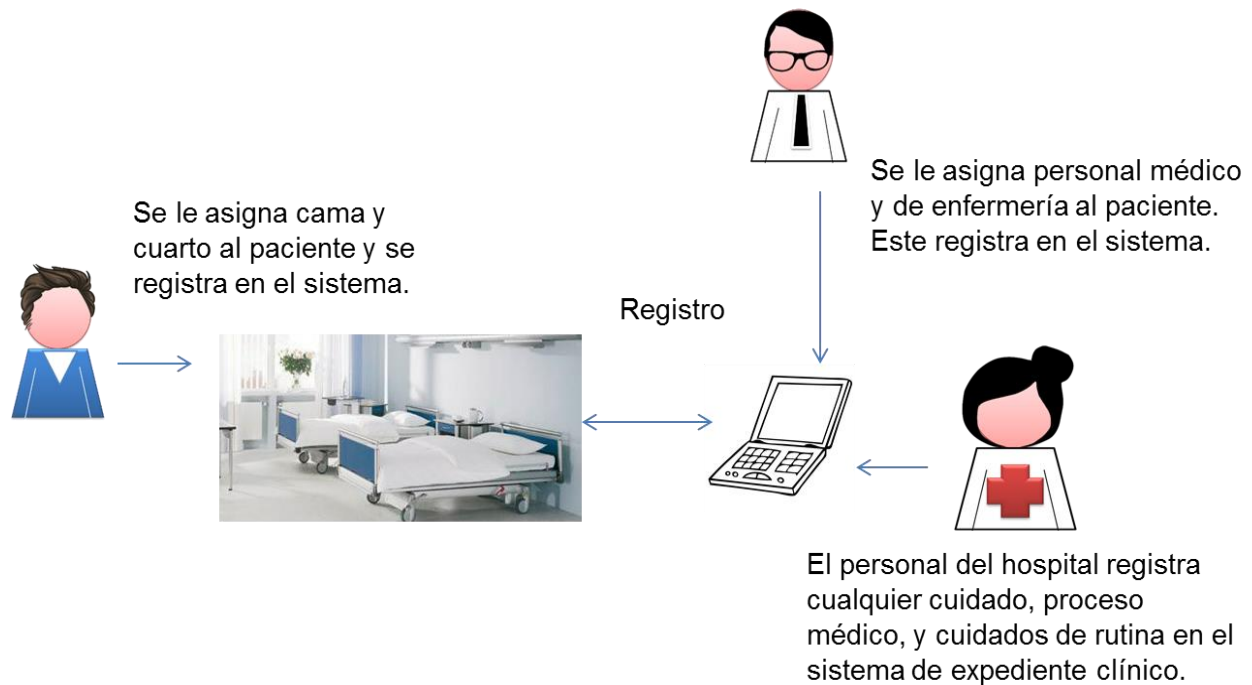


Figura 3.1.5. Hospitalización del paciente usando el sistema de expediente clínico.

3.2 Requerimientos generales del sistema.

Como se analizó el problema en el inciso 3.1, se pueden empezar a definir los requerimientos del expediente clínico de enfermería, entendiéndose por requerimiento una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema de software para que tenga valor y utilidad para el usuario, ver figura 3.2.1. En otras palabras, los requerimientos muestran qué elementos y funciones son necesarias para un proyecto, de acuerdo a lo anterior podemos delimitar este análisis de requerimientos en dos grupos generales y particulares.



Capítulo 3. Análisis y planteamiento del problema

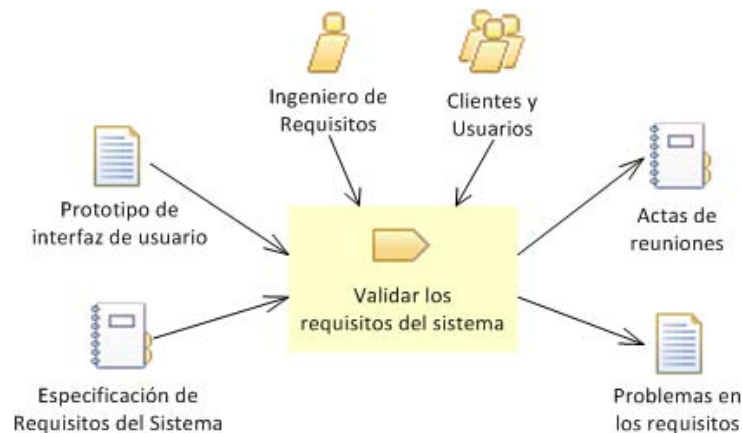


Figura 3.2.1 Levantamiento de requerimientos

La especificación de requerimientos suministra al desarrollador y al cliente, los medios para valorar el cumplimiento de resultados, procedimientos y datos, una vez que se haya construido. La tarea de análisis de los requerimientos es un proceso de descubrimiento y refinamiento, el cliente y el desarrollador tienen un papel activo en la ingeniería de requerimientos de software. El cliente intenta plantear un sistema que en muchas ocasiones es confuso para él, sin embargo, es necesario que describa los datos, que especifique las funciones y el comportamiento del sistema que desea. El objetivo es que el desarrollador actúe como un negociador, un interrogador, un consultor, o sea, como persona que consulta y propone para resolver las necesidades del cliente.

El análisis de requerimientos proporciona una vía para que los clientes y los desarrolladores lleguen a un acuerdo sobre lo que debe hacer el sistema. La especificación, producto de este análisis proporciona las pautas a seguir a los diseñadores del sistema.

Requerimientos Generales



Capítulo 3. Análisis y planteamiento del problema



- Facilidad de uso es uno de los requerimientos más importantes ya que el personal médico (enfermeros, médicos, terapeutas) no siempre tiene un contacto tan recurrente con el equipo de cómputo como el personal administrativo.
- El diseño del sistema web debe ser agradable a la vista, debe ser totalmente responsivo, es decir, que se adapte al dispositivo usado por el personal del hospital, que va desde un equipo de escritorio usado mayoritariamente por el personal administrativo, como el uso de Smartphone y tablets usados por enfermeros y médicos debido a su movilidad.
- Documentación: Entregar junto con el sistema un manual de usuario con lujo de detalle, es decir, que muestre el paso a paso cómo puede realizar cualquier acción dentro del sistema, de preferencia tenerlo en línea para definir un apartado de preguntas y respuestas para ayudar a otros usuarios con problemas similares.
- Tecnología: dar al personal del hospital un sistema web con los últimos estándares del mercado y no tenga problemas para ejecutarlo en el navegador de su preferencia y poder hacer las entradas vía teclado físico o touchscreen.
- Integración: el sistema no debe ser una solución aislada, es decir, que pueda alimentar una fuente de entrada de otro sistema, por medio de web services como un sistema administrativo y pueda extraer información del sistema médico.
- Flexibilidad: El sistema debe ser lo bastante flexible para que cualquier desarrollador de software ajeno al desarrollador principal pueda hacer adecuaciones si así se desea, por medio de los modelos del sistema descritos en el inciso 4.1.
- Bitácora: El sistema debe registrar toda la actividad que se realiza en el llenado del expediente así como su consulta, para evitar mal uso de la información o



Capítulo 3. Análisis y planteamiento del problema



modificaciones no adecuadas, teniendo así una integridad completa de la información.

- Aspectos apegados a la normatividad: como se ha mencionado en el inciso 1.2 todo debe estar apegado a la normatividad de los hospitales, para no tener problemas de tipo legal o fiscal.
- El tiempo de respuesta del sistema al realizar alguna acción por parte del usuario debe ser el mínimo posible, y siempre debe estar en línea el sistema.
- Seguridad: tener un sistema administrado por roles.

Requerimientos Particulares

- El registro, modificación, eliminación de catálogos solo podrán hacerlo jefes/supervisores de enfermería, catálogo de servicios médicos, catálogo de diagnósticos médicos, catálogo de procedimientos de enfermería, catálogo de pisos y camas, el registro de pacientes se hace por medio de un sistema administrativo.
- Se deben recabar las firmas del personal de enfermería para ponerlas en formato electrónico, para su inclusión en el expediente electrónico por medio de un dispositivo mostrado en la figura 3.2.2.



Figura 3.2.2 Dispositivo para capturar firma electrónica.

- El sistema debe registrar comorbilidad, peso, talla, alergias, signos vitales, escala del dolor, dietas, ingresos, egresos, medicamentos, escala glasgow,



Capítulo 3. Análisis y planteamiento del problema



procedimientos de enfermería, requisitos de la salud, escala maddox, fecha de instalación de sondas, planeación del alta, observaciones.

- El sistema debe permitir al enfermero de campo, no invertir demasiado tiempo en el llenado del expediente clínico de enfermería para no descuidar la atención al paciente, por lo tanto deben existir atajos e información previamente cargada en el sistema para acelerar su llenado como el peso, la talla, alergias, dietas, el sistema debe ir aprendiendo de los expedientes registrados de días anteriores del paciente durante su periodo de hospitalización, para que las modificaciones del día siguiente sean mínimas.
- El enfermero tiene a su cargo la atención de varios pacientes a los cuales debe elaborar su expediente electrónico, es por ello que el sistema debe permitir encontrar rápidamente todos los registrados por él, por medio de un apartado que se llame Mis Hojas del día.
- Una vez empezado el llenado del expediente clínico de enfermería, el guardado de la información debe ser de manera automática, evitando así la preocupación de que si guardo o no sus cambios.
- El llenado de la hoja es colaborativo, se realizará por medio de 3 turnos, el sistema, solo permitirá modificar o eliminar los registros de su autoría, pero podrá ver el trabajo de sus demás compañeros como solo lectura.
- El registro de signos vitales debe ser por hora y debe contemplar: temperatura, frecuencia cardiaca, tensión arterial, frecuencia respiratoria, laboratorio y gabinete.
- El registro de la escala del dolor debe ser por hora y totalmente visual, usando imágenes que representen el estado del dolor, ver figura 3.2.3.



Capítulo 3. Análisis y planteamiento del problema

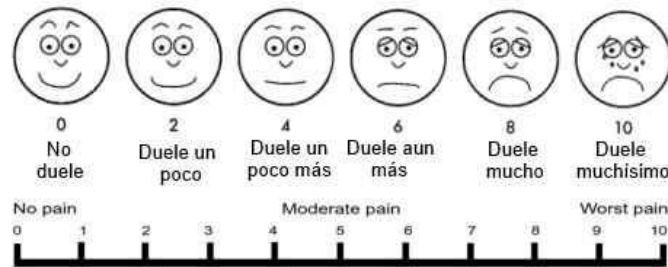


Figura 3.2.3 Escala del dolor gráfico

- El registro de ingresos, egresos, medicamentos debe ser por hora y se debe automatizar de tal forma, que una primera vez registrado el nombre, el sistema ayude al enfermero autocompletando el nombre en futuros registros.
- El registro de los procedimientos de enfermería es por turno, debe permitir al enfermero, poder encontrar de manera rápida el rubro donde se encuentran los procedimientos por medio de palabras clave, ya que el catálogo actual de rubros con procedimientos puede llegar a ser muy extenso.
- Al final del llenado del expediente clínico en cualquier turno, es necesario implementar un mecanismo donde se pida un password asociado a una firma digital del enfermero de campo que registra.
- El jefe o supervisor de enfermería de cualquier turno debe tener un módulo donde puede consultar de manera ágil los expedientes digitales registrados por los enfermeros de campo y poder hacer correcciones del turno correspondiente si así lo requiere y poder ingresar el password de su firma digital para acreditar que está conforme con lo registrado por el enfermero de su turno.
- El sistema debe permitir al enfermero de campo o el jefe/supervisor poder consultar expedientes clínicos de pacientes que ya fueron dados de alta, esto por auditorías internas y externas realizadas en el hospital.



Capítulo 3. Análisis y planteamiento del problema

- El enfermero debe poder visualizar de manera digital en formato pdf el expediente clínico de enfermería todo el llenado realizado tanto por él como por sus compañeros de turno y visualizar las firmas digitales de compañeros, jefes y supervisores de enfermería, apegado al formato escrito que se maneja actualmente, ver figura 3.2.4.

NOMBRE <u>Prado Flores Luis</u>		FECHA NAC <u>15/10/2000</u>		EDAD <u>15</u>		SEXO <u>H</u>		CAMA <u>1</u>																	
REGISTRO <u>N-00000000000</u>		SERVICIO <u>SERVICIO1</u>		DIAGNOSTICO <u>DIAGNOSTICO1</u>																					
DIAS DE ESTANCIA <u>0</u>				FECHA <u>26/07/2015</u>																					
LABORATORIO Y GABINETE	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7	
S																									
I																									
G																									
N																									
O																									
S																									
V																									
I																									
T																									
A																									
L																									
E																									
S																									
	PESO: <u>80.0Kg.</u>										TALLA: <u>1.8m.</u>														

Figura 3.2.4 Fragmento Expediente Clínico Enfermería Digital

- Se debe dar un periodo de gracia para el llenado completo del expediente, es decir, que el cierre de esta será hasta el día de mañana hasta las 12 am, una vez pasado este tiempo en automático la hoja será sólo accesible para consulta.

3.3 Planteamiento de la solución y posibles módulos.

La seguridad en un sistema de cómputo vía web debe cumplir dos procesos básicos que son la autenticación y la autorización, definiéndose autenticación como el proceso por el cual un usuario o servicio tiene que autenticarse para poder acceder a ciertos servicios que ofrece nuestro sistema y la autorización como el proceso por el cual se controlan las acciones que tiene un usuario o servicio normalmente ya autenticado puede realizar, para ello se le conceden o deniegan permisos. El sistema hace uso de Spring Security que viene como plugin del framework web Grails, el proceso se observa en la figura 3.3.1



Capítulo 3. Análisis y planteamiento del problema

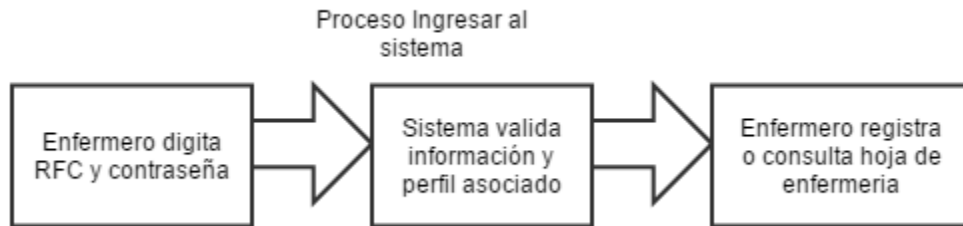


Figura 3.3.1 Diagrama de autenticación y autorización del sistema

Para el llenado del expediente clínico de enfermería una vez realizado el login, el enfermero de campo tendrá en su barra de herramientas dos botones iniciales, el botón Nuevo le ayudará a dejar el sistema sin ningún tipo de información para consultar o registrar un nuevo expediente clínico, el botón Mis Hojas le permitirá tener por medio de un único filtro de búsqueda que es turno, el localizar los expedientes más recientes realizados por él, como se observa en la figura 3.3.2.

Cualquier enfermero de campo, que ingrese al sistema podrá buscar el expediente clínico de enfermería de un paciente por nombre, número de registro o cama con dos filtros previos que son el piso o el servicio en el que se encuentra actualmente hospitalizado, así mismo se da la posibilidad de cargar registros históricos, es decir, le permitirá al enfermero abrir expedientes clínicos de pacientes que ya fueron dados de alta, una vez seleccionado el paciente aparecerá el botón Abrir y dando clic sobre este aparecerá un dialogo con todas las hojas disponibles asociadas al paciente para su posterior carga dependiendo de la fecha de elaboración seleccionada, como se observa en la figura 3.3.2.

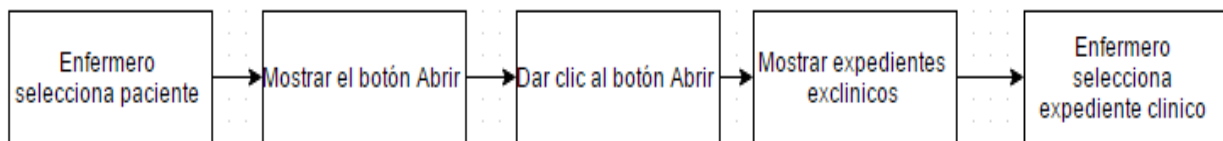


Figura 3.3.2 Diagrama selección de expediente clínico.

Para iniciar el registro del expediente clínico de enfermería se debe haber seleccionado un paciente hospitalizado, donde nos traerá toda la información asociada al paciente



Capítulo 3. Análisis y planteamiento del problema



como lo es la edad, peso, talla, sexo, religión, número de cama, días hospitalizado al momento, a qué servicio médico está asociado, así como el diagnóstico que le adjudicaron, la primera pestaña alergias y comorbilidad es el parteaguas del sistema para que se habiliten las demás pestañas de registro, es decir, existirá un botón Guardar/Firmar una vez puesto el password de la firma digital y validar la confirmación de registro tendremos acceso a la demás pestañas para su captura o modificación siempre y cuando el enfermero sea dueño de los registros, ver figura 3.3.3.



Figura 3.3.3 Módulo Alergias y Comorbilidad

Modulo Signos Vitales

Este módulo estará dividido en tres secciones, el primero es propiamente el registro de signos vitales, escala del dolor y dietas, ver figura 3.3.4.

Signos Vitales: está dividido en seis categorías asociadas a una hora en particular.

- Temperatura
- Frecuencia Cardíaca
- Tensión Arterial Sistólica
- Tensión Arterial Diastólica
- Frecuencia Respiratoria
- Laboratorio y Gabinete



Capítulo 3. Análisis y planteamiento del problema



Escala del dolor: Está definido entre un rango de 0-10 donde 0 denota ausencia de dolor y 10 dolor extremo, el enfermero indicará la hora de revisión y posteriormente dará clic en la imagen que corresponda, aquí mismo podrá consultar los registros realizados por otros enfermeros y puede eliminar los que él haya realizado.

Dieta: Se definen 4 cajas de texto donde de acuerdo al turno el enfermero registra la dieta puesta para ese día.



Figura 3.3.4 Módulo Signos Vitales

Modulo Control de Líquidos y Medicamentos

Este módulo estará dividido en 4 secciones, todos comparten dos botones, el botón Agregar que darán de alta los registros y el botón Mostrar que desplegará los registros al momento realizados en los diferentes turnos, permitiendo borrar aquellos que son de su propiedad, ver figura 3.3.5.

- Ingresos: Se tendrán dos renglones default que se detectó son recurrentes en el llenado del expediente, adicional a eso dando un clic en el botón Agregar Ingreso, se irán agregando renglones para escribir el nombre del ingreso y se escogerá el rango de aplicación de hora para posteriormente anotar la cantidad aplicada en mililitros, se tiene sección de llenado de Faltante por Pasar una por turno.



Capítulo 3. Análisis y planteamiento del problema



- Egresos: Todos los registros de este apartado son fijos teniendo cierta particularidad los apartados de Fuga que están representados con la notación de +(Escaso), ++(Mucho), +++(Abundante).
- Medicamentos: Tiene el botón de Agregar Medicamento, que irá agregando renglones, para posterior registro del nombre, hora de aplicación y dosis aplicada, el sistema nos dará autocomplete de los medicamentos ya registrados, también nos da la posibilidad de cambiar el nombre del medicamento si así lo requiere.
- Escala Glasgow y Otros: Tendrá 6 renglones default, así como agregar otro tipo de renglones con la mecánica de seleccionar la hora aplicación y el valor.



Figura 3.3.5 Módulo Control Líquidos y Medicamentos

Módulo Valoración de Enfermería y Diagnósticos e Intervenciones

Nos aparecerán tablas HTML que representan los rubros asociados a el segmento de valoración de enfermería y diagnósticos e intervenciones con procedimientos asociados por rubro, donde podrá marcarlo de acuerdo al turno seleccionado, por cada rubro se tiene la opción de Otro donde el enfermero pueda anotar algún procedimiento extra asociado a ese rubro médico, la sección de requisitos de desarrollo y de desviación de la salud son dos cuadros de textos donde se escribirá todo lo referente a ellos en el módulo de valoración de enfermería, ver figura 3.3.6.

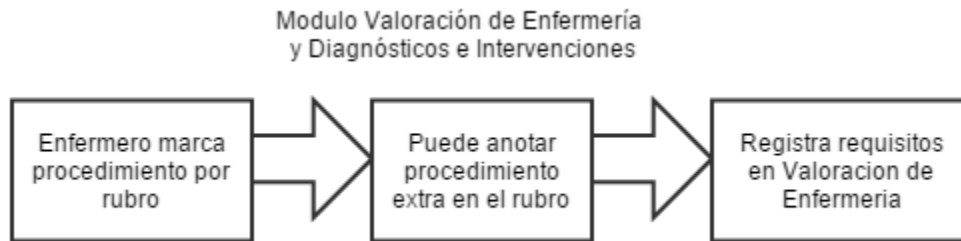


Figura 3.3.6 Módulo Valoración de Enfermería y Diagnósticos e Intervenciones

Módulo Indicadores de Calidad

Se manejan tablas HTML tipo check y option con el apartado de registro de Otro para anotar cualquier procedimiento asociado al rubro, en la sección de escala de Maddox el enfermero podrá registrar un valor entre 0-5 para su turno correspondiente, hay una sección donde podrá anotarse la instalación de Accesos Venosos y Sondas Vesicales como son la fecha, los días consecutivos el sistema los calculara a partir de la fecha, material, calibre y globo. El enfermero tiene un cuadro de texto asociado a su turno para registrar la planeación del alta y observaciones encontradas en ese día, ver figura 3.3.7.

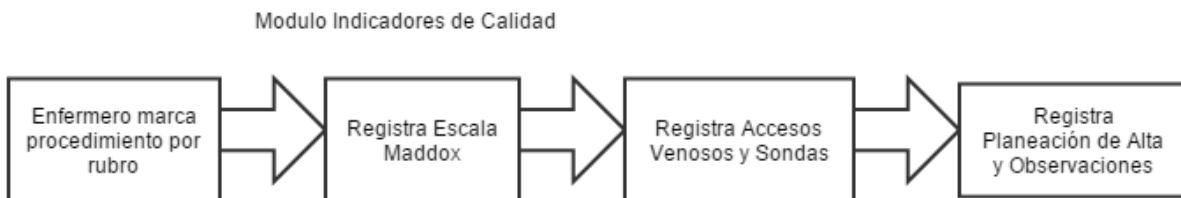


Figura 3.3.7 Módulo Indicadores de Calidad

Módulo Jefes/Supervisores

El sistema se maneja por medio de roles y el rol de jefe o supervisor de turno tendrá un panel de control, donde se muestre de manera gráfica todos los pisos del hospital y por cada piso los pacientes que ya cuenten con expediente clínico, para poder generar el expediente en formato pdf, modificar los registros ya sea por error de dedo u otro caso de lo capturado por el enfermero de campo y poner el password de la firma digital cuando haya conformidad de los registros capturados, también tendrá posibilidad de



consultar expedientes clínicos históricos, es decir, expedientes clínicos de pacientes que ya fueron dados de alta, ver diagrama 3.3.8.

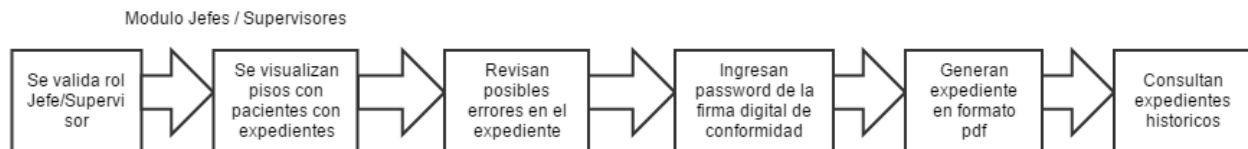


Figura 3.3.8 Módulo Jefes / Supervisores

3.4 Posicionamiento del software a utilizar.

Para la elección del software a utilizar se consideraron varios aspectos del proyecto como son; alcance, magnitud, adaptabilidad, cantidad de usuarios, licencias, etc. De este modo seleccionamos PostgreSQL como gestor de bases de datos, Groovy / Grails como lenguaje de programación y Apache Tomcat como servidor web.

Gestores de base de datos PostgreSQL.

Para la base de datos, realizamos una comparación de PostgreSQL con MySQL, ambos gestores poseen características similares, sin embargo cada uno está orientado a cubrir necesidades diferentes.

PostgreSQL¹⁴ se ha enfocado tradicionalmente en la fiabilidad, integridad de datos y características integradas enfocadas al desarrollador. Tiene un planificador de consultas extremadamente sofisticado, que es capaz de unir cantidades relativamente grandes de tablas eficientemente.

Intenta ser un gestor de bases de datos de mayor nivel a la altura de Oracle, Sybase o Interbase.

¹⁴ Manual de referencia y documentación del gestor de base de datos PostgreSQL 9.3
<http://www.postgresql.org/docs/9.3/static/index.html>



Capítulo 3. Análisis y planteamiento del problema



MySQL¹⁵, por otra parte, se ha enfocado tradicionalmente en aplicaciones web de lectura mayormente, usualmente escritas en PHP, donde la principal preocupación es la optimización de consultas sencillas.

El principal objetivo de MySQL es la velocidad, lo que lo convierte en uno de los mejores gestores con mayor rendimiento a la hora de realizar operaciones, sin requerir un consumo elevado de recursos

En la tabla 3.4.1 Se muestran las diferencias entre los gestores PostgreSQL y MySQL.

Características	Gestores de base de datos	
	PostgreSQL	MySQL
Sistemas de gestión de bases de datos	Objeto relacional	Relacional
Modelo de integridad	ACID, MVCC	ACID
Tipo de bloqueo	MVCC	Lock free model
Licencia	BSD	Licencia publica general GNU
Multihilo	No	Sí
Claves compuestas	Sí	Sí
Multiprocesos	Sí	Sí
Lenguajes de programación soportados	C, C++, Java, Python, Php, Ruby, Grooby, Etc.	C, C++, Java, Php, Ruby, Etc.
Transacciones	Sí	Sí
Seguridad	Mayor	Mayor
Implementación del estándar sql92-sql99	Sí	No
Gran escalabilidad	Sí	Sí

¹⁵ Manual de referencia y documentación del gestor de base de datos MySQL 5.7 <http://dev.mysql.com/doc/refman/5.7/en/>



Capítulo 3. Análisis y planteamiento del problema



Soporta distintos tipos de datos	Sí	Sí
Velocidad a la hora de realizar operaciones	Lenta	Mayor
Integridad referencial	Puede comprobarla	No puede comprobarla
Soporta un gran uso de datos	Sí	Sí
Configuraciones e interfaz de usuario	Menor facilidad	Mayor facilidad
Transacciones Rollback's	Sí	No
Subconsultas	Sí	No
Alta concurrencia	Sí	Sí
Plataformas soportadas	Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.	Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
Uso de recursos	Mayor uso de los recursos del sistema	Mínimo uso de recursos

Tabla 3.4.1 Diferencias entre PostgreSQL y MySQL

En la comparación entre PostgreSQL y MySQL, para aplicaciones que requieren rapidez de respuesta y bajo consumo de recursos, donde la velocidad y el número de consultas concurrentes sea algo primordial, como lo que sería una aplicación web, MySQL será la mejor opción.

Para sistemas en los que la consistencia de la base de datos sea fundamental, y se requiera una mayor seguridad de la información, PostgreSQL sería una mejor opción pese al gran consumo de recursos y a la lentitud de las operaciones.



Lenguajes de programación Groovy / Grails.

Para desarrollo de la aplicación, utilizaremos el framework Grails¹⁶, el cual es un ambiente de desarrollo de aplicaciones web libres, basadas en el lenguaje de programación Groovy¹⁷.

Groovy es un lenguaje de programación orientado a objetos, que puede ser utilizado como scripts, utiliza una sintaxis muy similar al lenguaje de programación Java y su código puede ser compilado en la JVM (máquina virtual de Java) así como interactuar otros códigos escritos en Java o bibliotecas.

Realizamos una comparación con el framework Symfony¹⁸ que está basado en el lenguaje de programación Php¹⁹.

Php es un lenguaje de programación orientado al desarrollo web de contenido dinámico el cual puede ser incorporado en algún documento HTML (lenguaje de marcas de hipertexto), este código es interpretado por un servidor web con un módulo Php que es capaz de entender el código y generar como resultado una página web.

En la tabla 3.4.2 se muestran las diferencias entre el lenguaje de programación Groovy y Php.

Características	Lenguajes de programación	
	Groovy	Php
Conexión con gestores de	MongoDB	Microsoft Access

¹⁶ Manual de referencia y documentación del framework Grails <https://grails.org/documentation.html>

¹⁷ Manual de referencia y documentación del lenguaje de programación Groovy <http://www.groovy-lang.org/documentation.html>

¹⁸ Manual de referencia y documentación del framework Symfony <http://symfony.com/doc/current/index.html>

¹⁹ Manual de referencia y documentación del lenguaje de programación Php <https://secure.php.net/docs.php>



Capítulo 3. Análisis y planteamiento del problema



base de datos	CouchDB MySQL PostgreSQL Oracle SQLite Elastic Search BigTable Cassandra Drizzle Titan Ehcache MemcacheDB Informix InterBase Microsoft BI CUBRID	PostgreSQL MongoDB CUBRID Microsoft BI InterBase Informix Oracle IBM DB2 SQLite MySQL MariaDB PHP/PDO firebird CouchDB BangDB
Programación orientada a objetos	Sí	Sí, además puede utilizar programación por procedimientos
Plataformas de desarrollo	Grails Eclipse Netbeans IntelliJ	symfony CakePHP Zend Framework Kohana lavarel Yii2
Patrón de diseño	Modelo vista- controlador	Modelo vista- controlador
Compatibilidad con otros lenguajes	Puede interactuar con el lenguaje Java (requiere JVM)	Puede ser incluido dentro del lenguaje HTML



Capítulo 3. Análisis y planteamiento del problema



Integración de bibliotecas externas	Sí	Sí
Lenguaje multiplataforma	Sí	Sí
Capacidad de expansión	Sí, utilizando clases y bibliotecas	Sí, utilizando extensiones
Soporte de comunicación para protocolos LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM, etc.	Sí	Si, además de soporte para otro tipo de protocolos, así como la utilización de objetos Java que utiliza de manera transparente como objeto Php
Seguridad	Menor seguridad y estabilidad	Mayor seguridad y estabilidad
Documentación y soporte	Documentación y soporte amplio	Documentación y soporte bastante amplio
Sintaxis	Sintaxis similar a Java, eliminando la mayor parte del código innecesario que no se requiere para efectuar la acción deseada.	Código e instrucciones sencillas y fáciles de interpretar
Reutilización de código	Sí, (Don't repeat yourself) reutilización de código	No

Tabla 3.4.2 Diferencias entre el lenguaje de programación Groovy y Php

Para aplicaciones que requieren una velocidad de respuesta elevada, Php será la mejor opción pues ofrece gran flexibilidad, estabilidad y velocidad de respuesta en las aplicaciones diseñadas con dicho lenguaje.



Capítulo 3. Análisis y planteamiento del problema



En el caso de Groovy ofrece un enorme alcance de desarrollo al tener la posibilidad de integrarse fácilmente con aplicaciones Java, lo que permite la creación de aplicaciones web innovadoras reduciendo el ciclo de desarrollo y ahorrando tiempo valioso.

Además de utilizar menos líneas de código, menor cantidad de errores, mayor expresividad y mayor flexibilidad,

Para el desarrollo de la aplicación se utilizara Grails lo que permitirá usar los beneficios de Java en el desarrollo de aplicaciones web, dicho framework, está basado en el modelo vista controlador permitiendo un desarrollo ágil de aplicaciones web.

Una de las mayores ventajas de Grails son los plugins, existen muchísimas funcionalidades que se pueden instalar de manera fácil y rápida, muchos de ellos basados anteriormente en Spring, Hibernate, Sitemesh (framework de desarrollo de aplicaciones para la plataforma Java) y modificados para su ejecución en Grails.

En la siguiente lista se muestran las características del framework Grails con el lenguaje de programación Groovy:

- MVC (modelo vista controlador)
- DRY (Don't repeat yourself) reutilización de código
- Inyección de dependencias
- 5 veces menos líneas de código
- Extensa librería de plugins



Servidor web Apache Tomcat.

Para la comparación del servidor web compararemos los servicios ofrecidos por Apache Tomcat²⁰ y Jetty²¹.

Apache Tomcat es un servidor web con soporte de servlets y JSPs, no es un servidor de aplicaciones e incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets.

Apache Tomcat puede funcionar como servidor web por sí mismo, aunque en sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día Apache Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Apache Tomcat fue escrito en lenguaje Java, por lo cual, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Jetty es un servidor HTTP basado en Java y un contenedor de Servlets escrito en Java.

El desarrollo de Jetty se enfoca en crear un servidor web sencillo y eficiente que puede ser fácilmente integrado en los dispositivos, herramientas, marcos de trabajo, servidores de aplicaciones, etc.

El desarrollo de Jetty fue creado en lenguaje de programación Java.

²⁰ Manual de referencia y documentación del servidor web Apache Tomcat <http://tomcat.apache.org/tomcat-8.0-doc/index.html>

²¹ Manual de referencia y documentación del servidor web Jetty <http://www.eclipse.org/jetty/documentation/>



Capítulo 3. Análisis y planteamiento del problema



En la tabla 3.4.3 se muestran las diferencias entre el servidor web apache Tomcat y Jetty.

Características	Servidor Web	
	Apache Tomcat	Jetty
Leguaje de desarrollo	Java	Java
Soporte HTTPS	Sí	Sí
Autenticación	Sí	Sí
Contenedor de Servlet	Sí	Sí
Soporte con entornos de desarrollo	Eclipse, IntelliJ IDEA y Netbeans, Grails	Eclipse, IntelliJ IDEA y Netbeans,
Hospedaje virtual	Sí	Sí
Servidor Java Script	Sí	Sí
Soporte Server Side Includes (SSI)	Sí	Sí
Soporte IPv6	Sí	Sí
Soporte Common Gateway Interface (CGI)	Sí	Sí
Consola de administración	Sí	Sí
Plataformas Soportadas	Windows, Linux, Mac OS	Windows, Linux, Mac OS
Documentación	Documentación en página web, con soporte de terceros y foros.	Documentación en página web, no cuenta con foros, ni soporte de terceros.

Tabla 3.4.3 Comparación de apache Tomcat y Jetty



3.5 Aplicación de la metodología eXtreme Programming –XP al sistema.

Justificación de la metodología XP

Se adopta la metodología **eXtreme Programming XP** (Programación Extrema) ya que se ajusta a las características y requerimientos del proyecto. Como estos requerimientos del sistema son un tanto imprecisos y cambian muy a menudo, se implementó la definición de XP por su simplicidad.

La justificación de la metodología XP en el sistema de expediente clínico, correspondiente a la definición del capítulo 2.5, es la siguiente:

- **Da lugar a una programación sumamente organizada.** La metodología XP permitirá organizar el código de tal manera que los integrantes del equipo pueden modificar el código, reutilizarlo e inclusive hacerlo mucho más simple, con el versionado de fuentes.
- **Ocasiona eficiencias en el proceso de planificación y pruebas.** Como los requerimientos son cambiantes y las pruebas se realizan durante todo el ciclo de vida, los tiempos reales se reducirán y se lograrán los objetivos en las fechas de término y entrega.
- **Cuenta con una tasa de errores muy pequeña.** La metodología XP, es la más destacada de las metodologías de desarrollo ágil de software²², se le exigirá el máximo esfuerzo al programador y los niveles de errores se reducirán al máximo.
- **Propicia la satisfacción del programador.** Como los tiempos y metas se cumplirán de acuerdo al plan, los integrantes del equipo de desarrollo tendrán un alto grado de motivación y satisfacción por el valor del trabajo.

²² Luis Joyanes Aguilar. Ingeniería del software: un enfoque práctico, Mikel Angoar, 1997. P63



Capítulo 3. Análisis y planteamiento del problema



- **Fomenta la comunicación entre los clientes y los desarrolladores.** Como el desarrollo cambiará en varias ocasiones, se tendrá forzosamente comunicación con el cliente y entre el mismo equipo de trabajo.
- **Facilita los cambios.** Se utilizará un sistema de versionado de código para que todos puedan modificar el código fuente en cualquier momento y así la última versión estará en el servidor.
- **Permite ahorrar mucho tiempo y dinero.** XP es la metodología más ágil, por lo tanto en el desarrollo de nuestro sistema, se ahorrará tiempo en las entregas y también costos.
- **Puede ser aplicada a cualquier lenguaje de programación.** En nuestro caso, el sistema estará desarrollado en Groovy Grails. Esta combinación es muy buena, ya que el lenguaje es bastante sencillo y la metodología es muy ágil, logrando excelentes indicadores.
- **Se hacen pruebas continuas durante el proyecto.** Durante el desarrollo, inclusive antes del desarrollo (Pruebas a priori), se realizarán varias pruebas para corregir y prever errores.
- **La XP es mejor utilizada en la implementación de nuevas tecnologías.** Debido a su nivel de complejidad, el constante trabajo en equipo y el desarrollo de módulos en pareja, lograremos reducir nuestros tiempos en el desarrollo del sistema y en la implementación y mantenimiento disminuir costos.
- **Es recomendable emplearla solo en proyectos a corto plazo.** Este punto parecería ser una gran desventaja, pero debido al alto grado de eficiencia de esta metodología, se pueden desarrollar proyectos medianos, como es el



Capítulo 3. Análisis y planteamiento del problema



desarrollo del sistema de expediente clínico y desarrollarlo entre un número pequeño de desarrolladores.

- **En caso de fallar, las comisiones son muy altas.** La garantía del desarrollo es muy alta, es por eso que si existiera una falla, su resolución se cobrará a un costo muy elevado. Por lo que las pruebas serán una parte indispensable para el desarrollo.
- **Requiere de un rígido ajuste a los principios de XP.** Para que sea exitosa la metodología XP, se requiere de cinco principios para que funcione y todo el equipo de trabajo de adapte (Figura 3.5.1):
 - Comunicación. El ejemplo más claro es con la persona que sabe más del lenguaje de programación Groovy, ya que es el que apoya y asesora a todos los demás.
 - Respeto. No debe afectar el desarrollo de los demás en forma negativa, y desperdiciando el tiempo de los compañeros del equipo ni del propio.
 - Simplicidad. En esta metodología, se comienza desarrollando las soluciones o requerimientos de los módulos más sencillos, añadiendo funcionalidad extra más tarde y no se invierte esfuerzo en futuros requerimientos que posiblemente no se vayan a necesitar.
 - Coraje. Se debe ser persistente en la resolución de problemas, realizar las tareas al momento y no dejarlas para después, elementos que no sean necesarios y sobre todo mejorar el sistema que se pudo resolver.
 - Retroalimentación. Se deben tomar seriamente los compromisos con el cliente, realizando las entregas en el tiempo establecido. La retroalimentación se dio para los integrantes del equipo, opinando sobre sus métodos de programación y posibles errores; retroalimentación del sistema, este proporciona reportes de las pruebas unitarias y de



integración; y retroalimentación del cliente, mediante las pruebas de aceptación.

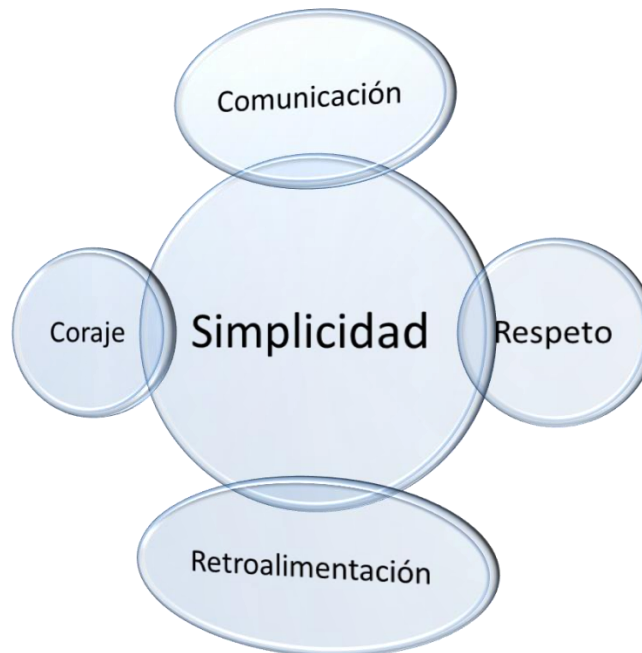


Figura 3.5.1. Principios de la metodología XP

- **Puede no siempre ser más fácil que el desarrollo tradicional.** Debido a que esta metodología es compleja, cuesta mucho trabajo apegarse a los valores y definiciones, por lo tanto, para lograr la eficiencia del sistema y lograr disminución de costos, el equipo se tiene que esforzar al máximo.

Aplicación de la metodología XP

El sistema de expediente clínico de enfermería se entregará en tres bloques. Dichos bloques serán desarrollados por separado, ya que la dependencia es mínima y al final se integrarán por partes.

Cada módulo será desarrollado en pareja para compartir y completar conocimientos, además de agilizar el desarrollo. Debido al aseguramiento de la calidad, se realizarán pruebas en todo momento, desde el análisis, diseño y desarrollo.



Capítulo 3. Análisis y planteamiento del problema



El primer bloque será el desarrollo del módulo principal de autenticación para ingresar al sistema, integrado con el desarrollo de los módulos de cuidados y registros básicos (Figura 3.5.2). El módulo de ingreso al sistema va a convivir con los tres primeros Módulos de alergias y comorbilidad, Módulo de signos vitales, y Modulo de control de líquidos y medicamentos, debido al orden en que se va a llenar el expediente clínico y cuidados enfocados al paciente. Esta fase está enfocada a usuarios con perfil de enfermero y de los registros principales obtenidos del estado del paciente.

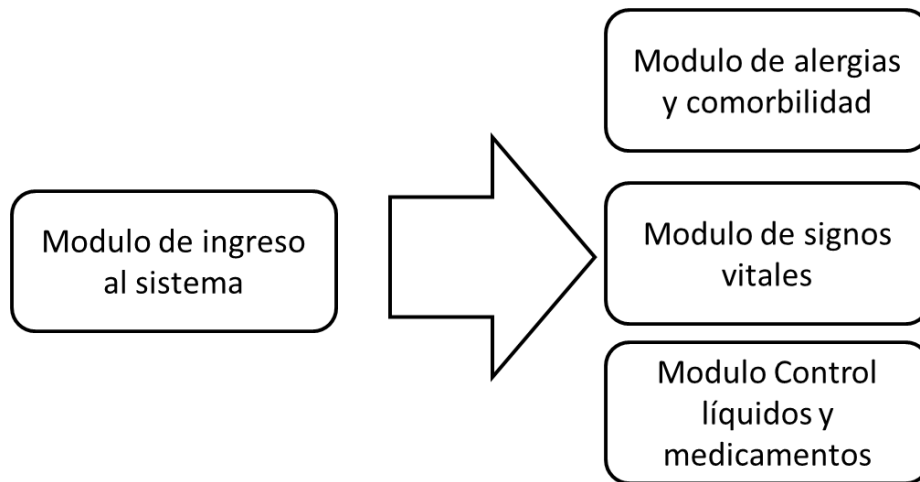


Figura 3.5.2 Bloque de integración del módulo de autenticación con los módulos de registro básico.

El segundo bloque se realizará integrando los módulos de valoración de enfermería y diagnóstico e intervenciones a lo ya desarrollado (Figura 3.5.3). Esta separación de módulos se debe a que se trata de creación de estadísticas, resultados y reportes asociados a los registros que se tienen en la primera fase. Además es donde los enfermeros pueden registrar el monitoreo de la terapia intravenosa en la escala de Maddox y donde se registran las altas y observaciones. En esta fase se hacen pruebas antes y después de la integración para validar su funcionamiento e interacción.



Capítulo 3. Análisis y planteamiento del problema

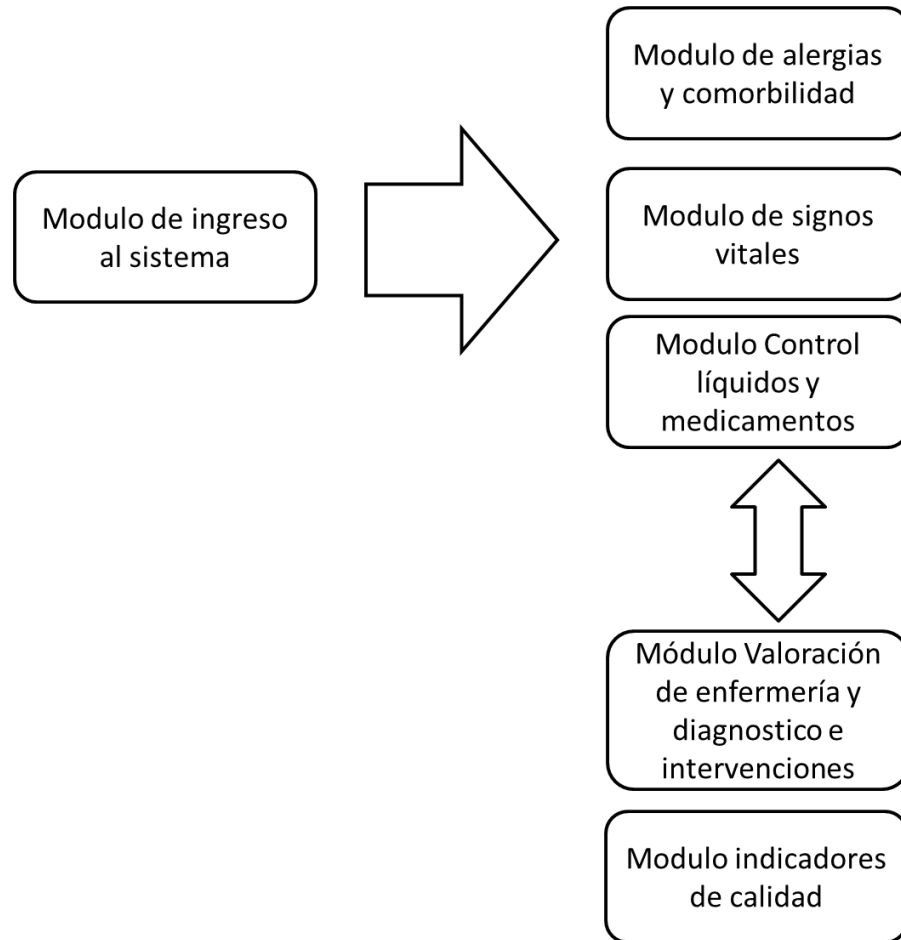


Figura 3.5.3 Boque de la integración con los módulos de estadísticas e indicadores.

El tercer bloque es la interacción de lo que se desarrolla con el módulo de jefes y supervisores (Figura 3.5.4). Este módulo tiene por separado la autenticación para los usuarios de perfil de jefes o supervisores, únicamente pueden ingresar los usuarios con este perfil. Se recicla parte del código del módulo de Modo de ingreso al sistema y al entrar no llega al módulo de registros básico, sino al sub módulo para monitorear los registros clínicos, corrección de errores, entre otras funciones. Finalmente este módulo tendrá convivencia con el resto del bloque para realizar consultas, supervisar a los enfermeros y generar los reportes del sistema.



Capítulo 3. Análisis y planteamiento del problema

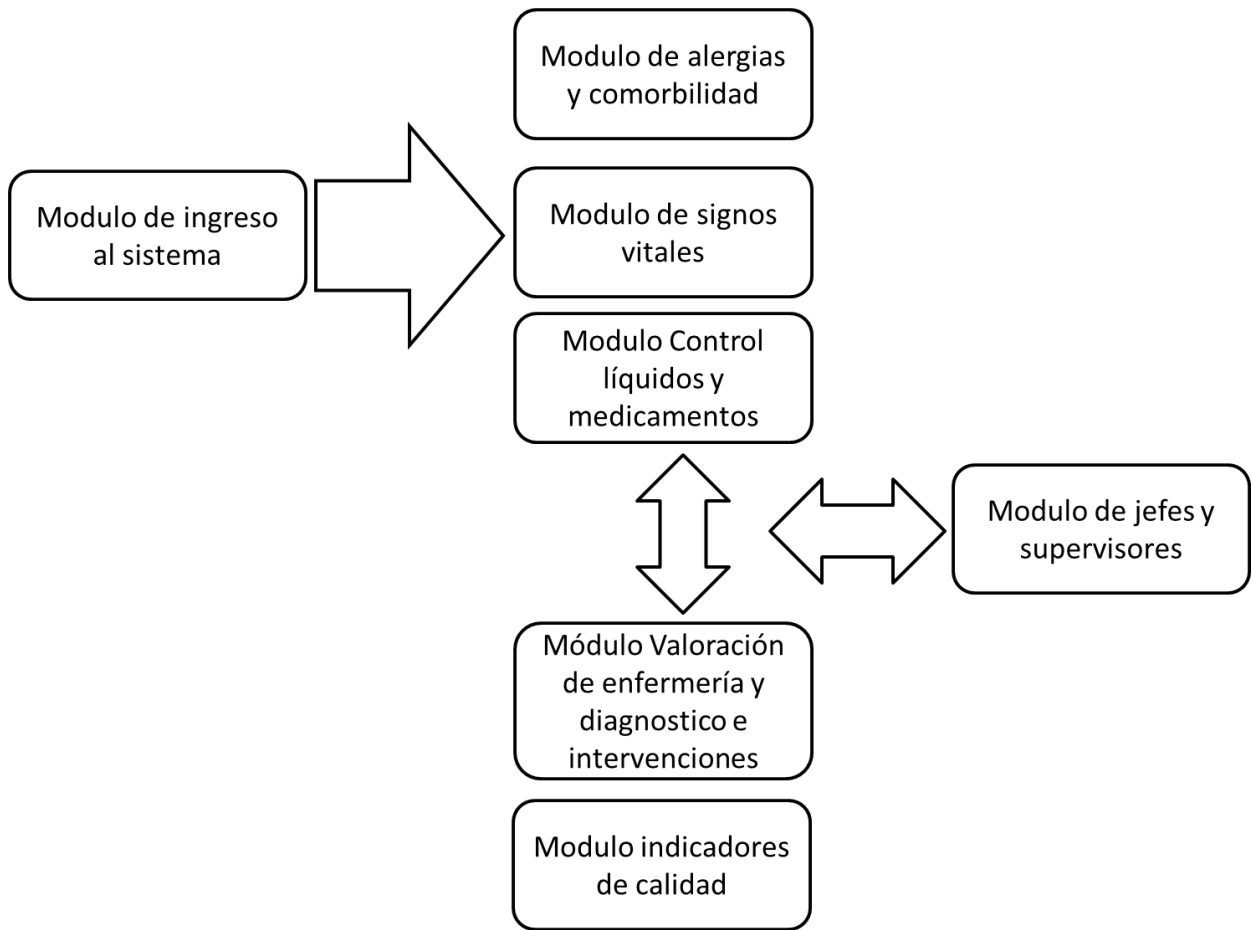


Figura 3.5.4 Bloque de la integración con el módulo de jefes y supervisores.

CAPÍTULO 4.
DISEÑO Y CONSTRUCCIÓN
DEL SISTEMA



4.1 Modelos del sistema.

4.1.1 Ciclos de desarrollo.

En el inciso 3.3 se detectaron los posibles módulos de acuerdo a los requerimientos generales del sistema del inciso 3.2, estos módulos de acuerdo a la metodología de desarrollo iterativo creciente y extreme programming-xp, se implementarán de manera progresiva basados en un análisis y diseño inicial y con constantes pruebas, con una estrecha comunicación con jefes y/o supervisores de enfermería, enfermeros de campo para ir refinando el sistema.

El equipo inicial de desarrollo está conformado:

- Un Administrador de Base de Datos.
- Un Arquitecto de Software.
- Un Ingeniero de Software.
- Un Ingeniero en Pruebas.
- Un Ingeniero de requerimientos.
- Programadores Java/Groovy

Para poder hacer una primera entrega, se necesita el módulo de autenticación y autorización funcional que sería prioridad uno. El arquitecto de software sugiere no reinventar la rueda para este módulo, es por eso que sugiere al ingeniero de software revisar la página de plugins de Grails para reutilizar los que se puedan, para este caso específico se usará el plugin de Spring Security como se observa en la figura 4.1.1.1.



Capítulo 4. Diseño y construcción del sistema



Spring Security Core Plugin

Plugin Collection

129 votes



Tags: [security](#), [spring security](#), [grails2.4](#) | Latest: 2.0-RC5 | Last Updated: 04 junio 2015 | Grails version: 2.3.0 > * 51% of Grails users

[Spring Security Core plugin](#) [Read more](#)

Dependency :

```
compile "spring-security-core:2.0-RC5"
```

[Documentation](#)

[Source](#)

[Issues](#)

Figura 4.1.1.1 Plugin Spring Security

Se acordó con los jefes del hospital tener un área para el equipo de desarrollo del sistema, ya que es indispensable que por lo menos una vez en la semana se puedan tener revisiones de los avances con algún jefe o supervisor de enfermería, de acuerdo a la metodología y se firmen documentos de conformidad.

Después de la entrega de la primera versión de autenticación y autorización, la entrega de los demás módulos debe ser una cada 2 semanas al ingeniero de pruebas para que ejecute la matriz de pruebas correspondiente, y corregir las posibles fallas, es decir, se debe estar entregando al cliente una versión alpha cada 3 semanas de un módulo implementado.

El administrador de base de datos trabajara en pareja con el Ingeniero de Software y programadores Java/Grails ya que como se usa Grails el modelo del dominio generará las tablas, llaves primarias, foráneas, índices, etc. el administrador de base de datos debe revisar este componente de Grails.

El Ingeniero de Pruebas debe familiarizarse con el componente Test de Grails y manejar una herramienta basada en Web para el seguimiento de errores como Bugzilla.



Capítulo 4. Diseño y construcción del sistema



Los programadores Java / Grails deben realizar pruebas unitarias, una vez validadas por el Ingeniero de Software, subir sus cambios a un sistema de versiones en la nube como Bitbucket, ver figura 4.1.1.2.

[Features](#)[Pricing](#)[Log in](#)[Get started](#)

Your team, your code, connected

Git and Mercurial code management for teams

Figura 4.1.1.2 Versionador en la nube de código fuente Bitbucket

4.1.2 Diagrama de contexto.

Este diagrama, permite de una manera global tener un entendimiento abstracto de lo que será el sistema. En la figura 4.1.2.1, se muestra el diagrama de contexto del sistema, en éste se expresa de manera modular su relación con los diferentes tipos de usuarios, las bases de datos origen y receptora, así como los sistemas que se alimentan de esta información.

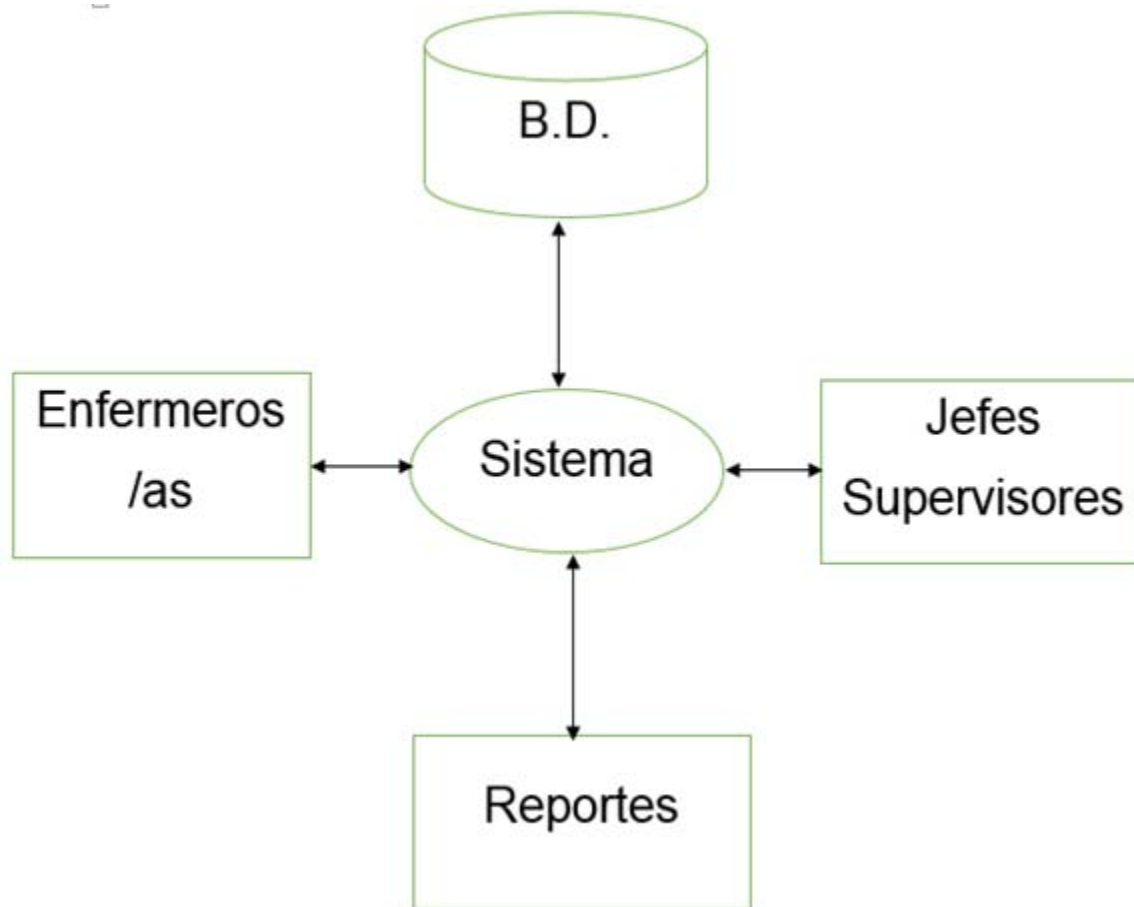


Figura 4.1.2.1 Diagrama de contexto

4.1.3 Diagrama de caso de usos.

Como se ha mencionado el modelo de casos de uso suele servir, para delimitar el alcance del sistema, esbozar quiénes interactúan con el sistema, visualizar cuáles son las funcionalidades esperadas y para revisar los requisitos solicitados por el cliente.

Un diagrama de casos de uso, muestra todas las interacciones entre casos de uso de un sistema o subsistema. Una de las grandes fuerzas del diagrama de caso de uso es su utilidad para definir el contexto de un sistema antes de construirlo. Los diagramas de

casos de uso siguientes son los principales para el análisis y modelado del sistema y se enuncian a continuación:

La figura 4.1.3.1 muestra el módulo de acceso al sistema que debe realizar el enfermero para poder continuar.

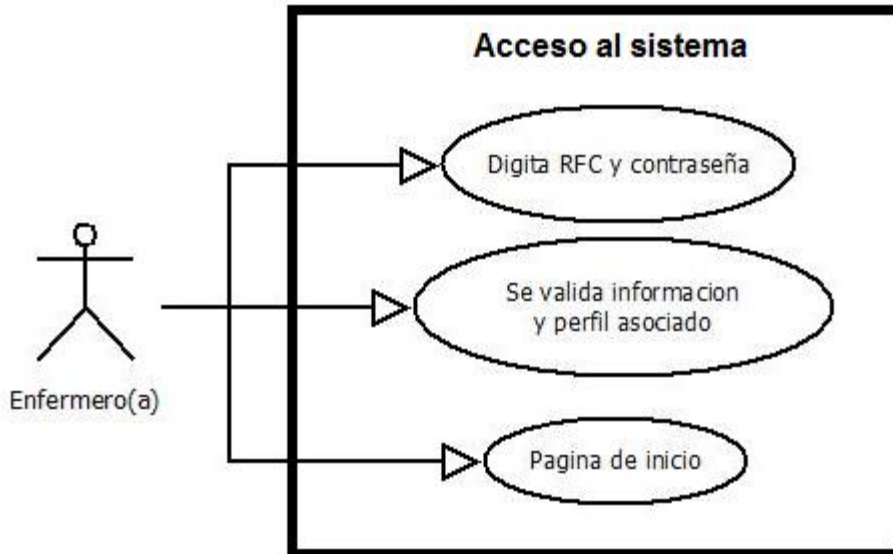


Figura 4.1.3.1 Acceso al sistema

En la figura 4.1.3.2 vemos que una vez que el enfermero accedió al sistema pasando por el módulo de acceso puede entrar a ver las hojas seleccionando el turno y se muestra la información de la hoja deseada.



Figura 4.1.3.2 Muestra de información

El siguiente caso muestra cómo podemos hacer la búsqueda de una paciente ingresando su nombre, registro o el número de cama en que se encuentra. Vea figura 4.1.3.3

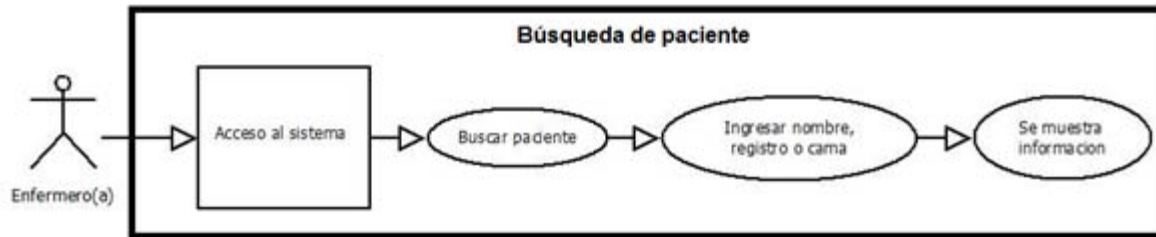


Figura 4.1.3.3 Búsqueda de paciente

A continuación vemos en la figura 4.1.3.4 vemos como es el proceso para crear un nuevo registro en el sistema y en la figura 4.1.3.5 vemos como es el proceso para registrar las alergias y afecciones del paciente. En la figura 4.1.3.6 se continúa con el llenado de los signos vitales para su registro en el expediente.

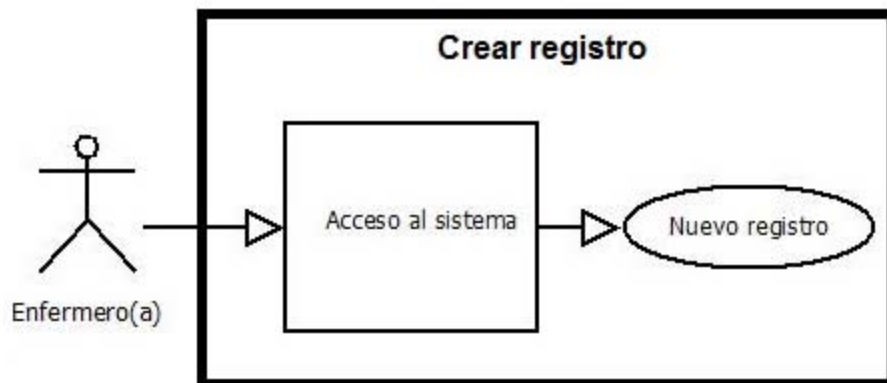


Figura 4.1.3.4 Crear registro



Figura 4.1.3.5 Registrar alergias y afecciones

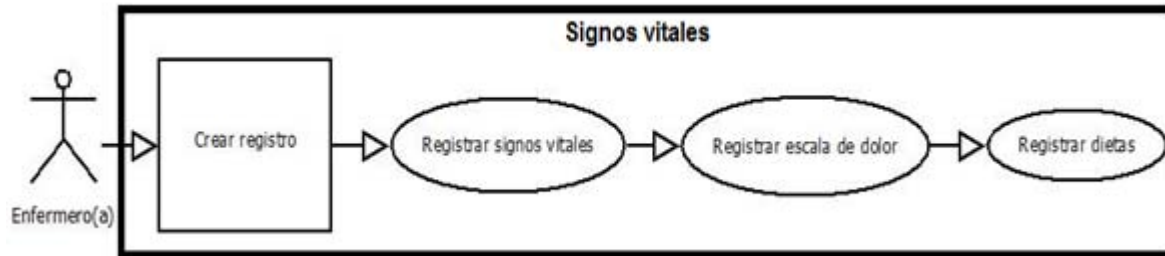


Figura 4.1.3.6 Signos vitales

En el caso siguiente de la figura 4.1.3.7 se puede registrar el control de líquidos y medicamentos que han sido suministrados al paciente desde su ingreso.

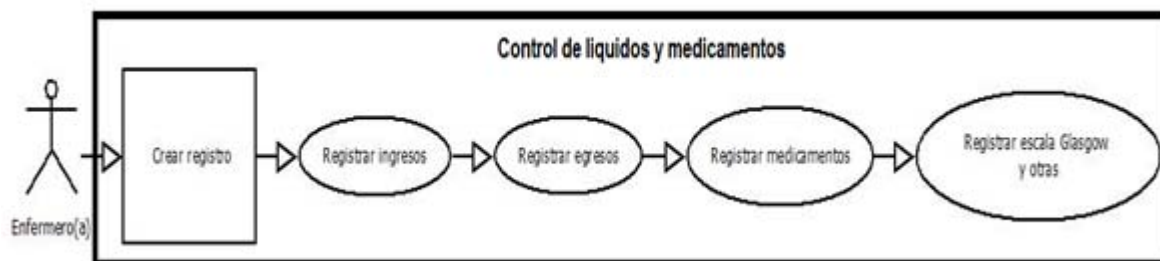


Figura 4.1.3.7 Control de líquidos y medicamentos

La figura 4.1.3.8 muestra como es el proceso para registrar la valoración de un paciente durante su hospitalización.

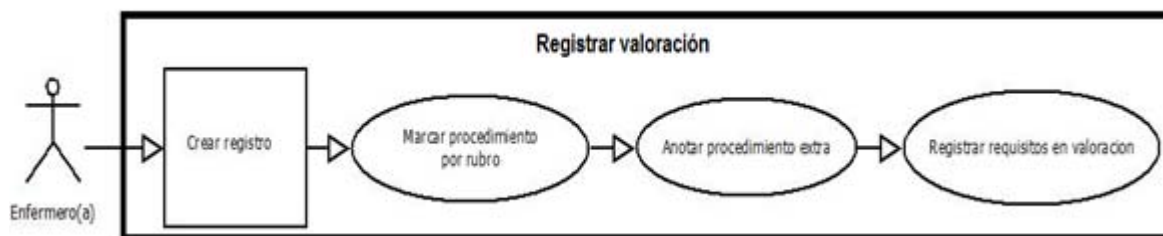


Figura 4.1.3.8 Registrar valoración

A continuación en este caso vemos cómo es que podemos registrar los indicadores de calidad que ha tenido el paciente. Vea figura 4.1.3.9



Figura 4.1.3.9 Registrar indicadores de calidad

4.1.4 Diagrama Entidad-Relación.

El modelo Entidad Relación está basado en dos conceptos fundamentales: el de entidad y el de relación. Y otros dos que ayudan a su comprensión: el dominio y los atributos.

Entidad: cualquier objeto, real o abstracto sobre el cual queremos tener información en la base de datos y que tiene existencia por sí mismo.

Relación: asociación o correspondencia entre entidades.

Dominio: conjunto de posibles valores que pueden tomar una propiedad o característica de una entidad. Por ejemplo, el valor católico, se toma del dominio religión.

Atributo: cada una de las propiedades que tiene un tipo de entidad o un tipo de interrelación. (Son los campos).

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas, ver Figura 4.1.3.1 del Diagrama Entidad Relación del Expediente Clínico de Enfermería.

Capítulo 4. Diseño y construcción del sistema

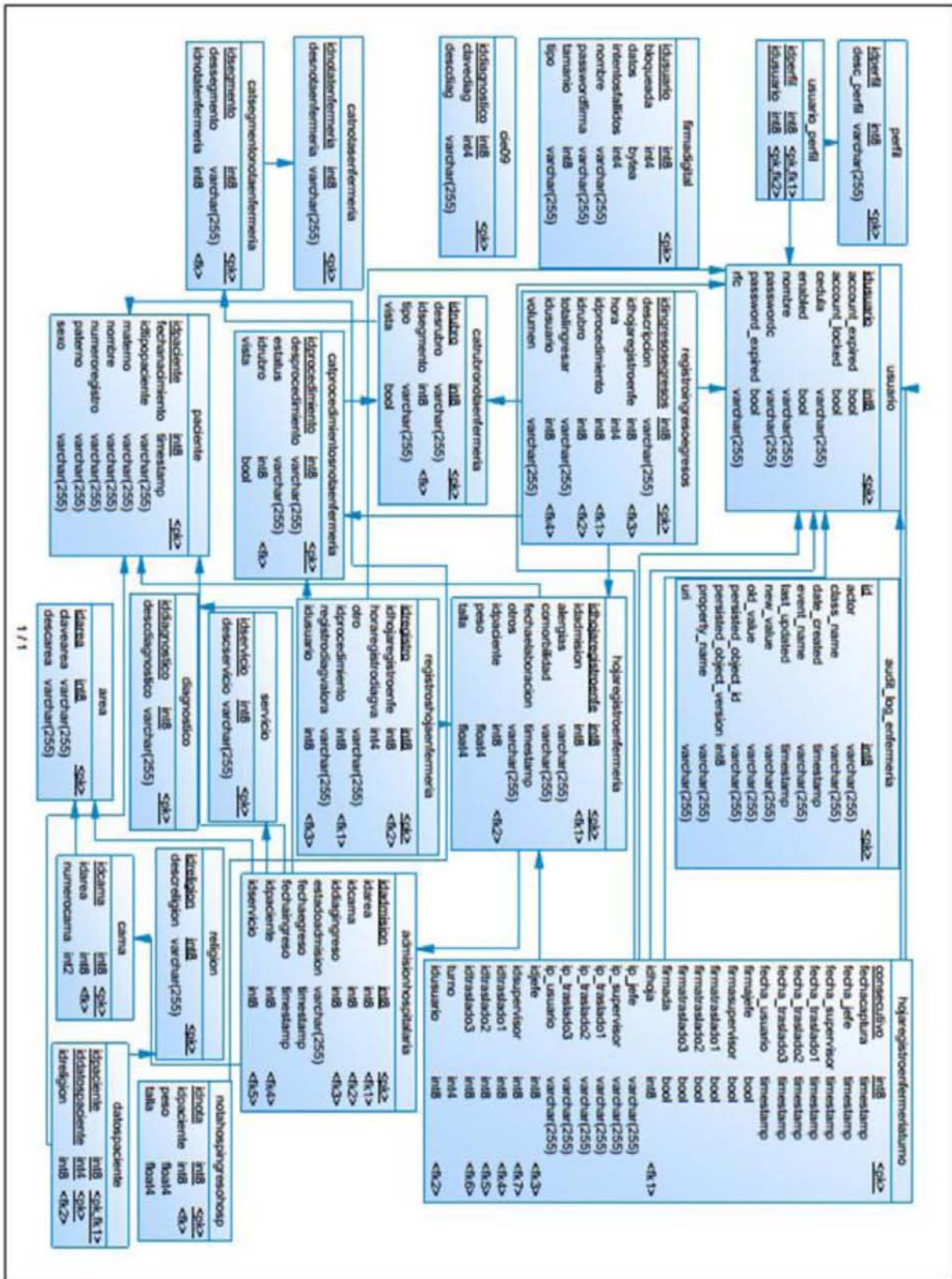


Figura 4.1.2.1. Diagrama Entidad-Relación



4.1.5 Diagrama de clase.

Es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.

El diagrama se utiliza para que la atención se centre en los aspectos lógicos de las clases en lugar de en su implementación.

Un diagrama de clases está compuesto por los siguientes elementos:

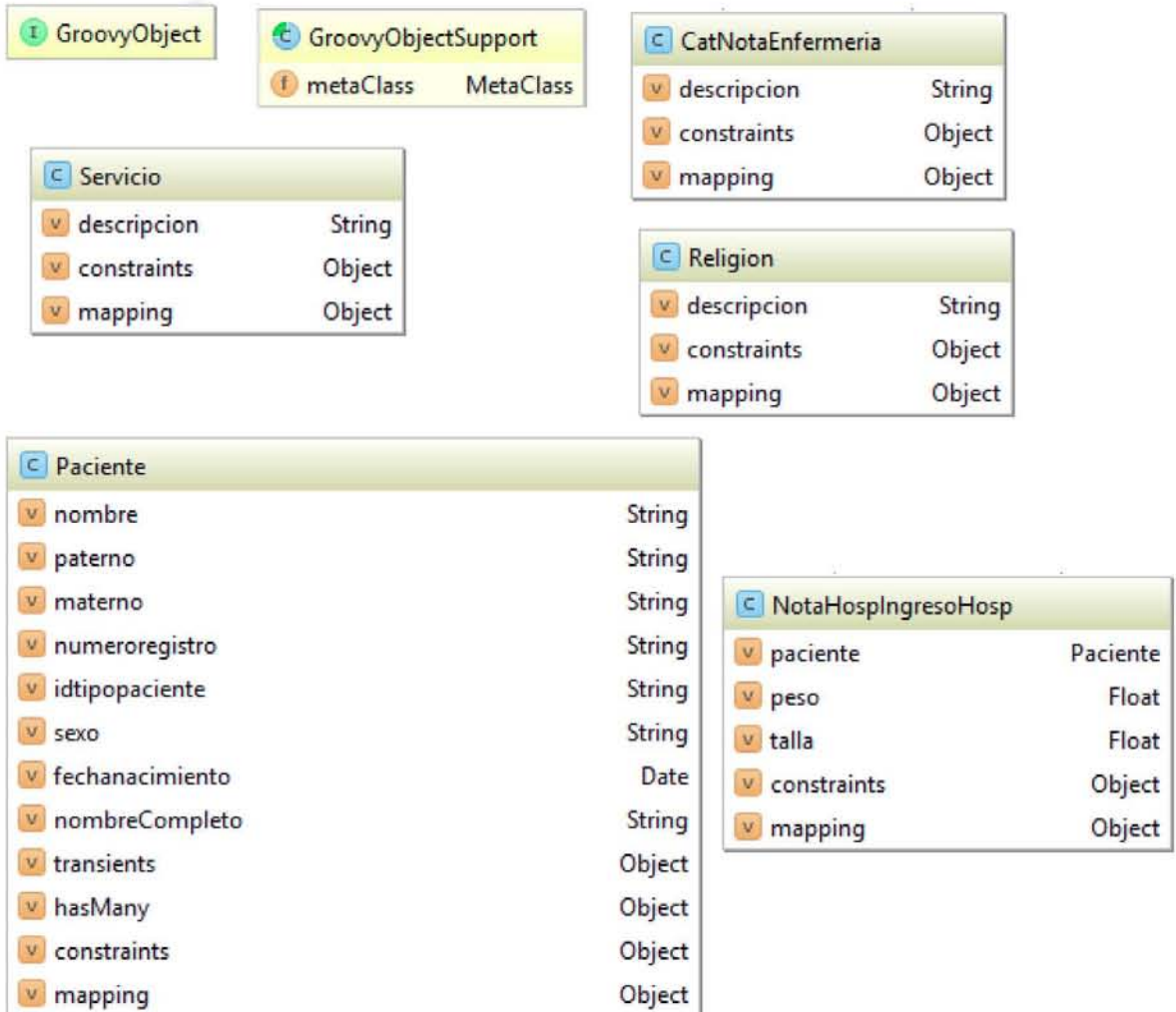
- Clase: atributos, métodos y visibilidad.
- Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

Los diagramas de clases pueden utilizarse para una gran variedad de propósitos:

- Para proporcionar una descripción de los tipos que se utilizan en un sistema y se pasan entre sus componentes que no tenga nada que ver con su implementación.
- Para clarificar el glosario de términos que se utiliza en la comunicación entre la aplicación y los usuarios y en las descripciones de las necesidades de los usuarios.



Capítulo 4. Diseño y construcción del sistema





Capítulo 4. Diseño y construcción del sistema



C Diagnostico	
v descripcion	String
v constraints	Object
v mapping	Object

C DatosPaciente	
v paciente	Paciente
v iddatospaciente	Integer
v religion	Religion
v constraints	Object
v mapping	Object

C Cie09	
v clavediag	Integer
v descdiag	String
v mapping	Object
v constraints	Object

C Cama	
v numerocama	Short
v area	Area
v constraints	Object
v mapping	Object

C Area	
v descripcion	String
v clavearea	String
v constraints	Object
v mapping	Object

C UsuarioPerfil	
v usuario	Usuario
v perfil	Perfil
v mapping	Object

C AdmisionHospitalaria	
v paciente	Paciente
v servicio	Servicio
v area	Area
v fechaIngreso	Date
v fechaEgreso	Date
v cama	Cama
v estadoadmission	String
v diagnosticoIngreso	Diagnostico
v constraints	Object
v mapping	Object
v transients	Object



Capítulo 4. Diseño y construcción del sistema



C Usuario	
v id	Long
v springSecurityService	Object
v username	String
v password	String
v cedula	String
v enabled	boolean
v accountExpired	boolean
v accountLocked	boolean
v passwordExpired	boolean
v nombre	String
v constraints	Object
v mapping	Object

C FirmaDigital	
v passwordfirma	String
v nombre	String
v tipo	String
v tamaño	Long
v datos	byte[]
v bloqueada	Integer
v intentosfallidos	Integer
v mapping	Object
v constraints	Object

C RegistroIngresoEgreso	
v hoja	HojaRegistroEnfermeria
v usuario	Usuario
v hora	Integer
v descripcion	String
v volumen	String
v totalingresar	String
v rubro	CatRubroNotaEnfermeria
v procedimiento	CatProcedimientoNotaEnfermeria
v constraints	Object
v mapping	Object



Capítulo 4. Diseño y construcción del sistema



HojaRegistroEnfermeriaTurno		
springSecurityService		Object
auditable		Object
hoja	HojaRegistroEnfermeria	
usuario		Usuario
turno		Turno
firmada		Boolean
fechaCaptura		Date
firmaJefe		Boolean
firmaSupervisor		Boolean
jefe		Usuario
supervisor		Usuario
firmaTraslado1		Boolean
firmaTraslado2		Boolean
firmaTraslado3		Boolean
traslado1		Usuario
traslado2		Usuario
traslado3		Usuario
ipUsuario		String
ipTraslado1		String
ipTraslado2		String
ipTraslado3		String
ipJefe		String
ipSupervisor		String
fechaUsuario		Date
fechaTraslado1		Date
fechaTraslado2		Date
fechaTraslado3		Date
fechaJefe		Date
fechaSupervisor		Date
constraints		Object
mapping		Object

RegistroHojaEnfermeria		
horaregistrodiagva		Integer
registrodiagvalora		String
hoja	HojaRegistroEnfermeria	
procedimiento	CatProcedimientoNotaEnfermeria	
usuario		Usuario
otro		String
constraints		Object
mapping		Object



Capítulo 4. Diseño y construcción del sistema



C HojaRegistroEnfermeria		
✓	springSecurityService	Object
✓	auditable	Object
✓	id	Long
✓	admision	AdmisionHospitalaria
✓	paciente	Paciente
✓	fechaElaboracion	Date
✓	peso	Float
✓	talla	Float
✓	alergias	String
✓	comorbilidad	String
✓	otros	String
✓	turnoActual	String
✓	turnoMatutino	HojaRegistroEnfermeriaTurno
✓	turnoVespertino	HojaRegistroEnfermeriaTurno
✓	turnoNocturno	HojaRegistroEnfermeriaTurno
✓	hasMany	Object
✓	mapping	Object
✓	has	Boolean
✓	dm	Boolean
✓	nef	Boolean
✓	ic	Boolean
✓	ir	Boolean
✓	signosVitales	List<SignoVital>
✓	dietas	List<RegistroHojaEnfermeria>
✓	rubrosValoracion	List<CatRubroNotaEnfermeria>
✓	rubrosDiagnostico	List<CatRubroNotaEnfermeria>
✓	rubrosIndicador	List<CatRubroNotaEnfermeria>
✓	requisitos	List<RegistroHojaEnfermeria>
✓	ingresos	List<Liquido>
✓	egresos	List<Liquido>
✓	medicamentos	List<Liquido>
✓	escalaOtros	List<Liquido>
✓	indicadores	List<IndicadorCalidad>
✓	escalaMadox	List<RegistroHojaEnfermeria>
✓	diagEnfermeriaObservaciones	List<RegistroHojaEnfermeria>
✓	tablaPrevencion	List<CatProcedimientoNotaEnfermeria>
✓	imageDir	String
✓	transients	Object



Capítulo 4. Diseño y construcción del sistema



C CatSegmentoNotaEnfermeria	
v descripcion	String
v padre	CatNotaEnfermeria
v constraints	Object
v mapping	Object

C CatRubroNotaEnfermeria	
v id	Long
v descripcion	String
v padre	CatSegmentoNotaEnfermeria
v vista	Boolean
v tipo	String
v constraints	Object
v mapping	Object

C CatProcedimientoNotaEnfermeria	
v id	Long
v descripcion	String
v padre	CatRubroNotaEnfermeria
v estatus	String
v vista	Boolean
v constraints	Object
v mapping	Object

4.1.6 Diccionario de Datos.

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Estos diccionarios se desarrollan durante el análisis de flujo de datos y ayuda a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño del proyecto.

Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.



Capítulo 4. Diseño y construcción del sistema

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

Ver figuras 4.1.5.1, 4.1.5.2, 4.1.5.3, 4.1.5.4, 4.1.5.5, 4.1.5.6, 4.1.5.7, 4.1.5.8, 4.1.5.9, 4.1.5.10, 4.1.5.11, 4.1.5.12 y 4.1.5.13.

Name	Code	HojaRegistroEnfermeriaTurno					Descripcion
		Data Type	Length	P	F	M	
consecutivo	consecutivo	int8	8	X		X	Consecutivo
fechacaptura	fechacaptura	timestamp				X	Fecha Captura
fecha_jefe	fecha_jefe	timestamp					Fecha Firma Jefe
fecha_supervisor	fecha_supervisor	timestamp					Fecha Firma Supervisor
fecha_traslado1	fecha_traslado1	timestamp					Fecha Traslado1
fecha_traslado2	fecha_traslado2	timestamp					Fecha Traslado2
fecha_traslado3	fecha_traslado3	timestamp					Fecha Traslado3
fecha_usuario	fecha_usuario	timestamp				X	Fecha Usuario
firmajefe	firmajefe	bool				X	Estatus Firma Jefe
firmasupervisor	firmasupervisor	bool				X	Estatus Firma Supervisor
firmatraslado1	firmatraslado1	bool				X	Estatus Firma Traslado1
firmatraslado2	firmatraslado2	bool				X	Estatus Firma Traslado2
firmatraslado3	firmatraslado3	bool				X	Estatus Firma Traslado3
firmada	firmada	bool				X	Estatus Firmada
idhoja	idhoja	int8	8		X	X	IdHoja
ip_jefe	ip_jefe	varchar(255)	255				IpMaquina Jefe
ip_supervisor	ip_supervisor	varchar(255)	255				IpMaquina Supervisor
ip_traslado1	ip_traslado1	varchar(255)	255				IpUsuario Traslado1
ip_traslado2	ip_traslado2	varchar(255)	255				IpUsuario Traslado2
ip_traslado3	ip_traslado3	varchar(255)	255				IpUsuario Traslado3
ip_usuario	ip_usuario	varchar(255)	255				IpUsuario
idjefe	idjefe	int8	8		X		IdJefe
idsupervisor	idsupervisor	int8	8		X		IdSupervisor
idtraslado1	idtraslado1	int8	8		X		IdTraslado1
idtraslado2	idtraslado2	int8	8		X		IdTraslado2
idtraslado3	idtraslado3	int8	8		X		IdTraslado3
tumo	tumo	int4	4			X	Turno
idusuario	idusuario	int8	8		X	X	IdUsuario

Figura 4.1.5.1. Diccionario de Datos Tabla HojaRegistroEnfermeriaTurno



Capítulo 4. Diseño y construcción del sistema

Name	Code	HojaRegistroEnfermeria					Descripcion
		Data Type	Length	P	F	M	
idhojaregistroenfe	idhojaregistroenfe	int8	8	X		X	Id Hoja
idadmision	idadmision	int8	8		X	X	Id Admision
alergias	alergias	varchar(255)	255			X	Alergias
comorbilidad	comorbilidad	varchar(255)	255			X	Comorbilidad
fechaelaboracion	fechaelaboracion	timestamp				X	Fecha Elaboración
otros	otros	varchar(255)	255			X	Otros
idpaciente	idpaciente	int8	8		X	X	Id Paciente
peso	peso	float4	4			X	Peso
talla	talla	float4	4			X	Talla

Figura 4.1.5.2. Diccionario de Datos Tabla HojaRegistroEnfermeria

Name	Code	RegistrosHojaEnfermeria					Descripcion
		Data Type	Length	P	F	M	
idregistro	idregistro	int8	8	X		X	Consecutivo
idhojaregistroenfe	idhojaregistroenfe	int8	8		X	X	Expediente asociado
horaregistrodiagva	horaregistrodiagva	int4	4				Hora
otro	otro	varchar(255)	255			X	Otros Datos
idprocedimiento	idprocedimiento	int8	8		X	X	Catalogo Procedimiento
registrodiagvalora	registrodiagvalora	varchar(255)	255				Otro Registro
idusuario	idusuario	int8	8		X	X	Id Usuario

Figura 4.1.5.3. Diccionario de Datos Tabla RegistroHojaEnfermeria

Name	Code	AdmisionHospitalaria					Descripcion
		Data Type	Length	P	F	M	
idadmision	idadmision	int8	8	X		X	Id Admision
idarea	idarea	int8	8		X	X	Id Area
idcama	idcama	int8	8		X	X	Id Cama
iddiagingresso	iddiagingresso	int8	8		X	X	Id Día Ingreso
estadoadmision	estadoadmision	varchar(255)	255			X	Estatus de la admision
fechaegreso	fechaegreso	timestamp				X	Fecha de Egreso
fechaingreso	fechaingreso	timestamp				X	Fecha de Ingreso
idpaciente	idpaciente	int8	8		X	X	Id Paciente
idservicio	idservicio	int8	8		X	X	Id Servicio

Figura 4.1.5.4. Diccionario de Datos Tabla AdmisionHospitalaria



Capítulo 4. Diseño y construcción del sistema

Name	Code	RegistroIngresoEgresos					Descripcion
		Data Type	Length	P	F	M	
idingrosoegresos	idingrosoegresos	int8	8	X		X	Consecutivo
descripcion	descripcion	varchar(255)	255				Descripcion
idhojaregistroenfe	idhojaregistroenfe	int8	8		X	X	Expediente Asociado
hora	hora	int4	4				Hora
idprocedimiento	idprocedimiento	int8	8		X		Id Procedimiento
idrubro	idrubro	int8	8		X		Id Rubro
totalingresar	totalingresar	varchar(255)	255				Total
idusuario	idusuario	int8	8		X	X	Id Usuario
volumen	volumen	varchar(255)	255				Volumen

Figura 4.1.5.5. Diccionario de Datos Tabla RegistroIngresoEgresos

Name	Code	Paciente					Descripcion
		Data Type	Length	P	F	M	
idpaciente	idpaciente	int8	8	X		X	Id Paciente
fechanacimiento	fechanacimiento	timestamp				X	Fecha Nacimiento
idtipopaciente	idtipopaciente	varchar(255)	255			X	Id Tipo Paciente
materno	materno	varchar(255)	255			X	Apellido Materno
nombre	nombre	varchar(255)	255			X	Nombre
numeroregistro	numeroregistro	varchar(255)	255			X	Numero Registro
paterno	paterno	varchar(255)	255			X	Apellido Paterno
sexo	sexo	varchar(255)	255			X	Sexo

Figura 4.1.5.6. Diccionario de Datos Tabla Paciente

Name	Code	Usuario					Descripcion
		Data Type	Length	P	F	M	
idusuario	idusuario	int8	8	X		X	Id Usuario
account_expired	account_expired	bool				X	Estatus Expiracion
account_locked	account_locked	bool				X	Estatus Bloqueado
cedula	cedula	varchar(255)	255			X	Cedula
enabled	enabled	bool				X	Estatus Habilitado
nombre	nombre	varchar(255)	255			X	Nombre Usuario
passwordc	passwordc	varchar(255)	255			X	Password
password_expired	password_expired	bool				X	Estatus expiracion
rfc	rfc	varchar(255)	255			X	RFC

Figura 4.1.5.7. Diccionario de Datos Tabla Usuario



Capítulo 4. Diseño y construcción del sistema



		FirmaDigital					
Name	Code	Data Type	Length	P	F	M	Descripcion
idusuario	idusuario	int8	8	X		X	Id Usuario
bloqueada	bloqueada	int4	4			X	Estatus bloqueada
datos	datos	bytea					Datos Adicionales
intentosfallidos	intentosfallidos	int4	4			X	Intentos Fallidos
nombre	nombre	varchar(255)	255			X	Nombre
passwordfima	passwordfima	varchar(255)	255			X	Password
tamano	tamano	int8	8			X	Tamaño
tipo	tipo	varchar(255)	255			X	Tipo

Figura 4.1.5.8. Diccionario de Datos Tabla FirmaDigital

		Perfil					
Name	Code	Data Type	Length	F	M	Descripcion	
idperfil	idperfil	int8	8		X	Id Perfil	
desc_perfil	desc_perfil	varchar(255)	255		X	Descripcion	

Figura 4.1.5.9. Diccionario de Datos Tabla Perfil

		Usuario_Perfil					
Name	Code	Data Type	Length	P	F	M	Descripcion
idperfil	idperfil	int8	8	X	X	X	Id Perfil
idusuario	idusuario	int8	8	X	X	X	Id Usuario

Figura 4.1.5.10. Diccionario de Datos Tabla Usuario_Perfil



Capítulo 4. Diseño y construcción del sistema



Cie09							
Name	Code	Data Type	Length	P	F	M	Descripcion
iddiagnostico	iddiagnostico	int8	8	X		X	Id Diagnostico
clavediag	clavediag	int4	4			X	Clave
descdiag	descdiag	varchar(255)	255			X	Descripcion

Figura 4.1.5.11. Diccionario de Datos Tabla Cie09

CatNotasEnfermeria							
Name	Code	Data Type	Length	P	F	M	Descripcion
idnotatenfermeria	idnotatenfermeria	int8	8	X		X	Id Nota
desnotaenfermeria	desnotaenfermeria	varchar(255)	255			X	Descripcion

Figura

4.1.5.12. Diccionario de Datos Tabla CatNotasEnfermeria

CatRubrosEnfermeria							
Name	Code	Data Type	Length	P	F	M	Descripcion
idrubro	idrubro	int8	8	X		X	Id Rubro
desrubro	desrubro	varchar(255)	255			X	Descripcion
idsegmento	idsegmento	int8	8		X	X	Id Segmento
tipo	tipo	varchar(255)	255			X	Tipo
vista	vista	bool				X	Vista

Figura 4.1.5.13. Diccionario de Datos Tabla CatRubrosEnfermeria



Capítulo 4. Diseño y construcción del sistema



4.2 Implementación de la base de datos.

Instalación de la base de datos PostgreSQL

Para el proyecto de la hoja de registro clínico de enfermería se instaló el gestor de base de datos PostgreSQL 9.4.4-3 en un sistema operativo Windows 7 de 64 bits.

El programa de instalación fue descargado desde la página de PostgreSQL <http://www.enterprisedb.com/products-services-training/pgdownload>, donde se seleccionó el tipo de sistema operativo para el cual se requería la instalación.

En la figura 4.2.1 se muestra la página donde se descargó el archivo de instalación.

The screenshot shows the 'Download PostgreSQL' page on the EnterpriseDB website. The page title is 'Download PostgreSQL'. Below the title, there is a 'Please Note' section stating 'Cookies should be enabled for the download process to function correctly'. The main content area displays three installer versions with their respective download links for different operating systems:

- Installer version Version 9.4.4** [Readme file for customers interested in using PL/Perl, PL/Python or PL/Tcl]
 - Linux x86-32
 - Linux x86-64
 - Win x86-32
 - Win x86-64
 - Mac OS X
- Installer version Version 9.3.9**
 - Linux x86-32
 - Linux x86-64
 - Win x86-32
 - Win x86-64
 - Mac OS X
- Installer version Version 9.2.13**
 - Linux x86-32
 - Linux x86-64
 - Win x86-32
 - Win x86-64
 - Mac OS X

The left sidebar contains navigation links for Subscriptions, PostgreSQL Overview, Download PostgreSQL, PostgreSQL Plus Cloud Database, EnterpriseDB Tools, PostgreSQL Enterprise Manager, xDB Replication - Multi Master, and xDB Replication - Single Master.

Figura 4.2.1 Página de descarga de PostgreSQL

Una vez descargado el archivo se procede a la instalación del gestor de base de datos.



Capítulo 4. Diseño y construcción del sistema



En la figura 4.2.2 se muestra la pantalla inicial de instalación de PostgreSQL.



Figura 4.2.2 Pantalla inicial de la instalación de PostgreSQL

El programa de instalación solicita que se proporcione una contraseña de administrador, como se muestra en la figura 4.2.3

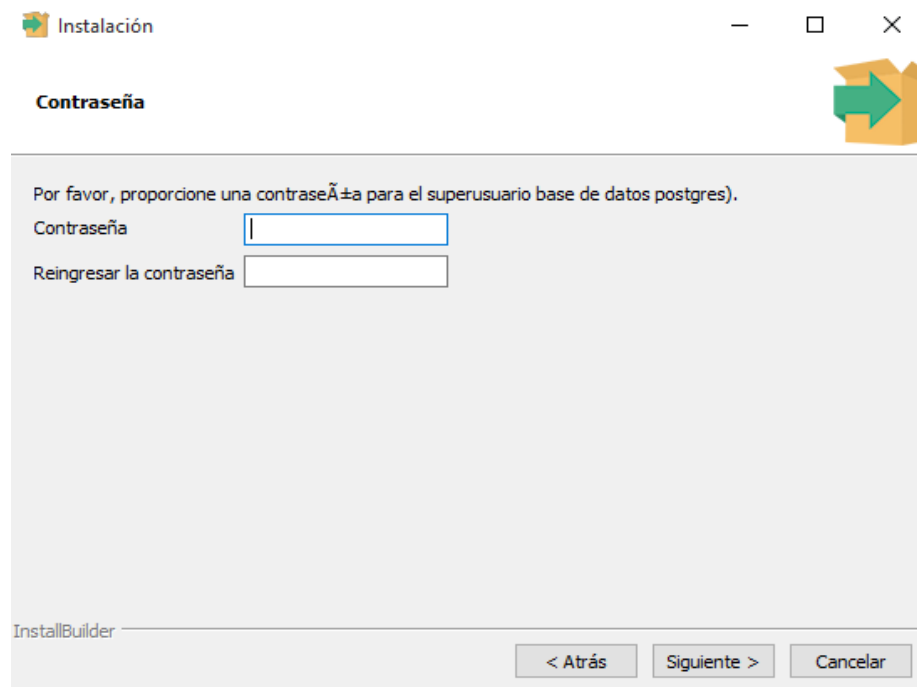


Figura 4.2.3 Instalación de PostgreSQL



Capítulo 4. Diseño y construcción del sistema



En la figura 4.2.4 se muestra la pantalla final de la instalación de PostgreSQL.



Figura 4.2.4 Instalación finalizada de PostgreSQL

El paquete de instalación incluye la herramienta PgAdmin III, que nos permitirá gestionar la Base de Datos de una manera sencilla y con una interfaz de usuario amigable, en la figura 4.2.5 se muestra la pantalla inicial de PgAdmin III

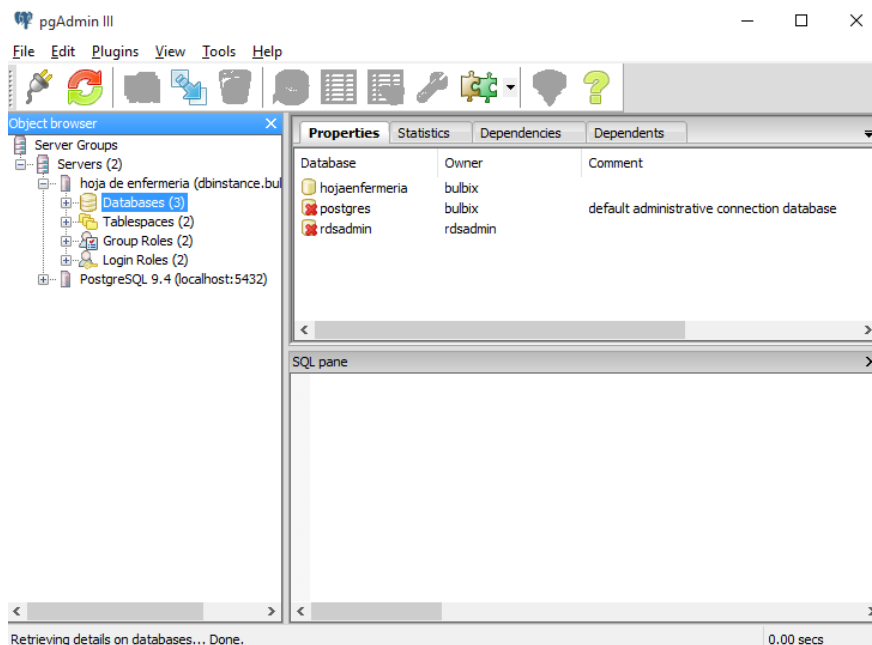


Figura 4.2.5 Herramienta PgAdmin III



Capítulo 4. Diseño y construcción del sistema



Creación de la base de datos

Para administrar la base de datos usaremos PgAdmin III, para ello, debemos situarnos en el Servidor PostgreSQL 9.4, y seleccionar con el botón derecho del mouse la opción conectar, nos pedirá nuestra contraseña de administrador PostgreSQL.

Para crear la base de datos, nos desplazamos por el navegador de objetos, hasta la opción **Databases** y con el botón derecho del mouse seleccionaremos la opción **New Database**. Como se muestra en la figura 4.2.6

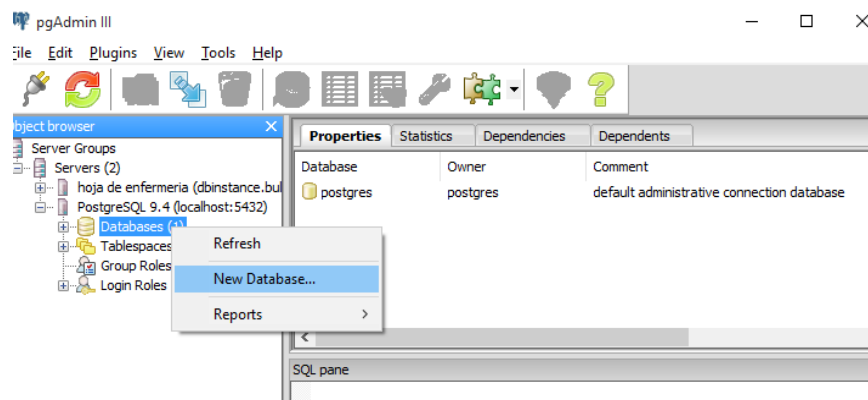


Figura 4.2.6 Creación de la base de datos en PgAdmin III

En el campo de **Name**, ingresamos el título hojaenfermeria como se muestra en la figura 4.2.7, y aceptaremos con el botón **OK**.



Capítulo 4. Diseño y construcción del sistema

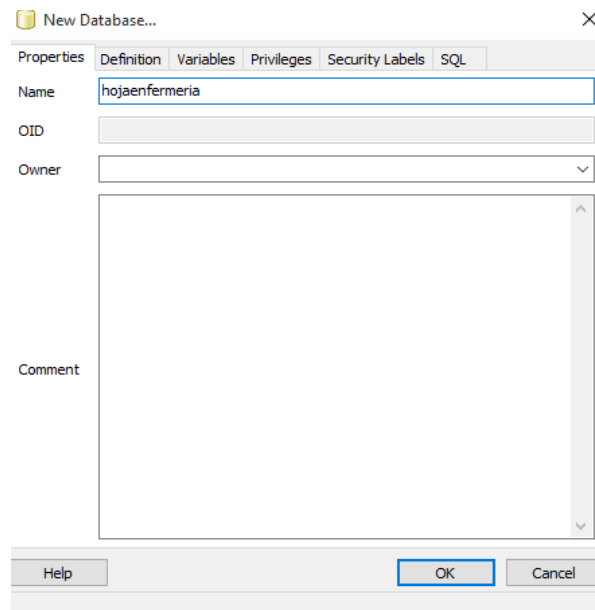


Figura 4.2.7 Creación de la base de datos con PgAdmin III

Para el sistema la base de datos creada fue *hojaenfermeria*, el procedimiento anterior corresponde a la creación de la base de datos mediante el siguiente script:

```
CREATE DATABASE hojaenfermeria;
```

Creación de una tabla

Para crear una tabla en la herramienta PgAdmin III, seleccionaremos la base de datos *hojaenfermeria*, desplegamos el menú de la base de datos y nos desplazamos hasta la opción **Schemas** -> **public**, daremos un clic con el botón derecho del mouse y seleccionaremos la opción **New Object** -> **New Table** como se muestra en la figura 4.2.8.



Capítulo 4. Diseño y construcción del sistema

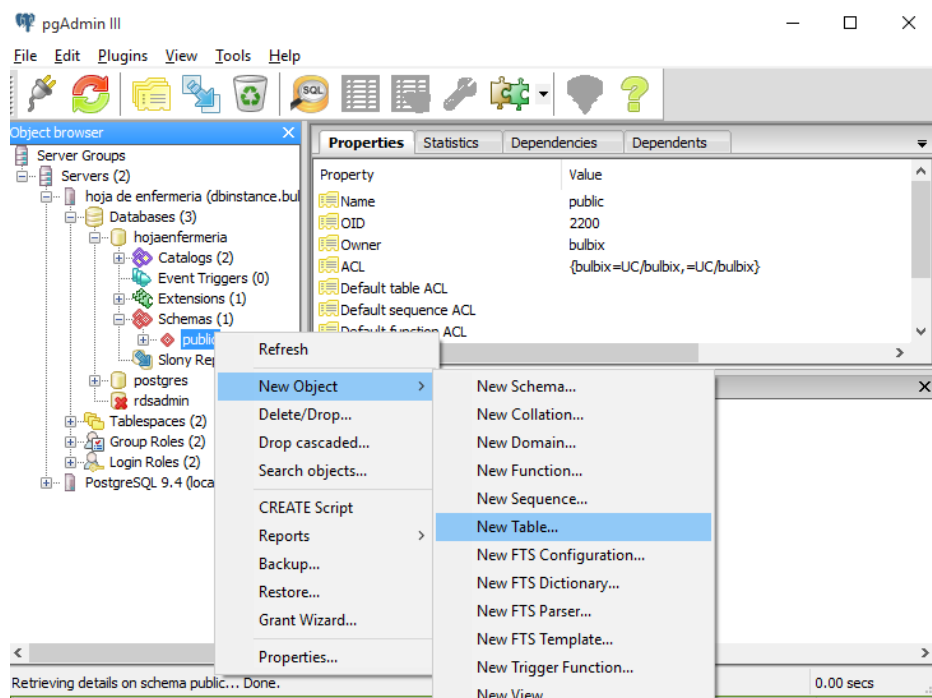


Figura 4.2.8 Creación de una tabla con PgAdmin III

Se abrirá una nueva ventana en la cual se podrán ingresar todas las características de la tabla, en la pestaña **Properties** en el campo **Name**, se podrá ingresar el nombre de la tabla, en la pestaña **Columns** se agregan los atributos, indicando el tipo de dato y longitud máxima, en la pestaña **Constraints**, se podrá seleccionar cuál de los campos anteriores funcionará como llave primaria, como se muestra en la figura 4.2.9.



Capítulo 4. Diseño y construcción del sistema

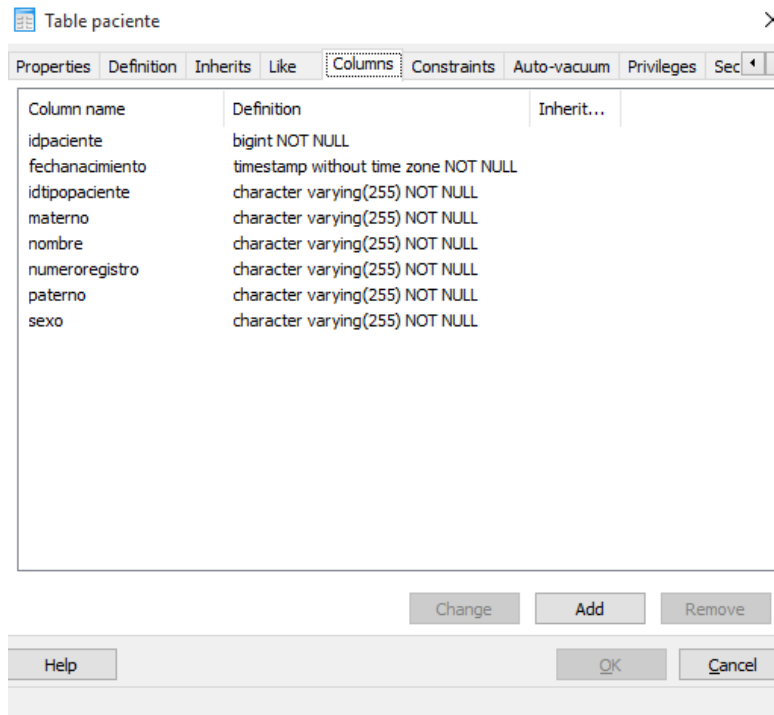


Figura 4.2.9 Atributos de una tabla con PgAdmin III

Para este ejemplo se creará una tabla de nombre paciente, con los atributos mostrados en la tabla 4.2.1

Tabla paciente				
Nombre del campo	Tipo de dato	Longitud	Valor nulo	Campo llave
idpaciente	Entero	255	No	Llave primaria
fechanacimiento	Fecha	No aplica	No	No
idtipopaciente	Cadena de caracteres	255	No	No
materno	Cadena de caracteres	255	No	No
nombre	Cadena de caracteres	255	No	No
numeroregistro	Cadena de caracteres	255	No	No
paterno	Cadena de caracteres	255	No	No
sexo	Cadena de caracteres	255	No	No

Tabla 4.2.1 Atributos de la tabla paciente



Capítulo 4. Diseño y construcción del sistema



El procedimiento anterior para crear una tabla, corresponde a la ejecución del siguiente script:

```
CREATE TABLE paciente
(idpaciente bigint NOT NULL,
fechanacimiento timestamp without time zone NOT NULL,
idtipopaciente character varying(255) NOT NULL,
materno character varying(255) NOT NULL,
nombre character varying(255) NOT NULL,
numeroregistro character varying(255) NOT NULL,
aterno character varying(255) NOT NULL,
sexo character varying(255) NOT NULL,
CONSTRAINT paciente_pkey PRIMARY KEY (idpaciente) );
```

Árbol de tablas

A través de la herramienta PgAdmin III podremos visualizar el árbol de tablas, como se muestra en la figura 4.2.10



Capítulo 4. Diseño y construcción del sistema

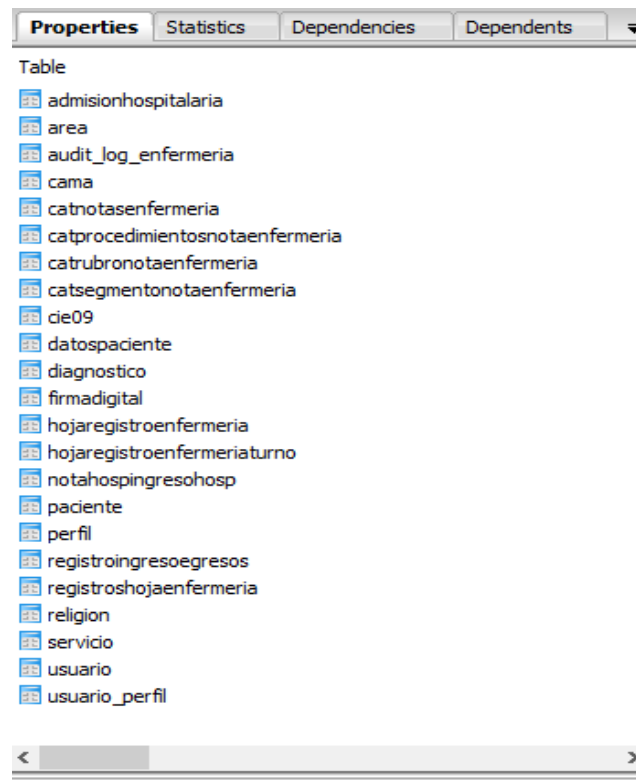


Figura 4.2.10 Árbol de tablas

Consultas en la base de datos

Una consulta es una operación a través de la cual se puede visualizar la información contenida en los registros de una base de datos, esta información nos sirve para poder actualizar, borrar, insertar o realizar operaciones más complejas en la base de datos

Un script para realizar una consulta simple en PostgreSQL es el siguiente:

```
SELECT atributos  
FROM tabla
```

Para buscar los campos **apellido paterno, nombre, fecha de nacimiento y sexo** de la tabla **paciente**, se utiliza el siguiente script, el resultado de esta consulta se visualiza en la imagen 4.2.11



Capítulo 4. Diseño y construcción del sistema



```
SELECT idpaciente, paterno, nombre, fechanacimiento, sexo
FROM paciente;
```

	idpaciente bigint	paterno character varying(255)	nombre character varying(255)	fechanacimiento timestamp without time zone	sexo character varying(255)
1	1	Prado	Luis	2000-10-15 00:00:00	H
2	4	Flores	Daniel	1995-01-05 00:00:00	H
3	5	Rios	Rocio	2001-11-20 00:00:00	M
4	6	Carrillo	Maria	1990-04-13 00:00:00	M
5	8	Bustamante	Marcela	1995-02-21 00:00:00	M
6	2	Juarez	Arturo	1986-12-03 00:00:00	H
7	7	Barrera	Manuel	1980-01-10 00:00:00	H
8	3	Alvarez	Carlos	1992-09-15 00:00:00	H
9	9	Estrada	Carmen	1999-03-30 00:00:00	M
10	10	Beltran	Ignacio	1996-05-11 00:00:00	H
11	11	Bravo	Alexa	1986-11-15 00:00:00	M
12	12	Gonzalez	Brenda	1994-12-15 00:00:00	M
13	13	Maldonado	Ximena	2000-02-15 00:00:00	M
14	14	Bolaños	Diana	1991-03-15 00:00:00	M
15	15	Perez	Paola	1990-10-15 00:00:00	M

Figura 4.2.11 Consulta tabla paciente en PostgreSQL

Para realizar una consulta de **apellido paterno, nombre, sexo, número de cama, y fecha de ingreso** de las tablas **paciente, admisionhospitalaria** y **cama**, se utiliza el siguiente script, el resultado de la consulta se puede apreciar en la figura 4.2.12

```
SELECT
paciente.idpaciente,
paciente.paterno,
paciente.nombre,
paciente.sexo,
cama.numerocama,
admisionhospitalaria.fechaingreso

FROM
public.paciente,
```



Capítulo 4. Diseño y construcción del sistema



public.admisionhospitalaria,
public.cama

WHERE

paciente.idpaciente= admisionhospitalaria.idpaciente AND
admisionhospitalaria.idcama=cama.idcama

Output pane

	idpaciente bigint	paterno character varying(255)	nombre character varying(255)	sexo character varying(255)	numerocama smallint	fechaingreso timestamp without time zone
1	1	Prado	Luis	H		2015-06-01 00:00:00
2	2	Juarez	Arturo	H		2015-04-01 00:00:00
3	3	Alvarez	Carlos	H		2015-04-01 00:00:00
4	4	Flores	Daniel	H		2015-04-01 00:00:00
5	5	Rios	Rocio	M		2015-04-01 00:00:00
6	6	Carrillo	Maria	M		2015-04-01 00:00:00
7	7	Barrera	Manuel	H		2015-04-01 00:00:00
8	8	Bustamante	Marcela	M		2015-04-01 00:00:00
9	11	Bravo	Alexa	M		2014-06-01 00:00:00
10	12	Gonzalez	Brenda	M		2014-06-01 00:00:00
11	13	Maldonado	Ximena	M		2014-06-01 00:00:00
12	14	Bolaños	Diana	M		2014-06-01 00:00:00
13	15	Perez	Paola	M		2014-06-01 00:00:00
14	16	Guerra	Frida	M		2014-06-01 00:00:00
15	17	Navarrete	Itzel	M		2015-01-01 00:00:00
16	18	Velazquez	Viviana	M		2015-01-01 00:00:00
17	19	Romero	Citlalli	M		2015-01-01 00:00:00
18	20	Mora	Maria Jose	M		2015-01-01 00:00:00
19	21	Rosales	Marco	H		2015-01-01 00:00:00

Figura 4.2.12

Consulta de varias tablas en PostgreSQL

En la siguiente consulta se obtendrá el **apellido paterno, nombre, sexo y cama** de los pacientes ingresados en el **piso 2** cuya religión es **crisiana**, para ello se utiliza el siguiente script, el resultado de la consulta se puede apreciar en la figura 4.2.13.

SELECT

paciente.idpaciente,
paciente.paterno,
paciente.nombre,



Capítulo 4. Diseño y construcción del sistema



paciente.sexo,
 cama.numerocama,
 area.clavearea,
 religion.descreligion

FROM

public.paciente,
 public.cama,
 public.area,
 public.datospaciente,
 public.religion,
 public.admisionhospitalaria

WHERE

paciente.idpaciente = datospaciente.idpaciente AND
 paciente.idpaciente = admisionhospitalaria.idpaciente AND
 cama.idcama = admisionhospitalaria.idcama AND
 area.idarea = admisionhospitalaria.idarea AND
 datospaciente.idreligion = religion.idreligion AND
 religion.descreligion='Cristiana' AND
 admisionhospitalaria.idarea=2

ORDER by paciente.idpaciente;

Output pane

	idpaciente bigint	paterno character varying(255)	nombre character varying(255)	sexo character varying(255)	numerocama smallint	descreligion character varying(255)	clavearea character varying(255)
1	3	Alvarez	Carlos	H	3	Cristiana	002
2	16	Guerra	Frida	M	1	Cristiana	002
3	17	Navarrete	Itzel	M	2	Cristiana	002
4	20	Mora	Maria Jose	M	5	Cristiana	002
5	23	Cruz	Cristian	H	8	Cristiana	002

Figura 4.2.13 Consulta compleja en PostgreSQL



Capítulo 4. Diseño y construcción del sistema



En la siguiente consulta se obtendrá el **apellido paterno, nombre, sexo, religión y fecha de ingreso**, de los pacientes que fueron ingresados del **01-01-2014** al **31-12-2014**. Para ello se utilizará el siguiente script, el resultado se puede apreciar en la figura 4.2.14.

SELECT

```
paciente.idpaciente,  
paciente.paterno,  
paciente.nombre,  
paciente.sexo,  
religion.descreligion,  
admisionhospitalaria.fechaingreso
```

FROM

```
public.paciente,  
public.datospaciente,  
public.admisionhospitalaria,  
public.religion
```

WHERE

```
paciente.idpaciente = datospaciente.idpaciente AND  
paciente.idpaciente = admisionhospitalaria.idpaciente AND  
religion.idreligion = datospaciente.idreligion AND  
(admisionhospitalaria.fechaingreso BETWEEN '2014-01-01' AND '2014-12-31')
```

```
ORDER by admisionhospitalaria.fechaingreso;
```



Capítulo 4. Diseño y construcción del sistema



Output pane

	idpaciente bigint	paterno character varying(255)	nombre character varying(255)	sexo character varying(255)	descreligion character varying(255)	fechaingreso timestamp without time zone
1	20	Mora	Maria Jose	M	Cristiana	2014-01-01 00:00:00
2	23	Cruz	Cristian	H	Cristiana	2014-01-01 00:00:00
3	6	Carrillo	Maria	M	Sin Religion	2014-04-01 00:00:00
4	29	Aguilar	Angel	H	Sin Religion	2014-04-01 00:00:00
5	1	Prado	Luis	H	Catolica	2014-06-01 00:00:00
6	16	Guerra	Frida	M	Cristiana	2014-06-01 00:00:00
7	15	Perez	Paola	M	Sin Religion	2014-06-01 00:00:00
8	11	Bravo	Alexa	M	Cristiana	2014-06-01 00:00:00
9	12	Gonzalez	Brenda	M	Catolica	2014-06-01 00:00:00
10	14	Bolaños	Diana	M	Cristiana	2014-06-01 00:00:00

Figura 4.2.14 Consulta por fechas en PostgreSQL

Las consultas también pueden ser generadas a través de PgAdmin III de manera gráfica, para ello debemos desplazarnos por el explorador de objetos hasta cualquier tabla de la base de datos y con el botón derecho del mouse seleccionar la opción **Scripts -> SELECT Script** como se muestra en la figura 4.2.15

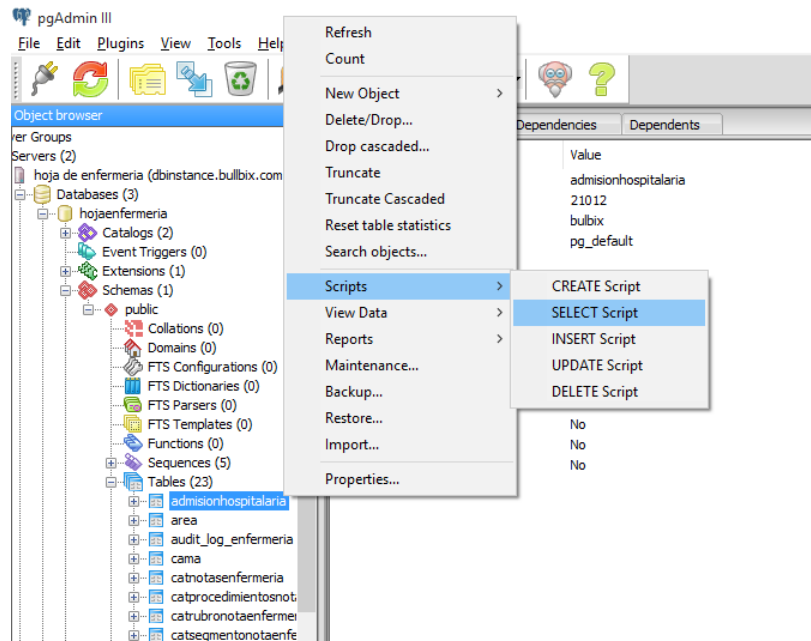


Figura 4.2.15 Creación de una consulta de manera gráfica en PgAdmin III

Se abrirá una ventana en la cual podremos seleccionar las tablas y atributos necesarios para poder realizar una consulta, como se muestra en la figura 4.2.16



Capítulo 4. Diseño y construcción del sistema

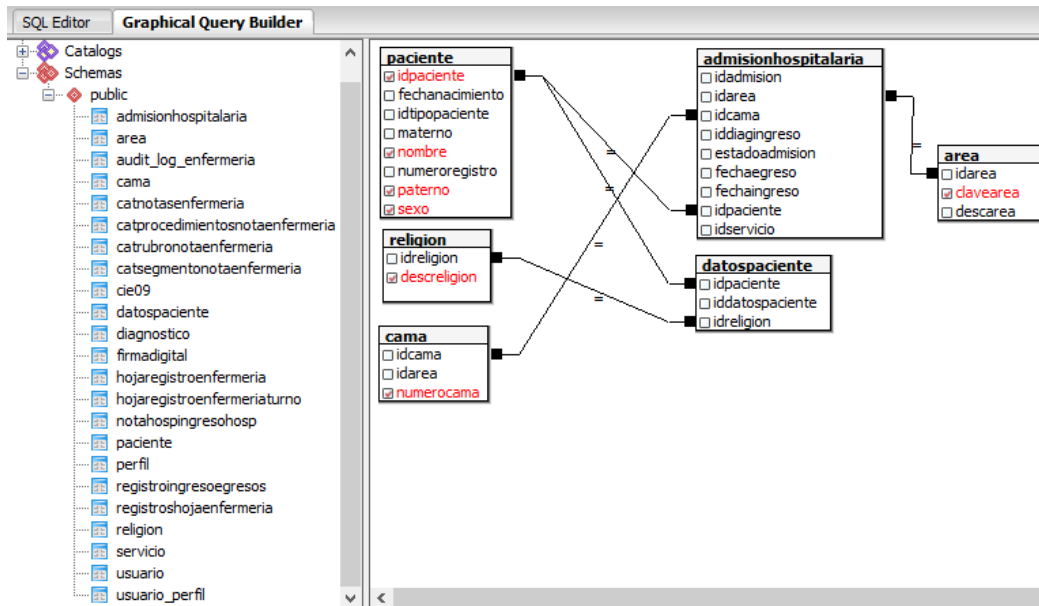


Figura 4.2.16 consulta gráfica en PgAdmin III

El resultado de la consulta anterior se muestra en la figura 4.2.17

Output pane							
Data Output							
	idpaciente bigint	nombre character varying(255)	paterno character varying(255)	sexo character varying(255)	descreligion character varying(255)	numerocama smallint	clavearea character varying(255)
1	1	Luis	Prado	H	Catolica	1	001
2	5	Rocio	Rios	M	Catolica	5	001
3	7	Manuel	Barrera	H	Catolica	7	001
4	11	Alexa	Bravo	M	Cristiana	11	001
5	13	Ximena	Maldonado	M	Cristiana	13	001
6	15	Paola	Perez	M	Sin Religion	15	001
7	16	Frida	Guerra	M	Cristiana	1	002
8	18	Viviana	Velazquez	M	Catolica	3	002
9	20	Maria Jose	Mora	M	Cristiana	5	002
10	22	Jesus	Rosas	H	Sin Religion	7	002
11	26	Rodrigo	Ibarra	H	Sin Religion	11	002
12	28	Raymundo	Ramirez	H	Catolica	13	002
13	30	Adrian	Medina	H	Sin Religion	15	002
14	6	Maria	Carrillo	M	Sin Religion	6	002

Figura 4.2.17 Resultado de la consulta gráfica en PgAdmin III



Instalación de Groovy / Grails

Para la instalación de Groovy el lenguaje de programación y Grails el web framework de desarrollo ágil, podemos ir a la página oficial de las soluciones y descargar las últimas versiones:

<http://www.groovy-lang.org/> Groovy 2.4, Ver Figura 4.2.18

★ Groovy 2.4

Groovy 2.4 is our latest official [version](#) of Groovy. Important: Releases before 2.4.4 weren't done under the Apache Software Foundation and are provided as a convenience, without any warranty.

Apache Groovy is an effort undergoing incubation at The Apache Software Foundation (ASF) sponsored by the Apache Incubator PMC. Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

2.4.4 distributions



[binary](#)



[source](#)



[documentation](#)



[SDK bundle](#)

Figura 4.2.18 Página descarga Groovy

<https://grails.org/> Grails 3.0.4 Ver Figura 4.2.19

🚀 Grails 3.0.4

Grails 3.0.4 is the latest official release of Grails

3.0.4 distributions



[binary](#)



[documentation](#)



[release notes](#)

Consult the [release notes](#) for more information. For historical release notes, refer to [Github](#)

Figura 4.2.19 Página descarga Grails



Capítulo 4. Diseño y construcción del sistema



La solución recomendada para empezar un proyecto en Groovy/Grails es descargar el Entorno de Desarrollo Integrado Groovy/Grails Tool Suite que está basado en Eclipse, por tanto la curva de aprendizaje es alta, ya que nos dará un todo en uno para el desarrollo de aplicaciones en Groovy/Grails.

<http://spring.io/tools/ggts/all>

GTS 3.6.4 Ver Figura 4.2.20



Figura 4.2.20

Página descarga GGTS

La descarga es un archivo zip con la siguiente estructura de directorios, ver Figura 4.2.21

- Una carpeta **ggts-bundle** que contiene el IDE en sus versión 3.6.3, con una versión de Groovy 2.3.10.
- Una carpeta **grails-2-4-4** con una versión de Grails la 2.4.4.
- Una carpeta **legal** con los términos y condiciones del producto.
- Una carpeta **pivotal-tc-server** con un servidor web basado en Apache Tomcat para desplegar nuestras aplicaciones.



Capítulo 4. Diseño y construcción del sistema

▼	ggts-bundle	hoy 11:24	--	Carpeta
	.DS_Store	hoy 11:24	6 KB	Documento
▼	ggts-3.6.4.RELEASE	hoy 11:25	--	Carpeta
	.DS_Store	hoy 11:25	6 KB	Documento
	.eclipseproduct	09/03/2015 22:07	192 bytes	Docu...dit.app
	artifacts.xml	09/03/2015 22:05	278 KB	Texto XML
▶	configuration	09/03/2015 22:05	--	Carpeta
▶	dropins	09/03/2015 22:05	--	Carpeta
	epl-v10.html	28/01/2015 12:08	13 KB	Texto HTML
▶	features	09/03/2015 22:05	--	Carpeta
	GGTS	09/03/2015 22:05	28 bytes	Alias
	GGTS.app	09/03/2015 22:05	94 KB	Aplicación
	license.txt	09/03/2015 21:33	11 KB	texto
▶	META-INF	09/03/2015 22:05	--	Carpeta
	notice.html	28/01/2015 12:08	9 KB	Texto HTML
	open_source_licenses.txt	09/03/2015 21:33	906 KB	texto
▶	p2	09/03/2015 22:03	--	Carpeta
▶	plugins	09/03/2015 22:05	--	Carpeta
▶	readme	09/03/2015 22:05	--	Carpeta
▶	grails-2.4.4	28/10/2014 15:35	--	Carpeta
▶	legal	09/03/2015 21:33	--	Carpeta
▶	pivotal-tc-server-developer-3.1.0.RELEASE	02/03/2015 18:18	--	Carpeta

Figura 4.2.21

Estructura de directorios de GGTS

Dando clic en el archivo GGTS.app ver Figura 4.2.21 abrirá el IDE que estará listo para iniciar un proyecto en Groovy/Grails como se muestra en la figura 4.2.22.



Figura 4.2.22

Iniciando GGTS

Instalación de Apache Tomcat

Para instalar Apache Tomcat, ingresaremos a la página oficial de descargas:

<http://tomcat.apache.org/download-80.cgi>



Capítulo 4. Diseño y construcción del sistema



Seleccionaremos la distribución que requerimos, en este caso una distribución para sistema operativo Windows de 64 bits, como se muestra en la figura 4.2.23

8.0.24

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, md5, sha1\)](#)
 - [tar.gz \(pgp, md5, sha1\)](#)
 - [32-bit Windows zip \(pgp, md5, sha1\)](#)
 - [64-bit Windows zip \(pgp, md5, sha1\)](#)
 - [64-bit Itanium Windows zip \(pgp, md5, sha1\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, md5, sha1\)](#)
- Full documentation:
 - [tar.gz \(pgp, md5, sha1\)](#)

Figura 4.2.23 Descarga de Apache Tomcat

Ejecutaremos el archivo descargado, y seguiremos las indicaciones del asistente de instalación, como se muestra en la figura 4.2.24

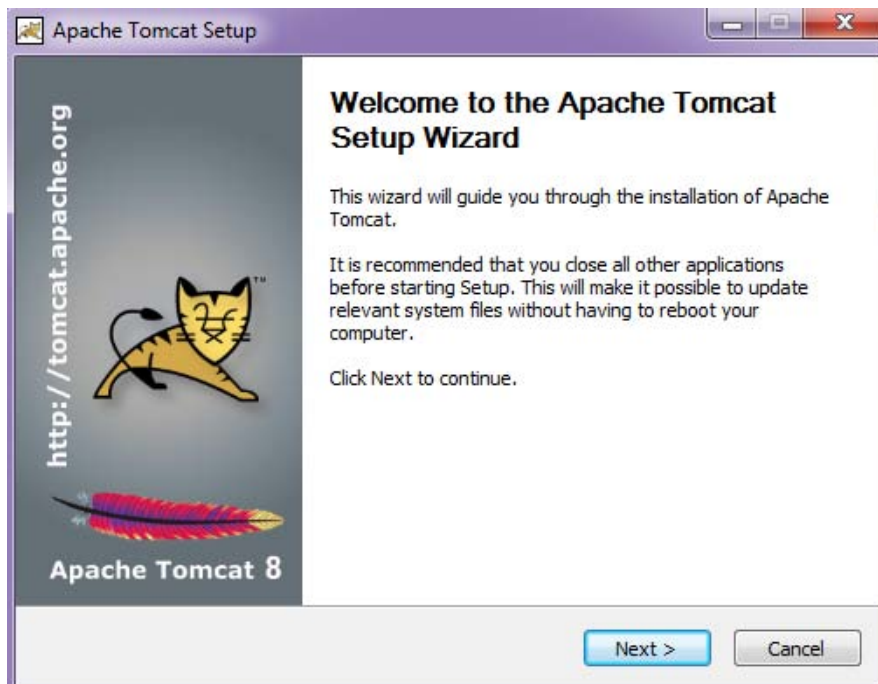


Figura 4.2.24 Instalación de Apache Tomcat

Seleccionaremos la ubicación de la máquina virtual de Java, como se muestra en la figura 4.2.25.



Capítulo 4. Diseño y construcción del sistema

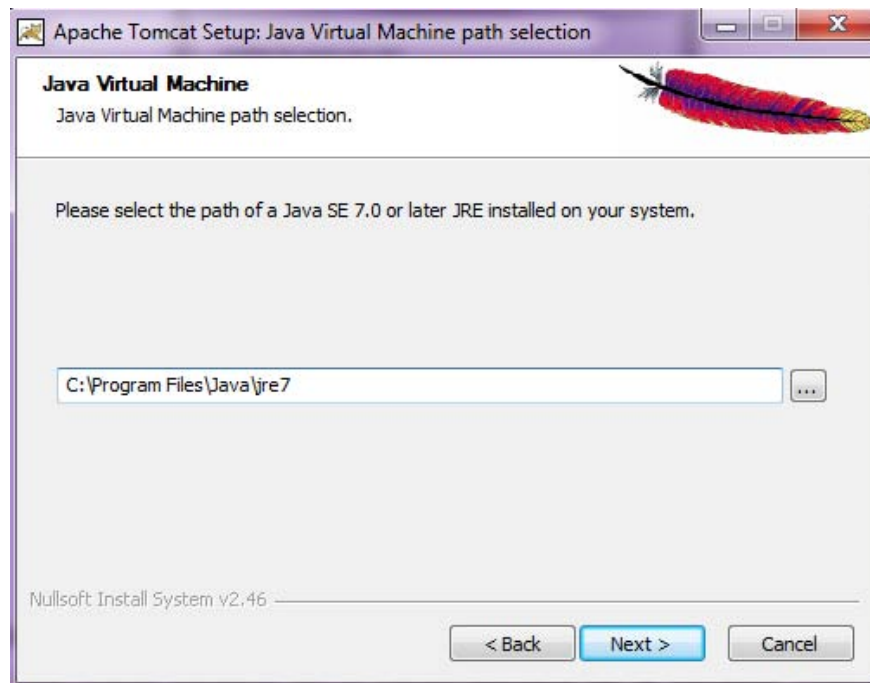


Figura 4.2.25 Selección de la máquina virtual de Java

Finalmente seleccionaremos el botón ***Install***, como se muestra en la figura 4.2.26

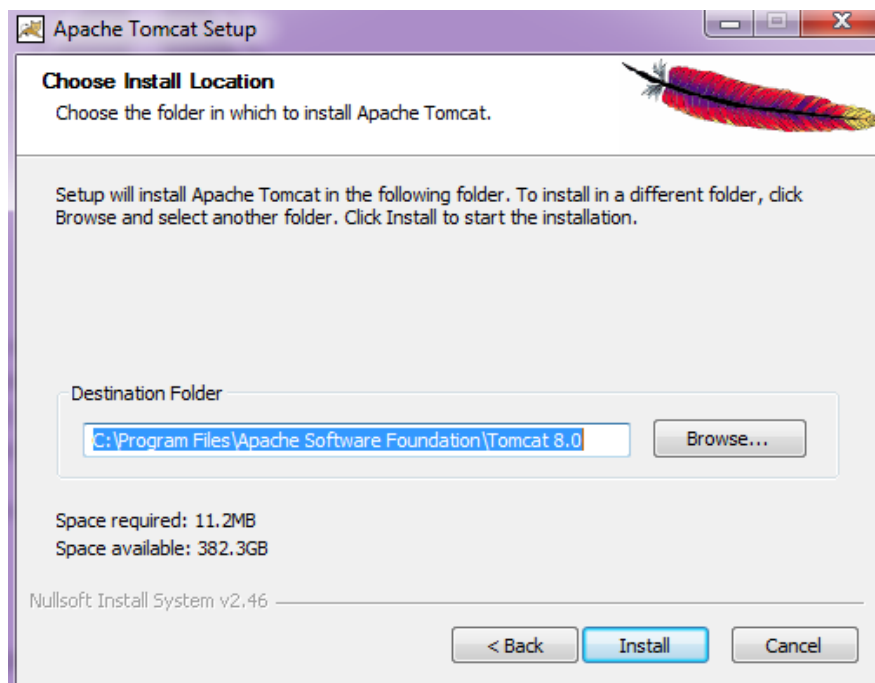


Figura 4.2.26 Finalizando la instalación



Capítulo 4. Diseño y construcción del sistema



Finalmente ingresamos a la dirección web <http://localhost:8080/> desde cualquier navegador, para verificar que la instalación se realizó de manera correcta, como se muestra en la figura 4.2.27.

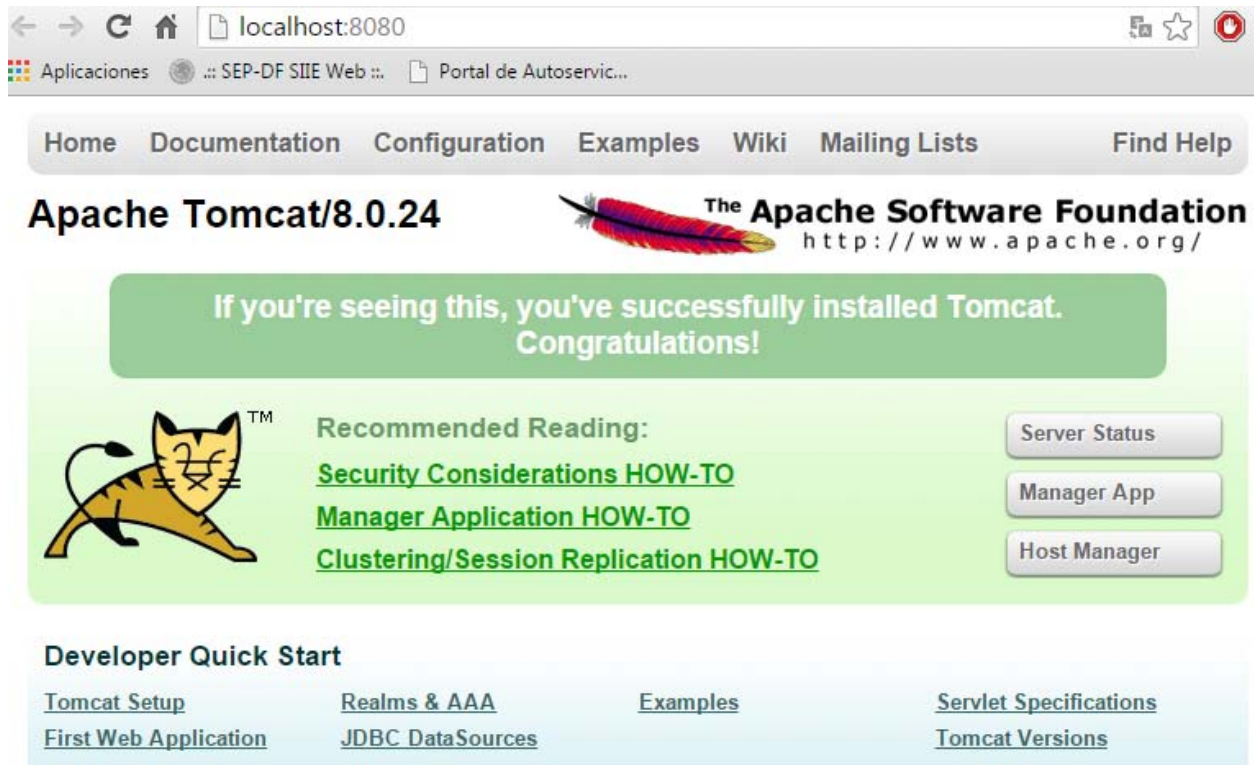


Figura 4.2.27

instalación exitosa de apache Tomcat

4.3 Diseño de la interfaz de usuario.

Implementación de Módulos con Grails

Como se mencionó el inciso 3.3 el sistema cuenta con los siguientes módulos que son:

- Alergias y Comorbilidad
- Signos Vitales
- Control de Líquidos y Medicamentos
- Valoración de Enfermería
- Diagnósticos e Intervenciones



Capítulo 4. Diseño y construcción del sistema



- Indicadores de Calidad
- Jefes / Supervisores

La tecnología empleada para el desarrollo de cada uno de los módulos del sistema es mediante Grails un framework open source rápido, robusto y dinámico para el desarrollo ágil de aplicaciones web, ver figura 4.3.1.



Figura 4.3.1

Logo del framework web Grails

Es rápido, porque al ser un framework de alta productividad permite terminar los proyectos web en semanas en lugar de meses. Al mismo tiempo permite obtener una retroalimentación inmediata y resultados veloces.

Es robusto, porque es potenciado por los frameworks de Spring e Hibernate. Es de alto desempeño y está diseñado para usar el patrón MVC, ver figura 4.3.2.

Es dinámico, porque permite la generación automática y sencilla de las capas de persistencia y presentación. Es extensible mediante plugins y tiene un GUI configurable con JQuery y SiteMesh, ver figura 4.3.2.



Capítulo 4. Diseño y construcción del sistema

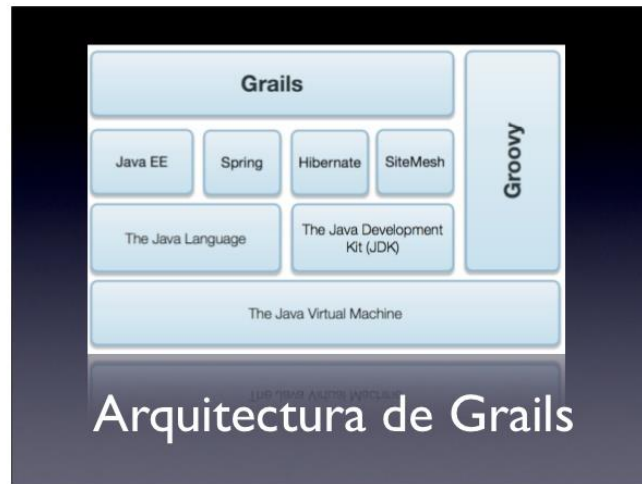


Figura 4.3.2 Arquitectura del framework web Grails

Para el desarrollo de aplicaciones en Grails lo más recomendable es el uso Groovy/Grails Tool Suite un IDE basado en Eclipse, en la figura 4.3.3 se visualiza la página inicial una vez abierto el IDE.

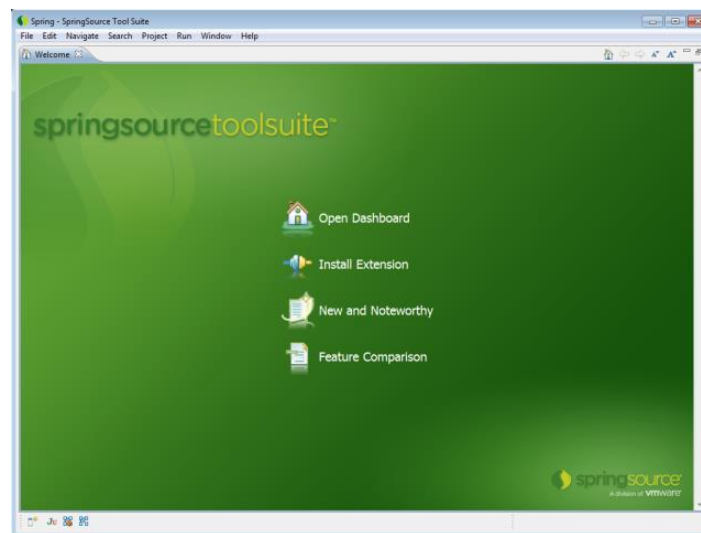


Figura 4.3.3 IDE para el desarrollo en Grails

Grails como otras soluciones de desarrollo ágil como Rails o Play Framework trabaja con comandos para la creación de los componentes, el IDE acelerará esta tarea



Capítulo 4. Diseño y construcción del sistema



mediante **wizards** (asistentes gráficos), para crear una aplicación Grails damos clic en la opción Grails Project y escribimos enfermería, ver figura 4.3.4

Project name:

Grails Installation

Use default Grails installation (currently 'Grails 2.2.3')

Use project specific Grails installation:

Install: [Configure Grails Installations....](#)

Figura 4.3.4 Creación del proyecto en Grails

Grails cuando crea el proyecto, por defecto crea una estructura de directorios definida, estructura que nos permite mantener un orden y conocer exactamente donde se encuentran todos y cada uno de los componentes de nuestra aplicación como se observa en la figura 4.3.5, la creación del proyecto expediente clínico de enfermería.

Convención sobre Configuración

Grails utiliza convención sobre configuración para configurarse. Normalmente, esto significa que el nombre y la ubicación de los archivos se utiliza en lugar de una configuración explícita, por lo tanto, solo se necesita familiarizarse con la estructura de directorios proporcionado por Grails, como se observa en la figura 4.3.5.

- grails-app - Directorio de nivel superior para las fuentes de Groovy
 - conf - Fuentes de configuración.
 - controllers - Controladores web - La C en MVC.
 - domain - El dominio de aplicación - La M en MVC.
 - i18n - Apoyo a la internacionalización (i18n).
 - services - La capa de servicio - La M en MVC.
 - taglib - Las librerías de etiquetas.



Capítulo 4. Diseño y construcción del sistema



- utils - Utilidades específicas.
- views - Páginas Groovy Server - El V en MVC.

- scripts - Tareas automatizadas.

- src
 - Groovy - Otras fuentes Groovy.
 - java - Otras fuentes de Java.

- test - Unidad e integración pruebas.

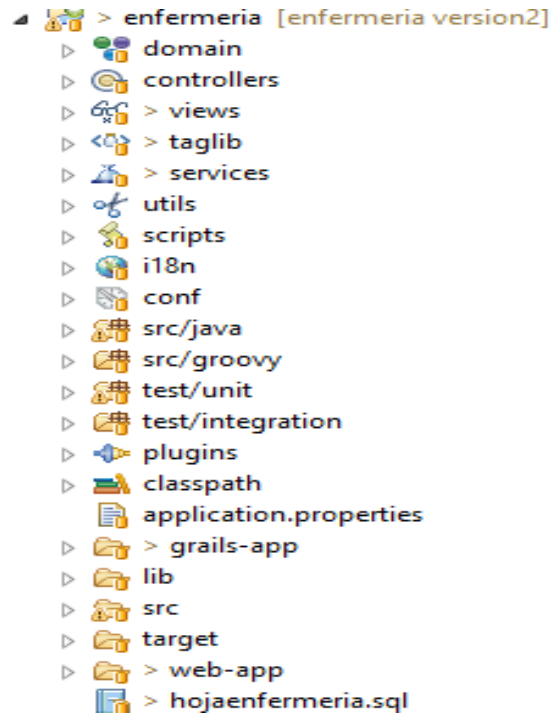


Figura 4.3.5 Estructura de directorios de un proyecto en Grails

Modelo del Dominio

Para definir el modelo del dominio posicionado sobre “domain”, hacemos New > Domain y escribimos CatProcedimientoNotaEnfermeria que es uno de los catálogos del sistema, esto creará un archivo Groovy donde se definirán atributos, relaciones con



Capítulo 4. Diseño y construcción del sistema



otros modelos y como este modelo será mapeado a un tabla en el modelo entidad relación por medio del apartado mapping, ver figura 4.3.6.

```
class CatProcedimientoNotaEnfermeria {  
  
    Long id  
    String descripcion  
    CatRubroNotaEnfermeria padre  
    String estatus  
    Boolean vista  
  
    static constraints = {  
    }  
  
    static mapping = {  
        table 'catprocedimientosnotaenfermeria'  
        version false  
        id column:'idprocedimiento'  
        descripcion column:'desprocedimiento'  
        padre column:'idrubro'  
        cache usage:'read-only'  
    }  
}
```

Figura 4.3.6 Definición de un modelo en Grails

Para lograr la comunicación con la base de datos en el archivo DataSource.groovy que está en el carpeta conf podremos ver que Grails hace uso de environments (development, test, production) para definir la url jdbc de conexión a la base de datos tal como se usa en Java y la opción dbCreate con el valor create-drop lo que hará es crear las tablas con sus campos, claves primarias, foráneas a partir del modelo del dominio, ver figura 4.3.7.

```
development {  
    dataSource {  
        dbCreate = "create-drop" // one of 'create', 'create-drop', 'update', 'validate', ''  
        url = "jdbc:postgresql://bulbix.cbhbdysmftcx.us-west-2.rds.amazonaws.com:5432/hojaenfermeria"  
        logSql = true  
    }  
}
```

Figura 4.3.7 Definición de la url jdbc de conexión a la base de datos

Servicios



Capítulo 4. Diseño y construcción del sistema



Los servicios en Grails nos ayudarán a encapsular la lógica del negocio en el expediente clínico de enfermería a través de la definición de métodos.

Para definir un servicio posicionado sobre “services”, hacemos New > Service y escribimos *HojaRegistroClinico* que es uno de los servicios del sistema, esto creará un archivo Groovy donde implementaremos el método *existeTurno*, así mismo se pueden ver la inyección de dependencias de otros servicios como *utilService*, ver figura 4.3.8.

```
class HojaRegistroClinicoService {  
  
    static transactional = true  
  
    def signosVitalesService  
    def valoracionEnfermeriaService  
    def utilService  
    def controlLiquidadosMedicamentosService  
    def indicadoresCalidadService  
  
    boolean existeTurno(Long idHoja, String turno){  
  
        def result = false  
  
        def registroTurno = HojaRegistroEnfermeriaTurno.createCriteria().get{  
            eq("hoja.id", idHoja)  
            eq("turno", Turno."${turno}")  
            maxResults(1)  
        }  
  
        if(registroTurno){  
            result = true  
        }  
  
        result  
    }  
}
```

Figura 4.3.8 Definición de un servicio en Grails

Controladores y Vistas

Para definir un controlador posicionado sobre “controllers”, hacemos New > Controller y escribimos *HojaRegistroClinico* que es uno de los controladores del sistema, dentro del controlador los métodos se denominan *action*, en ellos se ven llamadas a los servicios y los valores regresados son lo que utiliza la vista, ver figura 4.3.9.



Capítulo 4. Diseño y construcción del sistema



```
class HojaRegistroClinicoController {  
  
    def reporteHojaFacadeService  
    def hojaRegistroClinicoService  
    def utilService  
    def springSecurityService  
    def jefeSupervisorService  
  
    /**  
    * Pantalla Principal de la hoja  
    * @return  
    */  
    @Secured(['ROLE_ENFERMERIA'])  
    def index(){  
        def hojaInstance = new HojaRegistroEnfermeria()  
        hojaInstance.rubrosValoracion = utilService.consultarCatalogoRubro(S_VALORACION)  
        hojaInstance.rubrosDiagnostico = utilService.consultarCatalogoRubro(S_DIAGNOSTICOS_INTERVENCIONES)  
        hojaInstance.rubrosIndicador = utilService.consultarCatalogoRubro(S_INDICADORES_CALIDAD,true)  
        def pisos = utilService.consultarPisos()  
        def usuarioActual = springSecurityService.currentUser  
  
        hojaInstance.fechaElaboracion = hojaRegistroClinicoService.asignarFechaElaboracion()  
  
        [hojaInstance:hojaInstance,pisos:pisos,usuarioActual:usuarioActual]  
    }  
}
```

Figura 4.3.9 Definición de un controlador en Grails

También tendremos en el proyecto la carpeta views y otra llamada hojaRegistroClinico que se generó al crear el controlador y dentro de ella un archivo llamado index.gsp que es el mismo nombre del action de la clase controladora, es aquí donde se define todo el HTML y la comunicación con el controlador por medio de los valores devueltos que son hojaInstance, pisos, usuarioActual, en la figura 4.3.10, se ve el uso de los valores devueltos en la vista, se usa Groovy Server Pages que son etiquetas especiales para leer estos valores.

```
<table id="tablaLectura" style="display:none">  
    <tr>  
        <td>  
            <label>Paciente:</label>  
            <label class="cabecera">${hojaInstance?.paciente}</label>  
        </td>  
        <td>  
            <label>Turno:</label>  
            <label class="cabecera">${hojaInstance?.turnoActual}</label>  
        </td>  
        <td>  
            <label>Fecha:</label>  
            <label class="cabecera">${hojaInstance?.fechaElaboracion?.format('dd/MM/yyyy')}</label>  
        </td>  
    </tr>  
</table>
```

Figura 4.3.10 Impresión de los valores devueltos por el controlador en la vista.



TagLibs

El código HTML debe poder ser reutilizado cuando este solo lleve pequeñas variaciones en su interior para este cometido Grails usa taglibs, para definir un taglib posicionado sobre “taglib”, hacemos New > Taglib y escribimos Enfermería que es uno de los taglibs del sistema, dentro del taglib los métodos son etiquetas que podrán ser usadas dentro de las vistas, ver figura 4.3.11.

```
class EnfermeriaTagLib {  
  
    def utilService  
    def hojaRegistroClinicoService  
    def springSecurityService  
  
    def usuarioActual={ attrs, body ->  
  
        def html = springSecurityService.currentUser.nombre  
        out << html  
    }  
}
```

Figura 4.3.11 Definición de un taglib en Grails

En la figura 4.3.12 se puede observar el uso de la etiqueta predefinida usuarioActual, que devuelve el usuario logueado en el sistema.

```
<tr><td>  
    <div style="text-align:center; font-size:20px">  
        <sec:ifLoggedIn>  
            Usuario: <span style="color:blue"><g:usuarioActual/></span>  
        </sec:ifLoggedIn>  
    </div>  
</td></tr>
```

Figura 4.3.12 Uso de taglib en la vista en Grails

Para el despliegue de la aplicación web, Grails viene integrado con Apache Tomcat que es un contenedor de Servlets y JSP, solo es necesario dar clic derecho en el proyecto ir a la opción Run As y ejecutar run-app, para que Grails nos muestre la url del expediente clínico de enfermería para visualizar en un navegador web, ver figura 4.3.13.



```
Configuring Spring Security Core ...  
... finished configuring Spring Security Core  
| Server running. Browse to http://localhost:8091/enfermeria
```

Figura 4.3.13 Despliegue de la aplicación en Apache Tomcat

Pantallas Expediente Clínico de Enfermería

El diseño e implementación de las pantallas del expediente clínico de enfermería es la combinación de los componentes Grails anteriormente descritos, usando las mejores prácticas, patrones de diseño, apegados a los requerimientos generales del sistema descrito en el inciso 3.2.

Grails hace uso extensivo de plugins que nos darán funcionalidad agregada con módulos repetitivos como lo es el inicio de sesión y el manejo de roles, para este caso se usó el plugin Spring Security dando como resultado un formulario típico de inicio de sesión, el cual está conformado por los siguientes elementos básicos: (Ver Figura 4.3.14)

Please Login

Username:

Password:

Remember me

Login

Figura 4.3.14 Inicio de sesión con plugin Grails



Capítulo 4. Diseño y construcción del sistema



Una vez ingresado un RFC y password correcto el enfermero tendrá la siguiente pantalla principal con una de barra de herramientas, filtros de búsqueda, búsqueda de paciente, selección de turno y fecha, descripción de los datos del paciente y los módulos mencionados se representan en cada pestaña (Ver Figura 4.3.15)

Nuevo Mis Hojas

Piso: [TODOS] Servicio: [TODOS] Cargar Historicos:

Paciente: Turno: MATUTINO Fecha: 02/08/2015

Edad: Sexo: Religion: Cama: Dias Hosp:

Servicio: Diagnostico:

Alergias y Comorbilidad Signos Vitales Control de Líquidos y Medicamentos Valoración de Enfermería Diagnosticos e Intervenciones Indicadores de Calidad

GUARDAR / FIRMAR

Comorbilidad: HAS DM NEF IC IR

Peso: kg. Alergias: Otro:

Talla: m.

Figura 4.3.15 Pantalla principal, expediente clínico de enfermería

Ya sea para empezar un nuevo expediente clínico o consultar uno existente, el enfermero debe primero buscar el paciente, para este fin el sistema como en otros apartados hace uso de JQuery UI en la vista de Grails que es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plugins, widgets y efectos visuales para la creación de aplicaciones web, se usó el widget autocomplete como se observa en la figura 4.3.16

Paciente:

(N-0000000000) Prado Flores Luis 1 I

Figura 4.3.16 Autocomplete de pacientes



Capítulo 4. Diseño y construcción del sistema



Una vez seleccionado el paciente se cargaran los datos del paciente, para realizar este proceso como en otros apartados se hace extensivo de JQuery con AJAX en la vista de Grails para la comunicación con el server para no recurrir a la recarga de la página lo que provocaría una experiencia no grata en el usuario, el enfermero ahora podrá empezar a registrar el módulo de alergias y comorbilidad, para después dar clic en el botón Guardar/Firmar se le pedirá confirmación ver figura 4.3.17 y se abrirán los demás módulos.

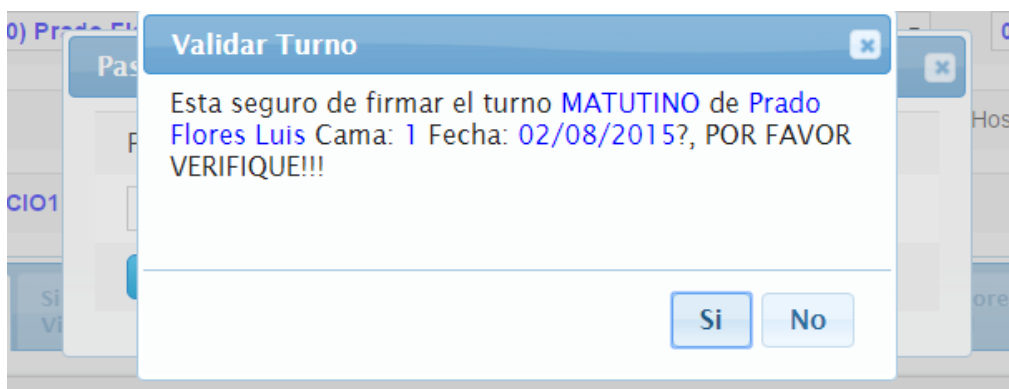


Figura 4.3.17 Confirmación de apertura de expediente

También el enfermero podrá consultar si el paciente ya tiene expediente clínico abierto, dando clic en el botón Abrir visualizará la siguiente pantalla, como se observa en la figura 4.3.18

Hojas Disponibles: (N-0000000000) Prado Flores Luis I I

Asociar turno: MATUTINO

Fecha Elaboracion	Matutino	Vespertino	Nocturno	Cargar Asociar	Traslado Paciente
26/07/2015	Luis			ACEPTAR	
20/07/2015	Luis			ACEPTAR	

Figura 4.3.18 Expedientes clínicos del paciente



Capítulo 4. Diseño y construcción del sistema



Para el llenado de los signos vitales aparece un solo renglón donde se podrá ir agregando más conforme se ha clic al botón Agregar Signo Vital como se observa en la figura 4.3.19.

AGREGAR SIGNO VITAL Para eliminar todo un renglon deben quedar vacios todos los signos vitales
 Signo vital Lab y Gab con hora 1 guardado

Hora	Temp.	F.C.	T.A. Sistolica	T.A. Diastolica	Frec Resp	Lab y Gab
1	45	78	80	80	45	xx
2						

Figura 4.3.19

Llenado de signos vitales

Para el llenado de la escala del dolor, se implementó de manera visual la selección del grado del dolor con la selección de una hora asociada como se observa en la figura 4.3.20.

MOSTRAR ESCALA DEL DOLOR Hora: 1

Figura 4.3.20

Llenado de la escala del dolor

Para el llenado de los ingresos, se dejaron dos registros fijos Medicamento Oral y Vía Oral por ser recurrentes, también por medio del botón Agregar Ingreso podremos ir agregando renglones de registro como se observa en la figura 4.3.21.

AGREGAR INGRESO

Ingreso	Hora Inicio	Hora Fin	Cantidad	Ex
Medicamento Oral	1	1		0
Via Oral	1	1		0

Figura 4.3.21

Llenado de los ingresos



Capítulo 4. Diseño y construcción del sistema



Para el llenado de los egresos, todos los elementos están ya definidos como se observa en la figura 4.3.22

Egreso	Hora Inicio	Hora Fin	Registro		
Diuresis	1	24		AGREGAR	MOSTRAR
Cateterismo	1	1		AGREGAR	MOSTRAR
Fuga	1	1	+ <input type="radio"/> ++ <input type="radio"/> +++ <input type="radio"/>	AGREGAR	MOSTRAR
Evacuacion	1	1		AGREGAR	MOSTRAR
Drenajes	1	1		AGREGAR	MOSTRAR
Vomito	1	1	<input type="checkbox"/> No presenta	AGREGAR	MOSTRAR
Sangrado	1	1		AGREGAR	MOSTRAR
Sonda Vesical	1	1		AGREGAR	MOSTRAR

.Figura 4.3.22 Llenado de los egresos

Para el llenado de los medicamentos, se podrán ir agregando tantos como se necesiten por medio del botón Agregar Medicamento como se observa en la figura 4.3.23. El sistema hará uso del widget JQuery UI Autocomplete para facilitar el registro.

AGREGAR MEDICAMENTO

Medicamento	Hora	Dosis			
naftalina	1	30 mg	AGREGAR	MOSTRAR	CAMBIAR
	1		AGREGAR	MOSTRAR	CAMBIAR

Figura 4.3.23 Llenado de los medicamentos

Para el llenado de la Escala de Glasgow se definió el apartado de Escala de Glasgow y Otro, donde se pueden ver renglones ya fijos que son recurrentes y se pueden ir agregando más renglones dando clic a botón Escala Glasgow y Otro como se observa en la figura 4.3.24.



Capítulo 4. Diseño y construcción del sistema



AGREGAR ESCALA GLASGOW Y OTRO				
Escala Glasgow Otro	Hora	Dosis		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Respuesta Ocular	1		AGREGAR	MOSTRAR
Respuesta Verbal	1		AGREGAR	MOSTRAR
Posicion en cama	1		AGREGAR	MOSTRAR
Perimetros	1		AGREGAR	MOSTRAR
Glucosa Capilar	1		AGREGAR	MOSTRAR

Figura 4.3.24 Llenado de la Escala de Glasgow y Otros

En los módulos de Valoración de Enfermería, Diagnósticos e Intervenciones e Indicadores de Calidad se presentan tablas de selección y opción de procedimientos de enfermería, para la implementación se hizo uso del componente taglib de Grails, como se observa en la figura 4.3.25.

Diagnosticos1	Indicadores1
Registro1 <input type="checkbox"/> M <input type="checkbox"/> V <input type="checkbox"/> N	Registro1 <input type="radio"/> Si <input type="radio"/> No
Registro2 <input type="checkbox"/> M <input type="checkbox"/> V <input type="checkbox"/> N	Registro2 <input type="radio"/> Si <input type="radio"/> No
Registro3 <input type="checkbox"/> M <input type="checkbox"/> V <input type="checkbox"/> N	Registro3 <input type="radio"/> Si <input type="radio"/> No
Otro: <input type="text"/>	Otro: <input type="text"/>

Figura 4.3.25 Tablas de Selección y Opción implementadas con taglibs

En el módulo Jefes/Supervisores se implementó un panel de control, donde se despliegan todos los pisos y en ellos los pacientes hospitalizados, para que una vez hecho clic en el turno, el jefe o supervisor valide el expediente capturado y pueda firmar de conformidad, ver figura 4.3.26.



Capítulo 4. Diseño y construcción del sistema



Paciente Historico: M V N Nueva Busqueda

001

Paciente

(N-00000000000) Prado Flores Luis 1 I M V N

002

Figura 4.3.26 Panel de Control Jefes y Supervisores

El resultado final de la captura de los pantallas de los módulos del expediente clínico de enfermería resulta en un archivo pdf que podrá generarse dando clic en PDF en la barra de herramientas ver figura 4.3.27 y 4.3.278

Nuevo Abrir PDF Mis Hojas

Paciente: (N-00000000000) Prado Flores Luis Turno: MATUTINO Fecha: 02/08/2015

Figura 4.3.27 Barra de herramientas con botón PDF

NOMBRE	Prado Flores Luis	FECHA NAC	15/10/2000	EDAD	15	SEXO	H	CAMA	1
REGISTRO	N-00000000000	SERVICIO	SERVICIO1	DIAGNOSTICO	DIAGNOSTICO1				
		DIAS DE ESTANCIA	0	FECHA	02/08/2015				

LABORATORIO Y GABINETE	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7
S I G N O S V I T A L E S																		xx						
Frecuencia Cardíaca																		78						
Frecuencia Respiratoria																		45						
Tension Arterial Sistólica																		80						
Tension Arterial Diastólica																		80						
Temperatura																		45						

PESO: 80.0Kg. TALLA: 1.8m.

SIN DOLOR 0 1 2 3 4 5 6 7 8 9 10 PEOR DOLOR

Figura 4.3.28 Fragmento del expediente clínico de enfermería generado por el sistema



4.4 Generación de pruebas y mantenimiento.

Pruebas²³

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software. La prueba de software es un elemento crítico para la garantía del correcto funcionamiento del software. Entre sus objetivos están:

- Detectar defectos en el software.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar casos de prueba que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Pruebas de integración

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez.

Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos. Es la fase de la prueba de software en la cual módulos individuales de software son combinados y probados como un grupo.

²³ Sitio WEB: http://www.ecured.cu/index.php/Pruebas_de_software
Consultada: 05-08-15



Pruebas de regresión

Se denominan pruebas de regresión a cualquier tipo de pruebas de software que intentan descubrir errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa.

Por lo tanto, en la mayoría de las situaciones del desarrollo de software se considera una buena práctica que cuando se localiza y corrige un **bug** (vulnerabilidad de software), se grave una prueba que exponga el bug y se vuelvan a probar regularmente después de los cambios subsiguientes que experimente el programa.

Las Pruebas de Regresión pueden usarse no solo para probar la corrección de un programa, sino a menudo usarse para rastrear la calidad de su salida.

Pruebas de validación

Las pruebas de validación son el proceso de revisión que verifica que el sistema de software producido cumple con las especificaciones y que logra su cometido. La validación es el proceso de comprobar que lo que se ha especificado es lo que el usuario realmente quería.

Se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales.

Tipos:

- Pruebas de aceptación: desarrolladas por el cliente.
- Pruebas alfa realizadas por el usuario con el desarrollador como observador en un entorno controlado (simulación de un entorno de producción).



Capítulo 4. Diseño y construcción del sistema



- Pruebas beta: realizadas por el usuario en su entorno de trabajo y sin observadores.

Prueba de estrés

Esta prueba se utiliza normalmente para romper la aplicación. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Pruebas Alfa

La versión alfa, es la primera versión del programa, la cual es enviada a los verificadores para probarla. Algunos equipos de desarrollo utilizan el término alfa informalmente para referirse a una fase donde un producto todavía es inestable, aguarda todavía a que se eliminen los errores o a la puesta en práctica completa de toda su funcionalidad, pero satisface la mayoría de los requisitos.

Se llevan a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas alfa se hacen en un entorno controlado. Se realizan después de que todos los procedimientos de prueba básicos, como las pruebas unitarias y pruebas de integración se han completado, y se produce después de las pruebas del sistema. Por lo general emplea a cualquiera de las pruebas de caja blanca o caja negra para probar el software. El sistema está probado para la funcionalidad de los empleados de la compañía solamente, y se trata de pruebas de navegación, entrada y salida de los mecanismos del software, Esta no es la versión final de software y cierta funcionalidad puede ser añadida al software incluso después de la prueba alfa.



Pruebas Beta

La versión Beta, representa generalmente la primera versión completa del producto, que es posible que sea inestable pero útil para que las demostraciones internas y las inspecciones previas seleccionen a clientes. Esta etapa comienza a menudo cuando los desarrolladores anuncian una congelación de las características del producto, indicando que no serán agregadas más características a esta versión y que solamente se harán pequeñas ediciones o se corregirán errores. Muchos de estos programas beta, son de uso privado solo permitiendo a un número determinado de usuarios probarlo, y de esta manera mantener un control más eficiente de la calidad y las opiniones de los usuarios que lo están probando. Este tipo de programas casi siempre incluye instrucciones específicas para reportar estos bugs y recibir ayuda en caso de ser necesario.

Dichas pruebas se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador. Las pruebas beta es la última fase de las fases de prueba y se hace utilizando técnicas de caja negra. La prueba involucra a los usuarios y que compruebe la funcionalidad requerida.

Pruebas de caja blanca

Son pruebas de software que se realiza sobre las funciones internas de un módulo. Están dirigidas a las funciones internas. Entre las técnicas usadas se encuentran:

- La cobertura de caminos: pruebas que hagan que se recorran todos los posibles caminos de ejecución.
- Pruebas sobre las expresiones lógico-aritméticas.
- Pruebas de camino de datos. Definición-uso de variables.



Capítulo 4. Diseño y construcción del sistema



- Comprobación de bucles. Se verifican los bucles para una o más interacciones, y luego para las interacciones máximas, máximas menos uno y más uno.

Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas (integración).

En los sistemas orientados a objetos, las pruebas de caja blanca pueden aplicarse a los métodos de la clase, pero según varias opiniones, ese esfuerzo debería dedicarse a otro tipo de pruebas más.

Pruebas de caja negra

Las pruebas de caja negra se centran principalmente en lo que “se quiere” de un módulo o sección específica de un software. Las pruebas de caja negra se limitan a que el **tester** (Ejecutor de pruebas) pruebe con “datos” de entrada y estudie como salen, sin preocuparse de lo que ocurre en el interior (Figura 4.4.1).

Éstas, principalmente, se centran en módulos de interfaz de usuario pero suelen ser útiles en cualquier módulo ya que todos o la mayoría tienen datos de entrada y salida que se pueden comprobar y verificar.

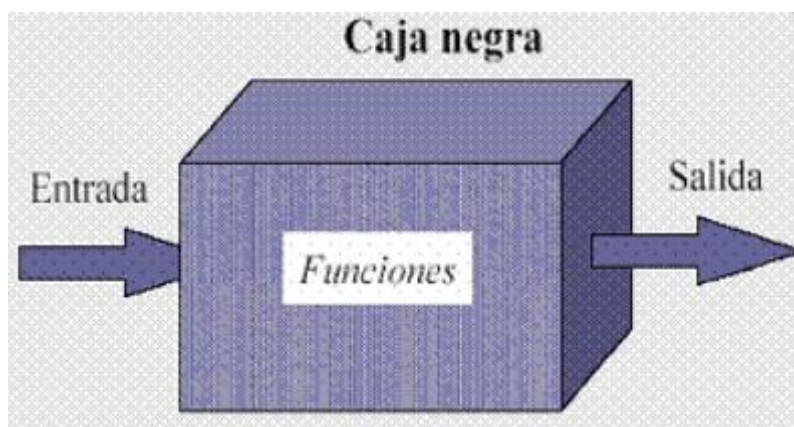


Figura 4.4.1 Representación pruebas de caja negra



Pruebas al sistema de expediente clínico de enfermería

Al sistema de expediente clínico se le realizaron las pruebas unitarias o de funcionalidad a través de una matriz de pruebas (Figura 4.4.5). Los casos y resultados se describen a continuación:

Caso 1. Se debe validar la autenticación. El usuario debe ingresar con un usuario y contraseña precargados en la base de datos. Si uno o ambos campos para esta autenticación no se encuentran en la base de datos, el sistema no permite acceder a los siguientes menús. El caso fue exitoso porque únicamente se permitió el acceso con el usuario y contraseña registrados en la base de datos (Figura 4.4.2).

Figure 4.4.2 is a screenshot of a web application's login page. The page has a light blue background. At the top, it says "Please Login". Below that, a red error message reads: "Sorry, we were not able to find a user with that username and password." There are two input fields: "Username:" with the text "ADRIAN" and "Password:" with five black dots. Below the password field is a checkbox labeled "Remember me". At the bottom is a "Login" button.

Figura 4.4.2 Acceso erróneo del sistema

Caso 2. La eliminación de los catálogos, únicamente la puede realizar el usuario con perfil de jefe o supervisor por cuestiones de seguridad. Si el usuario es diferente al perfil mencionado, el sistema no permite la acción de eliminar catálogos y muestra un mensaje. El caso fue exitoso porque se hizo la validación para borrar por el tipo de usuario.

Caso 3. El sistema digitaliza la firma por medio de un dispositivo electrónico, posteriormente se pide una contraseña para asociarla a la firma digital. No se puede dar de alta un usuario si no se tiene capturada su firma digital y su contraseña. El caso fue exitoso porque cuando se dio de alta el usuario, el sistema pide una contraseña y firma



Capítulo 4. Diseño y construcción del sistema



digital, si uno o ambos falta, el registro no se puede dar de alta y mostró un mensaje de error.

Caso 4. El sistema de registro clínico tiene a primera instancia el llenado de datos de comorbilidad para el cuidado o monitoreo del paciente, si falta uno o varios campos, no sigue al registro posterior. El caso fue exitoso ya que obliga al usuario a llenar por medio de mensajes de error de falta de datos, sino no se habilita la siguiente pestaña.

Caso 5. El usuario tiene 10 minutos como máximo para llenar el registro clínico. Si no se llena en el tiempo estipulado, se manda un mensaje de error de tiempo. El caso fue exitoso porque al llenar el expediente clínico, se registra exitosamente en la base. En el caso que excedió los 10 minutos, se mandó un mensaje de término de sesión.

Caso 6. El sistema muestra todos los registros hechos para un paciente realizados por los enfermeros y médicos en turno. El caso fue exitoso, ya que se realizaron tres casos de prueba con un paciente atendido por un solo enfermero y otro atendido por varios médicos o enfermeros y en ambos casos se imprimió el reporte completo.

Caso 7. El sistema debe guardar automáticamente el reporte cada vez que haya un cambio. Este caso se realizó con éxito porque cada dos segundos se actualiza la información en la base mientras haya cambios en el documento.

Caso 8. El usuario con perfil de enfermero, podrá modificar sus propios registros y sólo consultar el de los demás. Se creó de inicio un expediente, el creador de dicho documento, pudo consultarlo y modificarlo sin ningún problema. Un usuario diferente pudo consultar el expediente en cuestión pero no pudo realizar actualizaciones, por lo tanto el caso se realizó con éxito.

Caso 9. El sistema alerta al usuario para recabar la información de signos vitales del paciente cada hora. Se registró la información de signos vitales a las 14:00hrs. En el



Capítulo 4. Diseño y construcción del sistema



expediente clínico y se mantuvieron 59 minutos sin actualizar el documento. Pasando exactamente la hora, el sistema manda una alerta para realizar la actualización, siendo este caso exitoso.

Caso 10. El sistema alerta al usuario para el registro de la escala del dolor del paciente por medio de imágenes. Se registró la escala de dolor de un paciente a las 12:00hrs. Justo a las 13:00hrs, se alertó para actualizar el registro de la escala del dolor quedando el caso como exitoso.

Caso 11. El sistema alerta al usuario para recabar la información de altas, bajas y medicamentos cada hora. La información de altas, bajas y medicamentos, se realizó a las 6:00hrs y una hora después, el sistema alertó al usuario para realizar la actualización, por lo tanto el caso fue exitoso.

Caso 12. Autocompletar la información de pacientes ya registrados en el sistema. El caso se realizó con éxito ya que en la búsqueda del paciente se digitó una parte de su nombre y salieron todos los resultados que coincidieron en la consulta, así como el número de registro. El caso fue exitoso (Figura 4.4.3).

Pacientes de urgencias turno nocturno despues de las 12 ajustar la fecha a la actual. NO si es continuacion paciente dia anterior

Mis Hojas

Piso: [TODOS] Servicio: [TODOS] Cargar Historicos:

Paciente: Turno: [MATUTINO] Fecha: [05/08/2015]

Edad: Sexo: Religion: Cama: Dias Hosp:

Servicio: Diagnostico:

Alergias y Comorbilidad	Signos Vitales	Control de Liquidos y Medicamentos	Valoración de Enfermería	Diagnosticos e Intervenciones	Indicadores de Calidad
-------------------------	----------------	------------------------------------	--------------------------	-------------------------------	------------------------

Figura 4.4.3 Autocompletar de datos del paciente



Capítulo 4. Diseño y construcción del sistema



Caso 13. Autocompletar los rubros donde se encuentran los procedimientos de enfermeros y médicos. El caso resulto exitoso, ya que con alguna palabra que contuviera el procedimiento, se mostraban todos los resultados coincidentes.

Caso 14. Al realizar un registro, el sistema pide la contraseña del personal que modifica el expediente. El caso fue exitoso ya que para cualquier modificación del expediente clínico, se pide la contraseña asociada a la firma digital y cuando no se le pone o no coincide con la que está en la base de datos, se manda un mensaje de error(Figura 4.4.4).

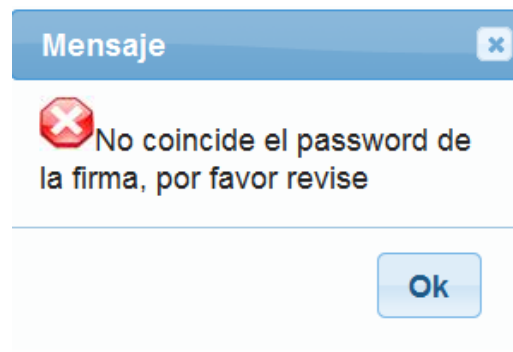


Figura 4.4.4 Contraseña errónea para firma electrónica.

Caso 15. El sistema, permite la actualización de los registros únicamente a los usuarios con perfil de jefe o supervisor. Se trató de realizar la modificación con perfil de enfermero de un registro generado por otro enfermero y se mandó un mensaje de error. Por otro lado, el usuario con perfil de jefe trato de modificar el mismo registro y el resultado fue exitoso.

Caso 16. El sistema permite hacer consultas de pacientes que ya fueron dados de alta. El caso fue exitoso porque se realizó la consulta con el perfil de enfermero y con el perfil de supervisor, para revisar el expediente clínico de un usuario que fue dado de alta una hora antes y en ambos casos se mostró dicho expediente.

Caso 17. El sistema muestra el expediente clínico en formato pdf, de todos los que hayan modificado dicho expediente. Se imprimió el expediente clínico en formato pdf y



Capítulo 4. Diseño y construcción del sistema



se registraron las firmas de los médicos y enfermeros de todos los turnos que modificaron este expediente. Por lo tanto el caso fue exitoso.

Caso 18. El sistema únicamente permite la modificación de los expedientes a los enfermeros hasta las 00:00hrs. El caso fue exitoso porque se realizó la modificación de un expediente creado por un enfermero a las 23:59hrs. Dos minutos después, ya no permitió la modificación para el enfermero que lo creó.



Capítulo 4. Diseño y construcción del sistema



# De caso	Caso de prueba	Caso	Resultados esperados	Resultados obtenido
1	Loguearse al sistema web	Loguearse con usuario y contraseña validos	Ingresar	Ingresar-OK
		Loguearse con usuario valido y contraseña invalida	Rechazar	Rechazar-OK
		Loguearse con usuario invalido y contraseña valida	Rechazar	Rechazar-OK
		Loguearse con usuario invalido y contraseña invalida	Rechazar	Rechazar-OK
2	Restringir la eliminación de catálogos, sólo podrán hacerla los usuarios con perfil de jefes o supervisores	Usuario con perfil de jefe elimina catálogos	Aprobar	Aprobar-OK
		Usuario con perfil de enfermero elimina catálogos	Rechazar	Rechazar-OK
3	Recabar las firmas en formato electrónico	Registrar usuario con firma electrónica	Aprobar	Aprobar-OK
		Registrar usuario sin firma electrónica	Rechazar	Rechazar-OK
4	Registrar los datos clínicos del paciente	Registrar datos clínicos del paciente	Insertar	Insertar-OK
		Registrar datos clínicos del paciente faltando un campo	Rechazar	Rechazar-OK
		Registrar sólo un dato clínico del paciente	Rechazar	Rechazar-OK
		Insertar el registro sin datos clínicos del paciente	Rechazar	Rechazar-OK
5	Restringir el tiempo del enfermero en el tiempo para el llenado del expediente	Llenar el expediente en 5 minutos	Insertar	Insertar-OK
		Llenar el expediente en 20 minutos	Rechazar	Rechazar-OK
		Llenar el expediente en una hora	Rechazar	Rechazar-OK
		Llenar el expediente y no cerrar el llenado	Rechazar	Rechazar-OK
6	Imprimir un expediente electrónico para mostrar el registro de los pacientes atendidos por un enfermero a lo largo del día	Imprimir expediente electrónico con 3 pacientes atendidos por un enfermero	Imprimir registros	Imprimir registros-OK
		Imprimir expediente electrónico con 1 paciente atendido por un enfermero	Imprimir registros	Imprimir registros-OK
		Imprimir expediente electrónico sin pacientes atendidos por ningún enfermero	Rechazar	Rechazar-OK
7	Guardar de forma automática el expediente clínico en determinado tiempo	Llenar expediente clínico y verificar el autoguardado en 1 segundos	Guardar	Guardar-OK
		Llenar expediente clínico y verificar el autoguardado en 10 segundos	Guardar	Guardar-OK
		No Llenar expediente clínico y verificar el autoguardado en 30 segundos	No Guardar	No Guardar-OK
		No Llenar expediente clínico y verificar el autoguardado en 1 minuto	No Guardar	No Guardar-OK
8	Modificar los registros propios del autor y consultar el de otros.	Usuario con perfil de enfermero modifica registro propio	Aprobar	Aprobar-OK
		Usuario con perfil de enfermero modifica registro ajeno	Rechazar	Rechazar-OK
		Usuario con perfil de jefe modifica registro propio	Aprobar	Aprobar-OK
		Usuario con perfil de jefe modifica registro ajeno	Aprobar	Aprobar-OK



Capítulo 4. Diseño y construcción del sistema



9	Recabar la información cada hora de los signos vitales.	Recabar la información de signos vitales 1 hora después de el registro pasado	No manda alerta	No manda alerta-OK
		Recabar la información de signos vitales 1 minuto después de el registro pasado	No manda alerta	No manda alerta-OK
		Recabar la información de signos vitales 1 hora con 5 minutos después de el registro pasado	Manda alerta	manda alerta-OK
10	Recabar la información cada hora de la escala del dolor por medio de imágenes	Recabar la información de la escala de dolor 1 hora después de el registro pasado	No manda alerta	No manda alerta-OK
		Recabar la información de escala de dolor 1 minuto después de el registro pasado	No manda alerta	No manda alerta-OK
		Recabar la información de escala de dolor 1 hora con 5 minutos después de el registro pasado	Manda alerta	manda alerta-OK
11	Recabar la información de altas, bajas y medicamentos cada hora	Recabar la información de altas, bajas y medicamentos 1 hora después de el registro pasado	No manda alerta	No manda alerta-OK
		Recabar la información de altas, bajas y medicamentos 1 minuto después de el registro pasado	No manda alerta	No manda alerta-OK
		Recabar la información de altas, bajas y medicamentos 1 hora con 5 minutos después de el registro pasado	Manda alerta	manda alerta-OK
12	Autocompletar los campos de los pacientes ya registrados	Escribir datos que si están registrados	Autocompletar	Autocompletar-OK
		Escribir datos que no están registrados	No autocompletar	No Autocompletar-OK
13	Autocompletar los rubros donde se encuentran los procedimientos por medio de palabras clave.	Escribir palabras que sean clave	Autocompletar	Autocompletar-OK
		Escribir palabras que no sean clave	No autocompletar	No Autocompletar-OK
14	Pedir un password asociado a la firma digital, al realizar un registro	Realizar un registro	Pide Password	Pide Password -OK
		Teclear datos sin registrarlos	No pide Password	No pide Password-OK
15	Permitir a los usuarios con perfil de jefes y supervisores, hacer correcciones del turno correspondiente y dar vistos buenos de los enfermeros en turno	Usuario con perfil de jefe da visto bueno	Aprobar	Aprobar-OK
		Usuario con perfil de enfermero da visto bueno	Rechazar	Rechazar-OK
		Usuario con perfil de jefe hace correcciones al expediente.	Aprobar	Aprobar-OK
		Usuario con perfil de enfermero hace correcciones al expediente.	Rechazar	Rechazar-OK
16	Permite a todos los usuarios hacer consultas de paciente que ya fueron dados de alta.	Consulta de datos de paciente que ya fue dado de alta	Aprobar	Aprobar-OK
		Consulta de datos de paciente que no ha sido dado de alta	Rechazar	Rechazar-OK
17	Mostrar el expediente clínico en formato pdf, con los registros de todos los involucrados y firmas digitales.	Mostrar expediente clínico con todas las firmas digitales de los usuarios que han modificado el expediente	Mostrar registro	Mostrar registro -OK
		Mostrar expediente clínico de paciente que no ha sido modificado	Rechazar	Rechazar-OK
18	Permitir modificar el expediente hasta cierto horario y después de este únicamente es usado para consulta.	Modificar un expediente a las 23:00hrs	Aprobar	Aprobar-OK
		Modificar un expediente a las 00:00hrs	Rechazar	Rechazar-OK
		Modificar un expediente a las 04:00hrs	Rechazar	Rechazar-OK
		Consultar un expediente a las 05:00hrs	Aprobar	Aprobar-OK

Figura 4.4.5 Matriz de pruebas unitarias.



Mantenimiento

“La modificación de un producto de software después de haber sido entregado (a los usuarios o clientes) con el fin de corregir defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio en el entorno”²⁴.

El servicio de mantenimiento de software, es una de las principales actividades en el proceso de ingeniería de software. Su principal función es mejorar y optimizar un software después de una entrega, así como la corrección de los defectos detectados posteriormente a la implementación del software.

Existen varios tipos de mantenimiento a continuación descritos.

- Mantenimiento preventivo.

Es el proceso por el cual se mejora y optimiza el software que se ha instalado, este mantenimiento se realiza para la prevención de posibles problemas que puedan llegar a surgir. Se realiza de manera periódica en todos los módulos, esto influye en el desempeño fiable del sistema, en la integridad de los datos almacenados y en un intercambio de información correcta a la máxima velocidad posible dentro de la configuración óptima del sistema.

- Mantenimiento correctivo.

Este mantenimiento tiene por objetivo localizar y eliminar los posibles defectos de los programas en todo el sistema. Estos defectos se pueden detectar cuando el comportamiento del sistema es diferente al de la especificación. Algunas de las fallas más comunes en los sistemas son:

- Procesamiento: salidas incorrectas de un programa.

²⁴ Estándar IEEE 1219 [IEEE, 1993]



Capítulo 4. Diseño y construcción del sistema



- Rendimiento: tiempo de respuesta demasiado alto en una búsqueda de información.
- Programación: inconsistencias en el diseño de un programa.
- Documentación: inconsistencias entre la funcionalidad de un programa y el manual de usuario.

- Mantenimiento perfectivo.

El mantenimiento perfectivo se define como el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario, desde algo tan simple como cambiar un formato de impresión hasta la incorporación de un nuevo módulo aplicativo. Este mantenimiento se divide en:

- Mantenimiento de Ampliación: orientado a la incorporación de nuevas funcionalidades.
- Mantenimiento de Eficiencia: que busca la mejora de la eficiencia de ejecución.

- Mantenimiento horizontal.

Consiste en el mantenimiento realizado a la infraestructura de hardware sobre el cual están instaladas las plataformas que soportan la implementación del sistema. Esto puede abarcar mantenimiento en:

- Disco duro
- Memoria RAM
- Procesadores
- Instalaciones eléctricas
- Instalaciones físicas
- Instalación de redes y cableado



Capítulo 4. Diseño y construcción del sistema



- Mantenimiento vertical.

Es el mantenimiento enfocado al software que en conjunto establece la plataforma para la implementación del sistema. Los cambios a la distinta paquetería puede ser con respecto a:

- Cambio a una nueva versión.
- Instalación de parches de la aplicación.
- Incremento de mejoras y nuevas capacidades del software.
- Homologación de versiones.
- Cambio de sistemas base (sistemas operativos).
- Adaptación a cambios de ambiente (por ejemplo, proveedores de internet).

4.5 Generación de reportes y análisis de resultados.

La generación de reportes es una pieza fundamental del sistema ya que permite ver a los usuarios del sistema si se cumplen con las características esperadas. Esto nos permite dar un mejor seguimiento a las necesidades del cliente. El proceso de generación de reportes que podemos guardar en diferentes formatos para su respaldo se inicia a continuación en la figura 4.5.1.

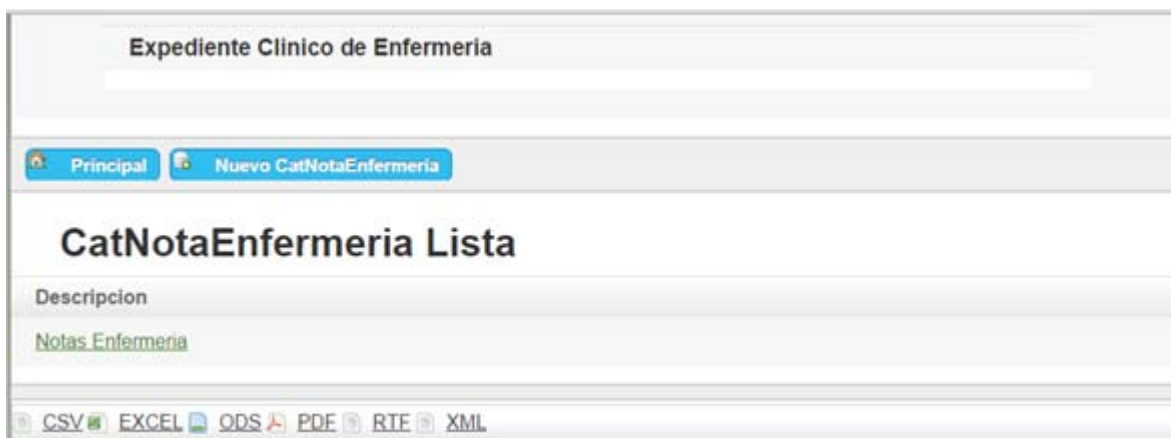


Figura 4.5.1 Notas de enfermería



Capítulo 4. Diseño y construcción del sistema



La siguiente figura 4.5.2 nos muestra la descripción y las notas de enfermería.

Expediente Clínico de Enfermería

Principal Nuevo CatSegmentoNotaEnfermeria

CatSegmentoNotaEnfermeria Lista

Descripcion	Padre
Valoracion Enfermeria	Notas Enfermeria
Diagnostico Intervenciones	Notas Enfermeria
Indicadores Calidad	Notas Enfermeria
Laboratorio Gabinete	Notas Enfermeria

CSV EXCEL QDS PDE RTE XML

Figura 4.5.2 Notas enfermería

El siguiente reporte de la figura 4.5.3 guarda la valoración, indicadores, diagnostico así como la descripción y el tipo.

Principal Nuevo CatRubroNotaEnfermeria

CatRubroNotaEnfermeria Lista

Descripcion	Padre	Tipo	Vista
Valoracion Enfermeria_1	Valoracion Enfermeria	check	Verdadero
Valoracion Enfermeria_2	Valoracion Enfermeria	check	Verdadero
Diagnosticos1	Diagnostico Intervenciones	check	Verdadero
Diagnosticos2	Diagnostico Intervenciones	check	Verdadero
Indicadores1	Indicadores Calidad	radio	Verdadero
Indicadores2	Indicadores Calidad	radio	Verdadero
Promocion Normalidad	Valoracion Enfermeria	check	Falso
Igresos	Valoracion Enfermeria		Falso
Medicamentos	Valoracion Enfermeria		Falso
Escala Glasgow Otros	Valoracion Enfermeria		Falso

1 2 3 Siguiente

CSV EXCEL QDS PDE RTE XML

Figura 4.5.3 Valoración, diagnostico e indicadores de calidad



Capítulo 4. Diseño y construcción del sistema



La imagen de la figura 4.5.4 nos muestra la valoración y la descripción del procedimiento de enfermería así como el estatus en el que está.

Descripcion	Estatus	Padre	Vista
Registro1		Valoracion Enfermeria 1	Verdadero
Registro2		Valoracion Enfermeria 1	Verdadero
Registro3		Valoracion Enfermeria 1	Verdadero
Otro.		Valoracion Enfermeria 1	Verdadero
Registro1		Valoracion Enfermeria 2	Verdadero
Registro2		Valoracion Enfermeria 2	Verdadero
Registro3		Valoracion Enfermeria 2	Verdadero
Otro.		Valoracion Enfermeria 2	Verdadero
Registro1		Diagnosticos1	Verdadero
Registro2		Diagnosticos1	Verdadero

Figura 4.5.4 Valoración y diagnostico

La imagen siguiente Figura 4.5.5 muestra la lista de pacientes e información sobre ellos.

Fechanacimiento	Idtipopaciente	Materno	Nombre	Numeroregistro	Paterno
2000-10-15 00:00:00.0	N	Flores	Luis	N-0000000000000001	Prado
1995-01-05 00:00:00.0	N	Luna	Daniel	N-0000000000000004	Flores
2001-11-20 00:00:00.0	N	Martinez	Rocio	N-0000000000000005	Rios
1990-04-13 00:00:00.0	N	Salgado	Maria	N-0000000000000006	Carrillo
1995-02-21 00:00:00.0	N	Bias	Marcela	N-0000000000000008	Bustamante
1986-12-03 00:00:00.0	N	Lopez	Arturo	N-0000000000000002	Juarez
1980-01-10 00:00:00.0	N	Laguna	Manuel	N-0000000000000007	Barrera
1992-09-15 00:00:00.0	N	Juarez	Carlos	N-0000000000000003	Alvarez
1999-03-30 00:00:00.0	N	Carmona	Carmen	N-0000000000000009	Estrada
1996-05-11 00:00:00.0	N	Gomez	Ignacio	N-0000000000000010	Beltran

Figura 4.5.5 Lista de pacientes



Capítulo 4. Diseño y construcción del sistema



La figura siguiente 4.5.6 muestra información como es la admisión, alergias, comorbilidad, la elaboración y el paciente al que corresponde.

The screenshot shows a web application interface for 'Expediente Clinico de Enfermeria'. It features a navigation bar with 'Principal' and 'Nuevo HojaRegistroEnfermeria' buttons. Below is a table titled 'HojaRegistroEnfermeria Lista' with columns for Admision, Alergias, Comorbilidad, Fecha Elaboracion, Otros, and Paciente. The table contains five rows of data. At the bottom, there are icons for exporting data to CSV, EXCEL, ODS, PDE, RTE, and XML.

Admision	Alergias	Comorbilidad	Fecha Elaboracion	Otros	Paciente
1		00000	2015-07-05 00:00:00 UTC		(N-000000000000) Prado Flores Luis
1	zzzzz	11000	2015-07-06 00:00:00 UTC	cxxxxxx	(N-000000000000) Prado Flores Luis
1		00000	2015-07-20 00:00:00 UTC		(N-000000000000) Prado Flores Luis
1		00000	2015-07-26 00:00:00 UTC		(N-000000000000) Prado Flores Luis
1	ninguna	11100	2015-08-02 00:00:00 UTC		(N-000000000000) Prado Flores Luis

Figura 4.5.6 Hoja de registro enfermería

Análisis de resultados

El sistema genera una hoja de registro clínico en formato pdf en el que hace un llenado automático con los datos del paciente lo que permite agilizar tiempos y poder realizar un mejor trato al paciente y poder ayudar a su diagnóstico y recuperación.

Como podemos ver el sistema cumplió con los requerimientos del cliente por lo que podemos decir que se cumplió con los resultados esperados.



Capítulo 4. Diseño y construcción del sistema



**DIRECCION QUIRURGICA
SUBDIRECCION DE ENFERMERIA
HOJA DE REGISTROS CLINICOS DE ENFERMERIA**

NOMBRE Prado Flores Luis		FECHA NAC 15/10/2000	EDAD 16	SEXO H	CAMA 1
REGISTRO N-0000000000		SERVICIO SERVICIO1	DIAGNOSTICO DIAGNOSTICO1		
DIAS DE ESTANCIA 0		FECHA 02/08/2015			

LABORATORIO Y GABINETE	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7
S																								
I																								
N																								
G																								
S																								
V																								
I																								
T																								
A																								
L																								
E																								
S																								

PESO: 80.0Kg. TALLA: 1.8m.

SIN DOLOR 0 1 2 3 4 5 6 7 8 9 10 PEOR DOLOR

INGRESOS	8	9	10	11	12	13	14	TP	15	16	17	18	19	20	TP	21	22	23	24	1	2	3	4	5	6	7	TP	TOTAL	
Soluciones/Hora																													
FxP M FxP V FxP N																													
Medicamento Oral								0.0							0.0													0.0	
Via Oral								0.0							0.0													0.0	
INGRESOS PARCIALES								TM 0.0							TV 0.0												TN 0.0	0.0	

Figura 4.5.7 Hoja de registro clínico parte 1

En la figura anterior vemos que se llena con los datos generales del paciente así como sus signos vitales al ingresar al hospital.

EGRESOS	8	9	10	11	12	13	14	TP	15	16	17	18	19	20	TP	21	22	23	24	1	2	3	4	5	6	7	TP	TOTAL
Soluciones/Hora																												
Diuresis								0.0							0.0													0.0
Cateterismo								0.0							0.0													0.0
Fuga								0.0							0.0													0.0
Evacuacion								0.0							0.0													0.0
Drenajes								0.0							0.0													0.0
Vomito								0.0							0.0													0.0
Sangrado								0.0							0.0													0.0
Sonda Vesical								0.0							0.0													0.0
EGRESOS PARCIALES								TM 0.0							TV 0.0												TN 0.0	0.0
BALANCE TOTAL DE 24 HORAS																												0.0

MEDICAMENTOS	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7
Escala de Glasgow																								
Respuesta Motora																								
Respuesta Ocular																								
Respuesta Verbal																								
Posicion en cama																								
Perimetros																								
Glucosa Capilar																								
Dolor(EVA)																								
Riesgo de Caída																								

Figura 4.5.7 Hoja de registro clínico parte 2



Capítulo 4. Diseño y construcción del sistema

En la figura anterior se muestran los síntomas del paciente y la respuesta a medicamentos.

VALORACION DE ENFERMERIA									
Escala Dolor					Promocion Normalidad				
TURNO		M	V	N	TURNO		M	V	N
0					Dieta General				
1					Dieta Matutino				
2					Dieta Vespertino				
3					Dieta Nocturno				
4					Fecha Instalacion V				
5					Dias V				
6					Calibre V				
7					Fecha Instalacion V2				
8					Dias V2				
9					Calibre V2				
10					Fecha Instalacion S				
					Dias S				
					Material S				
					Calibre S				
					Globo S				
					Diagnostico Matutino				
					Diagnostico Vespertino				
					Diagnostico Nocturno				
					Observaciones Matutino				
					Observaciones Vespertino				
					Observaciones Nocturno				
					Temperatura		45		
					Frecuencia Cardiaca		78		
					Tension Arterial Diastolica		80		
					Tension Arterial Sistolica		80		
					Frecuencia Respiratoria		45		
					Laboratorio Gabinete		xx		
					Faltante Pasar Matutino				
					Faltante Pasar Vespertino				
					Faltante Pasar Nocturno				
					Religion				
REQUISITOS DE DESARROLLO					Valoracion Enfermeria 1				
REQUISITOS DE DESVIACION DE LA SALUD					TURNO		M	V	N
					Registro1				
					Registro2				
					Registro3				
					Otro:				

Figura 4.5.8 Hoja de registro clínico parte 3

En la anterior figura se registra la valoración de enfermería de acuerdo al turno.

DIAGNOSTICOS E INTERVENCIONES DE ENFERMERIA									
Diagnosticos1				Diagnosticos2					
TURNO		M	V	N	TURNO		M	V	N
Registro1					Registro1				
Registro2					Registro2				
Registro3					Registro3				
Otro:					Otro:				
INDICADORES DE CALIDAD									
TM Planeacion del alta			TV Planeacion del alta				TN Planeacion del alta		
OBSERVACIONES									
ENFERMERAS			JEFES				SUPERVISORES		
MATUTINO	VESPERTINO	NOCTURNO	MATUTINO	VESPERTINO	NOCTURNO	MATUTINO	VESPERTINO	NOCTURNO	

Figura 4.5.9 Hoja de registro clínico parte 4

La figura anterior muestra las intervenciones que tuvieron los pacientes en los turnos indicados.

CONCLUSIONES



Conclusiones



- El sistema de Registro Clínico de Enfermería, cumplió con todos los objetivos planteados, proveyendo al usuario de un sistema con una interfaz amigable que simplificara las labores del personal médico y de enfermería.
- El sistema genera un reporte de manera electrónica, permitiendo un ahorro de papel y una mejora en beneficio del ambiente.
- Se redujo significativamente los costos de varios procedimientos. El sistema abatirá los costos de copias, impresiones, carpetas y sobre todo del espacio físico en el que son resguardados los expedientes médicos en la institución.
- El sistema cumple con los requerimientos solicitados por el personal médico y de enfermería, solventando las necesidades en beneficio de la productividad y de la calidad de la atención médica brindada.
- El desarrollo del sistema permitirá a la institución médica dar un paso hacia la automatización y optimización de los procesos, reduciendo costos de operación y mejorando los tiempos en los que se desempeña el trabajo.
- El sistema contribuye a que el personal médico y de enfermería utilicen las tecnologías de la información y comunicación en el desarrollo de sus labores, reduciendo errores humanos y mejorando la imagen de la institución médica.
- En el desarrollo del sistema se aplicaron conocimientos de las asignaturas, de Base de Datos, Administración de Proyectos de Software, Ingeniería de Software, Sistemas Operativos, Lenguajes de Programación, adquiridos durante la formación educativa por parte de la Facultad de Ingeniería.
- Se aplicaron conocimientos de bases de datos y lenguajes de programación adquiridos de manera extracurricular, que en conjunto con la formación académica, contribuyeron al desarrollo del trabajo de tesis.



Conclusiones



- El Programa de Apoyo a la Titulación, es una excelente opción para desarrollar el trabajo de tesis, pues a través de grupos de trabajo, nos crea compromisos para desarrollar un trabajo de calidad.
- El Programa de Apoyo a la Titulación, nos brindó las facilidades para entregar semanalmente un objetivo de la tesis, lo que permitió concluir el trabajo de manera exitosa.
- Los conocimientos adquiridos durante la formación académica por parte de la Facultad de Ingeniería de la UNAM, nos permitieron desarrollar el sistema de manera exitosa.
- La formación impartida por la Facultad de Ingeniería, nos permitió trabajar en equipo, compartiendo nuestros conocimientos y experiencias en el desarrollo del sistema y del trabajo de tesis.
- El desarrollo del proyecto nos permitió aplicar los conocimientos teóricos y técnicos adquiridos durante la formación educativa, así como, de la experiencia profesional, demostrando la calidad de los profesionistas egresados de la Facultad de Ingeniería.
- El desarrollo del sistema nos ha dejado enseñanzas y experiencias sobre el ámbito profesional, que nos permitirá brindar soluciones óptimas y de calidad en las empresas en las que nos desarrollemos profesionalmente.

BIBLIOGRAFÍA

1. Arquitectura Informática, ECURED.
http://www.ecured.cu/index.php/Arquitectura_Inform%C3%A1tica. Consulta 19-07-2015
2. Estándar IEEE 1219 1998. <https://standards.ieee.org/findstds/standard/1219-1998.html>. Consulta 06-08-15.
3. Flor, Julio. La polémica gestión de las historias clínicas. 2015, julio 6, de El médico interactivo.
<http://www.elmedicointeractivo.com/ap1/emiold/informes/gestion/historias.htm>. Consulta 06-08-15.
4. http://www.ecured.cu/index.php/Pruebas_de_software Consultada: 05-08-15
5. <https://laurmolina7821.wordpress.com/1-1-3-aplicaciones-de-2-3-y-n-capas/>
Consulta 21-07-2015
6. Luis Joyanes Aguilar. Ingeniería del software: un enfoque práctico, Mikel Angoar, 1997. P63
7. Manual de procedimientos del servicio de urgencias para hospitales generales, instituto de salud del estado de México.
8. Manual de referencia y documentación del framework Grails
<https://grails.org/documentation.html>. Consulta 06-08-15.
9. Manual de referencia y documentación del framework Symfony
<http://symfony.com/doc/current/index.html>. Consulta 06-08-15.
10. Manual de referencia y documentación del gestor de base de datos PostgreSQL 9.3 <http://www.postgresql.org/docs/9.3/static/index.html>. Consulta 06-08-15.
11. Manual de referencia y documentación del servidor web Jetty
<http://www.eclipse.org/jetty/documentation> Consulta 06-08-15.
12. Manual de referencia y documentación del servidor web Apache Tomcat
<http://tomcat.apache.org/tomcat-8.0-doc/index.html>. Consulta 06-08-15.
13. Manual de referencia y documentación del lenguaje de programación Groovy
<http://www.groovy-lang.org/documentation.html>. Consulta 06-08-15.
14. Muñoz R., Carlos. Cliente Servidor, Universidad de Colima, Sitio web:
http://docente.ucol.mx/rcarlos/public_html/cliente-servidor.htm Consulta 19-07-2015.

15. Neilla Rogelio. Soluciones Transaccionales. Sitio web:
http://www.oocities.org/ar/r_niella/Document/t_marco.htm Consulta 20-07-2015.
16. Norma oficial mexicana nom-004-ssa3-2012, del expediente clínico, Secretaria de Salud, México, 2012
17. Norma oficial mexicana nom-016-ssa3-2012, que establece las características mínimas de infraestructura y equipamiento de hospitales y consultorios de atención médica especializada, Secretaria de Salud y Consejo de Salubridad General, México, 2012.
18. Selman-Housein Abdo, Guía de acción para la excelencia en la atención médica, Científico Técnica, 2002.
19. Unidad Nacional de Educación Abierta y a Distancia. Colombia. Lección 2: Tipos de arquitecturas cliente-servidor.
http://datateca.unad.edu.co/contenidos/208017/ContLin/leccin_2_tipos_de_arquitecturas_clienteservidor.html Consulta 21-07-2015