

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO
“SISTEMA WEB DE SEGUIMIENTO Y
CONTROL DE PACIENTES INTERNOS”**

CASO: HOSPITAL ARCO IRIS

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: WENDY MILENKA QUIROGA RADA

**TUTOR METODOLÓGICO: M. Sc. ALDO RAMIRO VALDEZ
ALVARADO**

ASESOR: Lic. RAMIRO FLORES ROJAS

LA PAZ – BOLIVIA

2014



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

PROYECTO DE GRADO: SISTEMA WEB DE SEGUIMIENTO Y CONTROL DE
PACIENTES INTERNOS
“CASO: HOSPITAL ARCO IRIS”

Presentada por : Univ. Wendy Milenka Quiroga Rada

Para optar el grado académico de: **Licenciatura en Informática**

Mención : Ingeniería en Sistemas Informáticos

Nota numeral:

Nota Lineal :

Director de la Carrera de Informática: M. Sc. Edgar Clavijo Cárdenas

Tutor : M. Sc. Aldo Ramiro Valdez Alvarado

Asesor : Lic. Ramiro Flores Rojas

Tribunal :

Dedicatoria:

A mi papito Orlando que desde el cielo me dio toda la fortaleza necesaria para salir adelante, a mi madre Rosemary por todo su apoyo incondicional y a mis hijos Nicole y Dandiel por darme la fuerza para seguir adelante.

Con mucho cariño

Wendy Quiroga

Agradecimientos

Primeramente a nuestro Señor por iluminar siempre mi camino y darme la familia tan linda que tengo.

Agradecer a mi mamá por darme todo su apoyo y comprensión incondicional para que salga adelante y siempre confió en mí, a mi papito Orlando que desde el cielo me cuida y protege siempre, a mi querida hermana por acompañarme en todo momento, a mis lindos hijos Nicole y Dandiel por estar ahí siempre conmigo y a toda mi linda familia que están siempre para mí.

Al M. Sc. Aldo Ramiro Valdez Alvarado. Docente tutor, por sus consejos y por guiarme a lo largo del desarrollo de este proyecto.

Al Lic. Ramiro Flores Rojas, docente revisor, por brindarme su apoyo incondicional, tiempo dedicado y por su paciencia, que me ayudaron a culminar mi proyecto.

A la Universidad Mayor de San Andrés por acogerme en todo el tiempo que me tomo lograr esta gran meta.

A los docentes de la carrera de Informática.

Agradecer también a la Dirección de Sistemas del Hospital Arco Iris, especialmente a la Lic. Ana María Rada Contreras por darme la oportunidad de realizar este proyecto.

A mis amigos de la universidad por la linda amistad que me brindaron.

RESUMEN

El presente proyecto fue desarrollado en el Hospital “Arco Iris” en el sector de la Unidad de Cuidados Intensivos Neonatales (UCIN), que realizan el cuidado de los neonatos ayudándoles así a poder registrar virtualmente las historias clínicas.

En dicha unidad no cuenta con la sistematización de los historiales clínicos de los neonatos es decir todo lo realizan manualmente lo que ocasiona volúmenes de papelería con información.

Por lo mencionado anteriormente se desarrolló un Sistema Web de Seguimiento y Control de Pacientes Internos Neonatales optimizando el trabajo en el tiempo de procesos y además llevar un control adecuado de la información.

El producto obtenido cuenta con características requeridas por los usuarios, resultando una herramienta de ayuda para los procesos que se efectúan en la Unidad de Cuidados Intensivos Neonatales.

Este proyecto fue realizado con la metodología de desarrollo de software del Proceso Unificado Ágil denotado por el acrónimo AUP para el análisis y diseño del sistema, para el modelado del sistema se utilizó la propuesta de Ingeniería Web basado en UML UWE.

El departamento de Sistemas del Hospital Arco Iris trabaja sobre la plataforma Windows, es por este motivo que el sistema fue desarrollado con PHP, como lenguaje de programación se utilizó JavaScript y como gestor de base de datos se utilizó MySQL.

La calidad del sistema fue medida con la norma ISO 9126 con todas las características necesarias, también se efectuó una descripción de las amenazas del sistema y la aplicación de medidas de seguridad para el funcionamiento adecuado del sistema.

ÍNDICE GENERAL

1. MARCO REFERENCIAL	1
1.1. INTRODUCCIÓN	1
1.2. Antecedentes	2
1.2.1. Antecedentes Institucionales	2
1.2.2. Proyectos Similares	4
1.3. Planteamiento del Problema	6
1.3.1. Problema Central	6
1.3.2. Problemas Secundarios	6
1.4. Definición De Objetivos	7
1.4.1. Objetivo General	7
1.4.2. Objetivos Específicos	7
1.5. Justificación	8
1.5.1. Justificación Económica	8
1.5.2. Justificación Social	8
1.5.3. Justificación Tecnológica	9
1.6. Alcances y Limites	9
1.6.1. Alcances	9
1.6.2. Limites	10
1.7. Aportes	10
1.7.1. Aporte Práctico	10
1.7.2. Aporte Teórico	10
1.8. Metodología	11
2. MARCO TEÓRICO	13
2.1. Ingeniería de Software	13
2.1.1. Ciclo de vida del Software	13

2.2. Metodología Ágil AUP	14
2.2.1. Ciclo de Vida del Proceso Unificado Ágil (AUP)	15
2.2.2. Fases de AUP	16
2.2.3. Disciplinas de AUP	17
2.2.4. Disciplina del Modelado	18
2.2.5. Disciplina de la Implementación	23
2.2.6. Disciplina de Pruebas	25
2.2.7. Disciplina de Despliegue	28
2.2.8. Disciplina de Administración de la Configuración	32
2.2.9. Disciplina de Administración del Proyecto	33
2.2.10. Disciplina del Entorno	36
2.3. Ingeniería Web	38
2.3.1. Metodología de Modelado UWE	39
2.3.2. Modelado De La Uwe	40
2.3.3. Fases De La Uwe	42
2.4. Lenguaje de Código PHP	43
2.5. Sistema de Administración de Base de Datos MYSQL	44
2.6. Lenguaje HTML5	44
3. MARCO APLICATIVO	46
3.1. Introducción	46
3.2. Fase De Inicio	46
3.2.1. Modelado Del Negocio	46
3.2.2. Modelado De Casos De Uso Del Negocio	46
3.2.3. Modelado De Requerimientos	48
3.3. Fase De Elaboración	50
3.3.1. Modelado De Análisis	50
3.3.2. Modelo De Caso De Uso	50
3.3.3. Diagrama De Caos De Uso De Alto Nivel	50
3.3.4. Descripción De Casos De Uso	51
3.3.5. Diagrama De Paquetes	60
3.3.6. Diagrama de clases	61

3.3.7. Modelo de navegación -----	64
3.3.8. Modelo de navegación de menús -----	64
3.4. Fase De Construcción-----	64
3.4.1. Diseño De Interfaces -----	65
3.5. Fase De Transición -----	77
3.6. Políticas De Implementación -----	77
3.6.1. Corte y Cambio-----	78
3.6.2. Operación Paralela-----	78
3.6.3. Operación Piloto -----	78
3.6.4. Operación De Fase-----	79
3.6.5. Aplicación del método corte y cambio-----	79
4. CALIDAD DE SOFTWARE Y SEGURIDAD -----	80
4.1. Introducción -----	80
4.2. Características Propuestas por ISO-9126 -----	80
4.2.1. Confiabilidad -----	83
4.2.2. Funcionalidad-----	85
4.2.3. Mantenibilidad -----	93
4.2.4. Portabilidad -----	95
4.2.5. Usabilidad-----	96
4.3. Seguridad de Software -----	98
4.3.1. Amenazas.-----	98
4.3.2. Guías de seguridad-----	99
4.3.3. Tipos de seguridad para sistemas web . -----	99
5. ANÁLISIS DE COSTO/BENEFICIO-----	101
5.1. Introducción -----	101
5.2. Análisis De Costos -----	101
5.2.1. Costo Del Software Desarrollado-----	101
5.2.2. Costo De La Implementación Del Proyecto -----	106
5.2.3. Costo De Elaboración Del Proyecto -----	106

5.2.4. Costo Total-----	107
5.3. Análisis de Beneficios -----	108
5.4. Valor Neto Actual-----	108
5.5. Tasa interna de retorno -----	110
6. CONCLUSIONES Y RECOMENDACIONES -----	112
6.1. Conclusiones -----	112

BIBLIOGRAFÍA

ANEXO

ANEXO A - ÁRBOL DE PROBLEMAS

ANEXO B – ÁRBOL DE OBJETIVOS

ANEXO C - HISTORIA CLÍNICA DEL RECIÉN NACIDO

ANEXO D – HOJA DE ENFERMERÍA

ANEXO E – CONTROL DE ENFERMERÍA DE NEONATOLOGÍA

ANEXO F – CENTRO DE ENFERMERÍA UCIN - UCIP

ANEXO G – CUIDADOS INTERMEDIOS NEONATOLOGÍA

ANEXO H – SALA DE CRECIMIENTO Y DESARROLLO

ANEXO I – U.T.I. NEONATAL

DOCUMENTACIÓN

1. MARCO REFERENCIAL

1.1. INTRODUCCIÓN

De un tiempo a esta parte las aplicaciones de sistemas de información tienen su origen en casi todas las áreas de una empresa o institución y están relacionadas con todos los problemas de la organización de documentos, información y otros datos importantes.

Uno de los principios fundamentales en el desarrollo de los sistemas web en las instituciones, es que las aplicaciones son herramientas y no instrumentos que debe tenerse para utilizar la tecnología de la información; en consecuencia los sistemas web deben desarrollarse sobre la base de su propia capacidad para mejorar el desempeño de la organización. La modernización de los servicios archivísticos, adquiere una singular importancia, en virtud del carácter estratégico que estos servicios poseen con relación a la actividad regular de las unidades responsables que integran.

Hoy en día se puede ver que en Bolivia en la ciudad de La Paz no se tienen en los hospitales sistemas que ayudan a automatizar la información ya sea de los pacientes, tratamientos, medicamentos, seguimiento y demás. Ya sea por varios factores como económicos, sociales y hasta tecnológicos, por que como se ve no hay mucho apoyo en el sector de salud de parte de todas las autoridades viendo que esta es una principal necesidad ya que todos necesitamos de estos servicios.

Existe un sector el cual es muy preocupante hoy en día en el país que se trata de los recién nacidos prematuros, ya que en un estudio reciente se vio que en Bolivia el porcentaje de muertes en los hospitales por esta causa es alta ya sea por factores de malformación, hemorragias, asfixias, mal definidos, infecciones y otros. En el departamento de La Paz no se tienen muchos hospitales que pueden atender a este tipo de pacientes ya que al menos un 10 por ciento de los recién nacidos requiere de cuidados intensivos, tener cuidado con la limpieza por las infecciones, alimentación y tratamiento de los mismos.

Con este fin se desarrolla el presente proyecto de grado que se refiere a un sistema web para la Unidad de Cuidados Intensivos Neonatales del Hospital Arco Iris de la ciudad de La Paz, y así con este facilitar al personal de dicha unidad el manejo de expedientes clínicos así como sus altas y bajas médicas, laboratorios, archivos de farmacia, asignaciones y control de tratamientos entre otros.

Todo esto siempre cuidando en manejo de documentos médicos, respetando leyes, normas y reglamentos desarrollados por los médicos, autoridades y todo el personal de salud.

Y así poder manejar de una manera más segura la información de estos pequeños que son vulnerables, además de no saber hablar, caminar, mucho menos defenderse, y que debe luchar para poder sobrevivir porque nació con poco peso, porque se infectó o porque tuvo un nacimiento prematuro.

1.2. Antecedentes

1.2.1. Antecedentes Institucionales

Después de dar una breve introducción, viendo los principales postulados y la relación con la medicina y los pacientes veamos algunos antecedentes.

Una parte esencial para el ejercicio profesional del personal médico, es la historia clínica la cual actúa como un recordatorio para el manejo clínico del paciente. Antiguamente se podía observar que en los hospitales los médicos atendían individualmente todas las necesidades de los pacientes y las historias clínicas eran como un cuaderno de notas donde se registraban solo los datos más importantes según su criterio, en la actualidad no parece ser muy distinta la forma de elaborar dichas notas ya que estas ahora son manuales.

En los últimos años hubo muchos avances tecnologías en cuanto se refiere a los sistemas de información clínica, entre estos se tiene desde una simple agenda medica hasta sistemas informáticos médicos complejos.

La historia clínica informatizada es un documento electrónico que contiene los datos relacionados con el paciente, el cual tiene validez como un documento médico legal donde se registrara toda la relación del personal médico con el paciente, los actos y actividades realizadas con él y su salud esto con el fin de facilitar su asistencia.

El Hospital Arco Iris, pertenece a la Fundación "Arco Iris", fue fundado el 27 de septiembre de 2001 y abrió sus puertas a la atención en noviembre del mismo año. Fue construido por iniciativa del R.P. José Neunhofer con la finalidad de romper las barreras de exclusión económica y social permitiendo la atención médica a la población en general, principalmente niños y niñas de y en la calle y personas de extrema pobreza hasta los 18 años de edad de las ciudades de La Paz y El Alto.

La misión de la Unidad de Cuidados Intensivos Neonatales (U.C.I.N.) es atender a pacientes recién nacidos con alto riesgo perinatal con calidad, equidad, calidez para lograr una recuperación adecuada de los pacientes y egresarlos con una calidad de vida que les permita integrarse a su núcleo familiar y posteriormente a la sociedad sin limitaciones.

El personal médico a cargo de la U.C.I.N. realiza diagnósticos a partir de la captura de las medidas morfológicas internas y externas del feto, utilizando ultrasonido obstétrico y almacenando imágenes, videos, histogramas, gráficas de crecimiento, y otros datos importantes en el seguimiento perinatal.

Cuando se encuentran frente a un nuevo caso de malformación fetal, los asistentes médicos, registran mediante formularios en papel, información del paciente tales como datos personales, antecedentes patológicos, quirúrgicos, farmacológicos, ginecólogos, entre otros; para posteriormente, almacenarla en una hoja donde se centralizan los datos para su análisis estadístico.

Los médicos suscritos a la U.C.I.N. realizan un informe detallado de cada anomalía acontecida, con motivo de socializar dichos casos que fueron encontrados durante la prestación del servicio, el cual se expone ante la Junta de Diagnóstico Prenatal de

Malformaciones Fetales y Anomalías Congénitas, quienes se reúnen semanalmente debido al alto número de anomalías presentadas y toman decisiones sobre las pacientes.

Actualmente no existe ningún sistema de información computarizado en la U.C.I.N. de carácter administrativo que contribuya a la automatización de los procesos que se realizan a diario en dicha entidad, es por esto, que se reconoce la necesidad de desarrollar una herramienta basada en la web, que permita al personal de la U.C.I.N. el acceso a información relacionada a casos anómalos, con la cual se puedan generar posteriormente nuevas investigaciones.

El personal de la U.C.I.N. se ha mostrado entusiasta por la proposición del presente proyecto y están dispuestos a colaborar en todo sentido durante el proceso de diseño, análisis y desarrollo; aportando de su tiempo y abriéndome las puertas para el presente proyecto para poder observar y tener una visión clara de las necesidades que se presentan.

El uso de herramientas de desarrollo libres tales como IDE's; manejadores de bases de datos, y demás lenguajes robustos enfocados a la web, representan un ahorro sustancial en el pago de licencias de uso, convirtiéndose en un factor importante en la viabilidad del presente proyecto, además de permitir la futura actualización y elaboración de nuevos módulos al producto final que será el sistema de información a desarrollar.

El personal de la U.C.I.N. será ampliamente beneficiado, debido a que podrán utilizar el sistema web para soportar su trabajo, y realizar sus procesos eficientemente, puesto que pasarían de trabajar con un sistema manual a un sistema web computarizado.

1.2.2. Proyectos Similares

En cuanto a proyectos similares al Sistema Web de Seguimiento y Control de Pacientes Internos, podemos nombrar los siguientes ya que nos serán de mucha utilidad como apoyo:

- ✓ “Sistema de Seguimiento Clínico” Hospital del Niño”Dr. Ovidio Aliaga Uría”, dicho proyecto realizado por el Univ. Rodrigo Gutiérrez Miranda en la gestión 2005.

Este proyecto desarrolla un sistema que cubre de mejor forma el funcionamiento, el control adecuado, eficiente y correcto de la información referente a los pacientes, así como sus estadísticas para el beneficio del Hospital del Niño.

Para esto se desarrolló un sistema acorde a las necesidades del departamento de estadística, ayudando a la toma de decisiones de manera oportuna y eficaz, usando la metodología RAD (Desarrollo rápido de aplicación).

- ✓ “Sistema de Seguimiento y Control de Pacientes”, Hospital Villa Dolores, realizado por el Univ. Boris Condori Bustos en la gestión 2005.

Este sistema permite el mejor acceso y la manipulación de la información de los pacientes que genera la institución, aplicando la metodología orientada a objetos.

También se utilizó la herramienta UML (Unified Modeling Language).

- ✓ “Portal Web para el Hospital de la Mujer” elaborada por la Univ. María Eugenia Choque Tarqui, en la gestión 2008.

Este portal brinda apoyo en la gestión y organización de la información, permitiéndoles acceder a la información del hospital en cualquier momento y desde cualquier lugar.

En este portal se utilizaron las herramientas tecnológicas C Sharp para la programación, IIS 6 como servidor de aplicaciones Web y SQL Server 2005 como gestor de datos. Tales proyectos deben renovarse de una manera más eficaz con nuevos medios tecnológicos y así ponerlos más acorde con la actualidad.

1.3. Planteamiento del Problema

La mayoría de las instituciones sanitarias diariamente realizan como lo habíamos mencionado sus procesos de almacenamiento de la información de los pacientes manualmente en papel, sin el uso de medios tecnológicos.

1.3.1. Problema Central

En la actualidad la mayoría de los hospitales es común el manejo manual de la historia clínica en soporte de papel, que atrasan el proceso inmediato de la gran cantidad de datos, y eso implica la deficiencia en el control y seguimiento, transferencia y asignación; volviendo lenta la atención medica la cual tiene como consecuencia la pérdida de tiempo y en este caso la molestia de los padres del paciente neonato.

De aquí nace nuestra pregunta:

¿Cómo se puede mejorar el control de pacientes internos en la unidad de cuidados intensivos neonatales del Hospital Arco Iris?

1.3.2. Problemas Secundarios

Se puede ver que existen muchos más problemas recurrentes en los hospitales, estos serán mostrados en el árbol de problemas. Como los más relevantes tenemos:

- ✓ **LA INFORMACIÓN DE LOS PACIENTES NEONATOS AL SER REDACTADA MANUALMENTE EN ALGUNOS CASOS SE ENCUENTRA EN UN LENGUAJE INCOMPENSIBLE EN LA HISTORIA CLÍNICA, LO CUAL TRAE DIFICULTAD AL LEER LA MISMA QUE PUEDE SER MAL INTERPRETADA.**
- ✓ Se puede tropezar con la existencia de múltiples expedientes clínicos referentes a un mismo paciente, teniendo un volumen masivo papeles y estos ocupan espacio físico.

- ✓ La transferencia y recuperación de la información del paciente de un expediente clínico se realiza con lentitud, lo que ocasionara pérdida de tiempo en la atención al paciente.
- ✓ Asignación deficiente y tradicional de reserva de turnos, todo esto en papel que no permite almacenar y procesar la gran cantidad de datos de los pacientes.
- ✓ El traslado continuo del expediente clínico, generando riesgos de dispersión, intrincado, deterioro y pérdida de información de los pacientes.

1.4. Definición De Objetivos

1.4.1. Objetivo General

Desarrollar un sistema web de seguimiento y control de pacientes internos para la Unidad de Cuidados Intensivos Neonatales del Hospital Arco Iris que contribuirá a la transición de la historia clínica tradicional, que usa soportes de papel hacia su sustitución con soportes digitales que nos permita almacenar y procesar gran cantidad de datos para el control y seguimiento.

1.4.2. Objetivos Específicos

- Proceder a la recopilación de datos, mediante el uso del teclado de su ordenador para la gestión centralizada y uso del archivo único, eliminando el soporte del papel ayudando a la mejor comprensión del historial.
- Centralizar los expedientes clínicos en un archivo único y en medios ópticos, solucionando las limitaciones de los recursos hardware para almacenar dicha información y evitar el crecimiento voluminoso de papel almacenado en espacios físicos.
- Facilitar el acceso rápido a la historia clínica para el diagnóstico y tratamiento oportuno de los bebés, siempre cumpliendo las características de seguridad y exactitud, respetando al derecho de intimidad personal y familiar del paciente.

- Elaborar la documentación que sirva como referencia para el usuario sobre el uso del Sistema de Información.
- Brindar a la U.C.I.N. las historias clínicas informatizadas, en forma ordenada, mediante el uso del sistema web, que será el resultado de la interacción del médico con el paciente teniendo así no tener que tropezar con riesgos de dispersión o pérdida de la información de los pacientes.
- Diseñar una base de datos y las interfaces de usuario con base en el análisis de requisitos de software y las especificaciones de diseño.
- Elaborar pruebas de funcionalidad y carga pico a la aplicación web.

1.5. Justificación

1.5.1. Justificación Económica

Desde mucho antes del uso de las computadoras en las organizaciones, éstas reunían, almacenaban y actualizaban información en el transcurso normal de su diario desempeño.

Es por esto que se hace visible la necesidad de implementar un Sistema de Información para el seguimiento y control del historial clínico, desde la perspectiva de la eficiencia, para brindar apoyo en los procesos diarios que se realizan en la U.C.I.N. del Hospital Arco Iris, y así de esta manera ahorrar tiempo en busca de los historiales así acelerar la atención a los pacientes, como en los materiales que se usan para administrar los historiales de los pacientes.

1.5.2. Justificación Social

Un sistema web de seguimiento y control de pacientes, permitirá que el hospital optimice las principales tareas con los pacientes neonatos, mejorando el tiempo de servicio por el sistema que será implementado en red, permitiendo al personal realizar cualquier consulta del paciente de una forma más rápida y segura.

La facilidad de consulta demandada por el personal médico beneficiara tanto al paciente como a sus familiares, el mismo permitirá búsquedas de información rápida y oportuna al momento que se la requiera.

El sistema que controla el historial del paciente proporciona información rápida y oportuna particularmente al personal médico del U.C.I.N. permitiendo así actuar con más precisión y exactitud en los tratamientos, laboratorios y demás.

1.5.3. Justificación Tecnológica

El proyecto a desarrollar, se realiza por la necesidad que tiene la el hospital Arco Iris, ya que no cuenta con un buen control sistematizado de los historiales de sus pacientes, optimizando así los servicios que presta dicho hospital.

De esta manera lograremos sistematizar toda la información de los pacientes ya que tendremos un sistema web que mejorara los tiempos tanto como en la generalización de tiempo de los reportes, registros como en la modificación y búsqueda de datos.

1.6. Alcances y Limites

1.6.1. Alcances

El siguiente proyecto de grado a desarrollar, pretende desarrollar un Sistema Web de Seguimiento y Control de Pacientes Internos, que nos permitirá:

- Registro de los datos del paciente así como el registro de ingreso, datos de la internación, asignación de infraestructura, seguimiento del paciente, registro de medicamentos administrados y el egreso del paciente, esto a cargo de los doctores.
- Desechando medios digitales y desechando las historias en papel que ocupaban lugares físicos ayudando a los doctores como al personal del U.C.I.N. a la facilidad de acceder a los historiales.

- Tener mayor control y no tener limitas para expresar todo lo que se refiere al paciente.
- Consultas, estas nos permitirán ver, manejar toda el historial clínico del paciente.

1.6.2. Limites

En el presente proyecto de grado no se tomara en cuenta

- La parte administrativa, pues no se registrara a los doctores ni a las personas que trabajan en la U.C.I.N.
- La parte contable ya que estas están fuera del alcance del sistema, pues este solo se centrara en el paciente, y la interacción con los doctores.

1.7. Aportes

1.7.1. Aporte Práctico

El Sistema Web de Seguimiento y Control de Pacientes Internos beneficiar al Hospital Arco Iris con el rápido acceso y la buena manipulación de la información de los pacientes del U.C.I.N. así también:

- La tenencia de mayor control sobre los historiales.
- Proporcionar información del paciente de la forma más clara y no redundante ayudando al mejor manejo del historial del paciente.
- Cubriendo las falencias que existen actualmente en el U.C.I.N.

1.7.2. Aporte Teórico

Para el desarrollo del mismo trabajaremos con aplicación de la metodología aup, es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de

negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas.

Y para el modelado trabajaremos con proceso del desarrollo UWE para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web.

UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML. Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

1.8. Metodología

Para desarrollar el siguiente proyecto de grado vemos en la parte de ingeniería a la metodología aup:

- ✓ Descomposición de un proyecto grande en mini-proyectos
- ✓ Cada mini-proyecto es una iteración
- ✓ Las iteraciones deben estar controladas
- ✓ Cada iteración trata un conjunto de casos de uso

Las ventajas que tendremos con esta serán:

- ✓ Detección temprana de riesgos
- ✓ Administración adecuada del cambio
- ✓ Mayor grado de reutilización
- ✓ Mayor experiencia para el grupo de desarrollo

Y en la arquitectura:

- ✓ Arquitectura en software
 - Diferentes vistas del sistema: estructural, funcional, dinámico, etc.
 - Plataforma en la que va a operar
 - Determina la forma del sistema
- ✓ Arquitectura: determina la forma del sistema
- ✓ Casos de uso: determinan la función del sistema

En la parte de la técnica trabajaremos con uwe:

- En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web.
- Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo navegacional, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad
- UWE define vistas especiales representadas gráficamente por diagramas en UML.
- UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

2. MARCO TEÓRICO

2.1. Ingeniería de Software

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos.

Esta disciplina trasciende la actividad de programación, que es el pilar fundamental a la hora de crear una aplicación. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema.

2.1.1. Ciclo de vida del Software

Cabe destacar que el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro etapas: concepción, elaboración, construcción y transición.

La concepción fija el alcance del proyecto y desarrolla el modelo de negocio; la elaboración define el plan del proyecto, detalla las características y fundamenta la arquitectura; la construcción es el desarrollo del producto; y la transición es la transferencia del producto terminado a los usuarios.

Una vez que se completa este ciclo, entra en juego el mantenimiento del software. Se trata de una fase de esta ingeniería donde se solucionan los errores descubiertos (muchas veces advertidos por los propios usuarios) y se incorporan actualizaciones para hacer frente a los nuevos requisitos. El proceso de mantenimiento incorpora además nuevos desarrollos, para permitir que el software pueda cumplir con una mayor cantidad de tareas.

Un campo directamente relacionado con la ingeniería de software es la arquitectura de sistemas, que consiste en determinar y esquematizar la estructura general del proyecto, diagramando su esqueleto con un grado relativamente alto de especificidad y señalando los distintos componentes que serán necesarios para llevar a cabo el desarrollo, tales como aplicaciones complementarias y bases de datos. Se trata de un punto fundamental del proceso, y es muchas veces la clave del éxito de un producto informático.

Los avances tecnológicos y su repercusión en la vida social han afectado inevitablemente el proceso de desarrollo de software por diversos motivos, como ser el acceso indiscriminado de los usuarios a cierta información que hasta hace un par de décadas desconocía por completo y que no pueden comprender, dado que no poseen el grado de conocimiento técnico necesario.

2.2. Metodología Ágil AUP

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas, Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad.

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.

2.2.1. Ciclo de Vida del Proceso Unificado Ágil (AUP)

La naturaleza serial en aup es capturada en cuatro fases: iniciación, elaboración, construcción y transición.

Las disciplinas son ejecutadas en una manera iterativa, definiendo las actividades las cuales los miembros del equipo ejecutan para construir, validar y liberar software funcional que cumpla con las necesidades de sus involucrados. las disciplinas son: modelado, implementación, pruebas, despliegue, administración de la configuración, administración del proyecto y entorno.

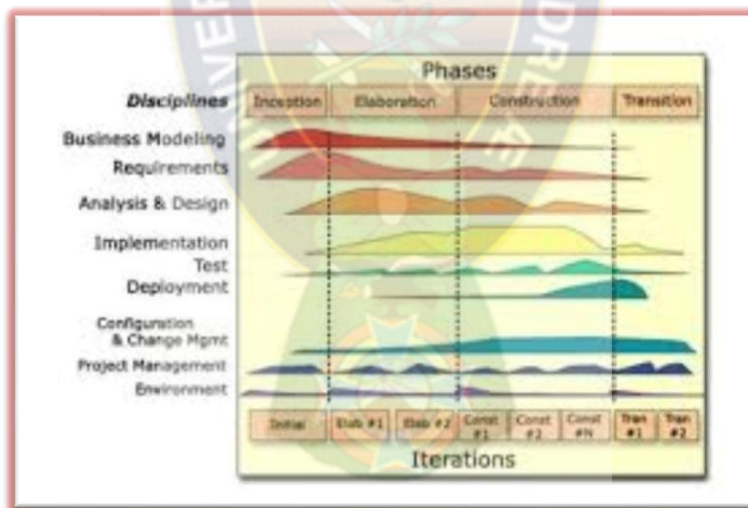


Figura 2.1: Fases y disciplinas del AUP
[Fuente: Ambler, S. 2005]

2.2.2. Fases de AUP

El Proceso Unificado Ágil consta de cuatro fases que el proyecto atraviesa de forma secuencial. Dichas fases son, al igual que en el Proceso Unificado de Rational:

Fases	Metas	Hitos
1. Inicio	Identificar el alcance inicial de proyecto, una arquitectura inicial del sistema y obtener un presupuesto inicial del proyecto y una aceptación de los involucrados.	Objetivos de Ciclo de Vida(LCO, por su siglas en inglés)
2. Elaboración	Probar arquitectura del sistema.	Arquitectura del Ciclo de Vida (LCA, por su siglas en inglés)
3. Construcción	Construir un software funcional sobre una base regular e incremental, las cuales cumplan con las prioridades más importantes para los involucrados o usuarios del proyecto.	Capacidad Operacional Inicial (IOC, por su siglas en inglés)
4. Transición	Validar y desplegar el sistema en su ambiente de la producción.	Liberación del Producto (PR, por su siglas en inglés)

Tabla 2.1: Fases del AUP
[Fuente: Ambler, S. 2005]

2.2.3. Disciplinas de AUP

A lo largo de las cuatro fases, se desarrollan actividades relativas a siete disciplinas de manera iterativa:

Disciplina	Descripción
Modelado	La meta de ésta disciplina es entender el negocio de la organización, el dominio del problema que el proyecto aborda e identificar una solución viable para abordar el dominio del problema.
Implementación	La meta de ésta disciplina es transformar su modelo(s) en un código ejecutable y realizar una prueba de nivel básico en una unidad particular de prueba.
Pruebas	La meta de ésta disciplina es ejecutar una evaluación de los objetivos para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema función como fue diseñado y verificar que los requerimientos están completos.
Despliegue	La meta de ésta disciplina es planificar la entrega del sistema y ejecutar el plan para que el sistema esté disponible para los usuarios finales.
Administración de la Configuración	La meta de ésta disciplina es administrar el acceso a los entregables o productos del proyecto. Esto incluye no sólo el rastreo de versiones del producto en el tiempo, sino que también incluye controlar y administra los cambios que ocurran.
Administración del Proyecto	La meta de ésta disciplina es dirigir las actividades que se llevan a cabo en el proyecto. Esto incluye administración del riesgo, la

	dirección de personas (asignar tareas, seguimiento de los procesos, etc.), y coordinar con los sistemas y personas fuera del alcance del proyecto para que el este termine a tiempo y dentro del presupuesto.
Entorno	La meta de ésta disciplina es apoyar el resto de los esfuerzos por garantizar que, el proceso adecuado, la orientación (normas y directrices) y herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario.

Tabla 2.2: Disciplinas del AUP
[Fuente: Ambler, S. 2005]

2.2.4. Disciplina del Modelado

Su objeto es entender la lógica de negocio de la aplicación, el dominio del problema del proyecto e identificar una solución viable para el dominio del problema.

A continuación lo veremos fase por fase:

Fases	Actividades
Inicio	<p>Inicial, modelado de requerimientos de alto nivel (ver Figura 1). Los interesados deben participar activamente en el modelo de requerimientos de alto nivel el cual define el alcance inicial para el proyecto y proporciona suficiente información para una estimación aproximada. Debería considerar:</p> <ul style="list-style-type: none"> • Explore el uso de casos de uso. • Identifique los procesos del negocio para la creación de diagramas de flujo de datos (DFDs). • Identifique las entidades principales del negocio y sus relaciones

trabajando con modelos de dominio livianos.

- Identifique las principales reglas del negocio y requerimientos técnicos. Por ahora, el nombre de entidades, reglas y requerimientos técnicos del negocio que son suficientes (puede obtener los detalles a través del modelo de lluvia de ideas durante la fase de Construcción).
- Inicie el desarrollo de un glosario que describa términos importantes técnicos y del negocio.
- La comprensión de la estructura política dentro de su comunidad de partes interesadas a través del modelado de la organización.
- El tratamiento de requisitos como una lista priorizada que evolucione en el tiempo (esto apoya un verdadero soporte de cambio de administración, no prevención de cambios). Casos de uso, reglas del negocio, y requerimientos técnicos pertenecen a la pila.

Inicial, modelado de requerimientos de alto nivel (ver Figura 1). Su objetivo principal es identificar una estrategia de arquitectura viable, entrada crítica dentro del plan del proyecto tanto como en el esfuerzo de implementación. La mejor forma de trabajo es poner personal técnico, incluyendo algunos sino todos los desarrolladores, juntos en un lugar para desarrollar una estrategia de arquitectura que se discute en las pizarras creando diagramas de estilo libre quizás algún tipo de forma inicial de modelo de despliegue.

Elaboración	Identificar riesgos técnicos. Sus necesidades de trabajo, en particular, sus casos de uso y requerimientos técnicos, ponen de manifiesto los posibles riesgos técnicos a su proyecto. Estos riesgos pueden incluir la introducción de nuevas tecnologías a su organización, un nuevo uso de las
-------------	---

tecnologías existentes, importante carga o estrés en su aplicación o sistemas actuales externos. La máxima prioridad debe ser abordar los riesgos por su esfuerzo de implementación en el desarrollo de un extremo a extremo del esqueleto del sistema.

Modelado de la Arquitectura. Como usted construye el prototipo de la arquitectura necesitará modelar por lluvia de ideas algunos detalles para pensar pedazos de la arquitectura.

Prototipado de interfaces de usuario. De forma paralela al desarrollo del prototipo de la arquitectura también debe considerar Prototipado de interfaces de usuario de varias de las principales pantallas. Usted no querrá hacer demasiado, porque sus necesidades de prototipos se pueden cambiar y, por tanto, su trabajo tendrá que ser desechado. Tu objetivo en el momento debe de entender las principales pantallas / páginas de su interfaz de usuario, en el entendimiento de que va a cambiar durante la Construcción, e identificar la "apariencia" básica del sistema.

Construcción Análisis de modelo de lluvia de ideas. Durante las iteraciones de la Construcción deberá trabajar cerca de sus interesados del proyecto para entender sus necesidades en una base de just-in-time (JIT). Aspectos importantes:

Participación activa de interesados y modelado inclusivo que usan técnicas y herramientas simples que son críticas para su negocio.

Si lo desea, puede profundizar en los detalles de sus casos de uso, quizás visualmente utilizando diagramas de flujo o diagramas de actividad UML en vez de descripciones de texto.

Explore las reglas del negocio y los requerimientos técnicos en la misma

forma.

Puede que necesite hacer interfaces de activos legados tales como sistemas actuales o una base de datos. El Análisis actual puede ser difícil y trabajo "políticamente cargado.

En lugar de las descripciones de casos de uso, de reglas del negocio y de requerimientos técnicos, usted puede encontrar más efectivo simplemente escribir casos de prueba de aceptación. Esto le permite acercarse a una única fuente de información porque no es necesario para capturar el requisito tanto en un documento de requerimientos como en una descripción de prueba.

Debido a que la interfaz de usuario es el sistema a muchas de sus partes interesadas, es muy probable que descubra que prefieren centrarse en las pantallas e informes en lugar de mirar el desarrollo de otros productos de su trabajo, a fin de estar preparados para el prototipo.

Mantenga su glosario del proyecto actualizado si lo tiene.

Diseño por modelo de lluvia de ideas. Durante las iteraciones de la Construcción su objetivo es hacer sólo lo suficiente para modelar para pensar en el diseño de un simple requerimiento, o sólo una parte, antes de la implementación del requerimiento. Modeladores ágiles modelan directo con sus usuarios, no simplemente pasar los modelos a los mismos, y, a menudo, asumen el papel de promotor. Es probable que desee crear:

Diagrama de secuencia de UML. Este diagrama representa la lógica dinámica dentro del código fuente. Son parte de su modelo de proyecto y usualmente se tiran lejos hasta que se tiene una buena herramienta CASE con compatibilidad para ingeniería inversa. Las pizarras son geniales

herramientas para crear nuevos diagramas.

Modelo de despliegue. Típicamente crear algún ordenamiento del diagrama de resumen representado en la arquitectura del sistema de despliegue/ red.

Diagramas de estilo libre. Comúnmente creados en pizarras y luego son borrados cuando no se necesitan. Diagramas de estilo libre son la técnica más común de dibujado.

Documentos de resumen del sistema típicamente incluyen diagramas de estilo libre.

Diagrama de Clases UML. Si va a hacer algún diagrama de clases use una herramienta de modelado que le permita generar código fuente. Su diagrama de clases debe estar basada en un modelo de dominio (si existe)

Modelo de amenazas de seguridad. Si problemas de seguridad preocupan, entonces, debe considerar modelarlas para ayudarle a pensar generar las amenazas potenciales así también como tratarlas.

Modelo físico de datos. Este es probablemente el modelo de diseño más importante, que debería considerar el uso de una herramienta CASE para desarrollar y mantener en el tiempo, sobre todo una herramienta que genera DDL código. Es posible tomar un enfoque ágil para el modelado de datos.

Documento crítico decisiones de diseño. A como toma decisiones de diseño debe considerar grabar que son se ven a simple vista, o que usted crea que alguien en el futuro realmente le gustaría conocer, también inicie su documentación de resumen del sistema..

Transición Modelado de lluvia de ideas. Necesitará hacer algún modelado en el momento (Just In Time) para tratar de entender las causas principales de un defecto.

Finalice la documentación de resumen del sistema. El mejor momento para finalizar su documentación general del sistema es durante de la fase cuando el alcance de su sistema está realmente establecida. Realice su diseño crítico de decisiones, si lo documenta en la Construcción, como una base desde la cual se construye este documento. Cualquier otra información importante que quiera en este documento es un resumen del alcance del sistema y de los diagramas de arquitectura críticos (ahora es cuando se debería de colocar todos los diagramas de estilo libre y bocetos de pizarras en un buen diagrama usando una herramienta de dibujo).

Tabla 2.3: Disciplina del modelado
[Fuente: Ambler, S. 2005]

2.2.5. Disciplina de la Implementación

Esta se encarga de transformar los modelos en código ejecutable y realizar pruebas básicas, en particular pruebas unitarias. Fase por fase tendremos:

Fases	Actividades
Iniciación	<p>Prototipado técnico. Es posible que tenga que "picar" un pequeño aspecto de un requisito con el fin de entender lo suficiente, lo que le permite estimar el esfuerzo requerido. Estos prototipos son típicamente pequeños. "tire" las piezas de código.</p> <p>Prototipado de Interfaces de Usuario. Para más usuarios las interfaces de</p>

usuario (UI) -- pantallas, reportes, y manuales-- en el sistema. Cuando las UI son potencialmente complejas, o cuando los usuarios quieren ver qué es lo que van a obtener antes de comprarlo, debería considerar prototipar, al menos, las pantallas principales. El prototipo de UI, usualmente arroja en este punto, lo que será usado para convencer al usuario para que entienda las necesidades del usuario (que se explorará como parte del esfuerzo del modelado).

Elaboración Probar la arquitectura. Las actividades críticas dentro de la fase de Elaboración es identificar la arquitectura potencial y luego probar que esta arquitectura funcione a través del desarrollo de la arquitectura del prototipo extremo a extremo para su sistema, y a la vez mitigando gran parte de los riesgos técnicos en su proyecto. Los prototipos técnicos como son la calidad de producción de código que constituye el fundamento o "skeleton" de su sistema.

Construcción Primeras pruebas. Obtenga un acercamiento de la base del Desarrollo Dirigido por Pruebas (TDD) para todos los aspectos de la aplicación.

Construya constantemente. Creaciones diarias son un buen comienzo, pero idealmente usted quiera construir su sistema cada vez que el código fuente cambie. Automatice esto usando un producto como Cruise Control el cual monitorea su sistema de control de versiones para cambios a su código y regeneración a como las necesite.

Evolución de la lógica de dominio. Implemente su lógica del negocio in sus clases de negocio/dominio.

Evolucionar las interfaces de usuario. La interface de usuario es el sistema para la mayoría de usuarios. Esfuércese por hacer su software tan usable

como sea posible siguiendo las estrategias de diseño de interfaces de usuario y usabilidad.

Evolucionar el esquema de datos. Su esquema de datos debe ir evolucionado en conjunto con su dominio y código de interfaces de usuario. Necesitará re factorizar su base de datos justo como cuando re factoriza cualquier otro tipo de código.

Desarrollo de interfaces de activos existentes. Necesitará acceder frecuentemente a la funcionalidad dentro de los sistemas existente. Esto se puede hacer a través de una variedad de medios, incluyendo una interfaz de servicios web, un C-API, procedimientos almacenados, y así sucesivamente. El análisis actual es a menudo una parte importante de sus esfuerzos de modelado.

Generar el script de conversión de datos. Deberá a menudo acceder a las fuentes de datos actuales. Estas fuentes de datos actuales suelen ser objeto de diseño y / o problemas de calidad y como resultado se deberá crear un script de conversión de datos para poner los campos de datos en un formato utilizable.

Transición	Corregir defectos. Concéntrese en la corrección de defectos encontrados como resultado de las pruebas.
-------------------	--

Tabla 2.4: Fases de la Disciplina de la Implementación

[Fuente: Ambler, S. 2005]

2.2.6. Disciplina de Pruebas

Realizar una evaluación de los objetivos para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema funciona como fue diseñado y verificar que los requisitos se cumplen.

Fases	Actividades
Iniciación	<p>Planificación inicial de pruebas. Deben ser a muy alto nivel al principio. El objetivo principal es identificar cuántas pruebas necesita hacer, quien será el responsable de hacerlas, el nivel de participación requerido por los usuarios, y los tipos de herramientas y los entornos necesarios (un punto de la disciplina de Entorno).</p> <p>Exanimación inicial de los productos de trabajo del proyecto. Hacia el final de esta fase el plan de proyecto, la visión y mucho más deben estar disponibles. Este producto es a menudo examinado, típicamente como parte del hito de revisión, por los involucrados del proyecto.</p> <p>Exanimación inicial de modelos. A un alto nivel, el modelo de requerimientos inicial, y quizás un modelo de arquitectura inicial, deben ser producidos por su esfuerzo de modelado. Usted puede optar por la revisión de este trabajo con las partes involucradas, particularmente si quiere comunicar el alcance y la arquitectura potencial de su sistema a un gran rango de personas que están activamente envueltos en el desarrollo de los modelos.</p>
Elaboración	<p>Validación de la Arquitectura. Usted debe tomar un enfoque de desarrollo controlado por pruebas (TDD) para construir su prototipo técnico el cual compruebe la arquitectura de su sistema. Un aspecto importante de hito de revisión es la validación de la arquitectura, que podría ser algo tan sencillo como presentar una visión general de la arquitectura y los resultados de sus esfuerzos de los prototipos para los interesados. O, podría ser tan complejo como revisión formal de todo su trabajo durante esta fase.</p> <p>Evoluciones su modelo de pruebas. Su equipo deberá desarrollar un</p>

paquete de pruebas de regresión, compuesta por la unidad de pruebas de su desarrollo controlado por pruebas (TDD) en la implementación, su aceptación de pruebas de su modelado, y las pruebas de su sistema (p.e. funcionamiento, integración, carga, .. pruebas). Usted también necesita mantener la trazabilidad entre los requerimientos, las pruebas y su código fuente que muestran como ha validado los requerimientos de implementación. En este punto su reporte de defectos será simplemente la salida de su paquete de pruebas.

Construcción Pruebas de software. Además de las unidades de prueba de los desarrolladores deberá hacer pruebas de instalación del script de despliegue o liberación, sistema de pruebas de esfuerzos tales como la carga / pruebas de tensión y las pruebas de función, y sus pruebas de aceptación de usuario. Debido a que su sistema evoluciona a través de sus proyectos, su paquete de pruebas también lo hará. Lo más común es que promueva su código en un ambiente de pruebas de pre-producción, lo mejor de la fase de Transición serán las actividades de prueba.

Evolucione su modelo de pruebas. Ver arriba.

Transición Validación del sistema. Usted se concentrará en las "grandes pruebas" de actividades tales como las del sistema, las de integración y las de aceptación, y las pruebas piloto/beta. Su objetivo es probar completamente el sistema dentro del ambiente de pruebas de pre-producción.

Validación de la documentación. Su documentación de sistema (vista general del sistema, usuarios, soporte, y documentación de operaciones), y sus materiales de capacitación necesitará validarlos. Todo esto puede ser hecho por medio de las revisiones o mejor aún como parte de sus pruebas piloto/betas.

Analice su modelo de pruebas. Va a tener que seguir ejecutando de paquete de pruebas de regresión y actualizarlo tanto como necesite, hasta que su sistema esté listo para ser desplegado en producción. Su reporte de defectos será aproximadamente más formal, los defectos detectados serán registrados, junto con los detalles para que los desarrolladores puedan corregirlos.

Tabla 2.5: Fases de la Disciplina de la Pruebas
[Fuente: Ambler, S. 2005]

2.2.7. Disciplina de Despliegue

Planear la entrega del sistema y ejecutar el plan para hacer que el sistema quede disponible para los usuarios finales.

Fases	Actividades
Iniciación	<p>Identificar el rango liberación potencial. Por ejemplo, se le solicita entregar su sistema antes de que finalice el año, pero después de la entrega de otra aplicación que se tiene planeada poner en producción al final de septiembre. Por lo tanto su rango de liberación es entre el 1° de octubre y el 31 de diciembre, pero usted decide ser conservador y establecerlo entre el 1° de noviembre y el 15 de diciembre, tomando en cuenta problemas de implementación del otro proyecto y los días festivos del final de diciembre. Definir el rango de liberación potencial en las primeras fases del proyecto, lo ayudará en los esfuerzos de planeación.</p> <p>Comience con un plan de entregables de alto nivel. Este esfuerzo se debe enfocar en planificar los entregables de su sistema, identificando el rango de liberación potencial. Su principal objetivo debe ser determinar una estrategia general de implementación: basada en entender del proyecto si</p>

¿tiene sentido liberar el software todo de una sola vez o con fases de entregables? Porque aún no se ha finalizado la estrategia de la arquitectura, se debe decidir comenzar a planear la implementación durante la fase de Elaboración.

Elaboración Actualizar el plan de implementación. Una parte importante de definir la arquitectura es definir la configuración de los entregables del sistema, tal vez se pueda soportar una configuración cliente/servidor de tres capas para usuarios internos conectados a su propia red, una interfaz basada en HTML para usarla en Internet, y una versión "stand-alone" usuarios únicos para quienes no tengan conexión. Individualmente cada configuración de entregable puede ser documentada como parte del modelo de despliegue que define cómo están organizados los componentes del software actual y de qué manera cómo están los componentes de hardware. Entender la configuración ayuda a identificar los diferentes tipos de instalación que se deben realizar, y a la vez ofrece una imagen global de proceso de implementación.

Construcción Desarrollar el script de instalación. Como desarrolla el sistema debe también escribir y probar el script de instalación necesario para entregarlo en la pre producción del ambiente de pruebas. Estos scripts deben estar escritos de manera que se puedan reconfigurar fácilmente para ponerlo en producción.

Desarrollar notas del entregable. Sus notas de entregables deben resumir los avances que posee el entregable actual del sistema que actualmente está construyendo.

Desarrollar documentación inicial. Además de entregar software funcional, también se debe entregar la documentación del sistema

(operaciones, soporte, visión general, y la documentación al usuario), así como su material de formación. Si se dispone a crear nuevos documentos, entonces puede al menos comenzar a definir su estructura y tomar notas de la misma. Usted no debe dedicar mucho tiempo a la documentación por el momento, porque su sistema está en desarrollo.

Actualice su plan. De manera cómo progresa el desarrollo del sistema, debe avanzar su plan de implementación. Es probable que deba renegociar el plan de implementación con sus departamentos de operación y soporte, así como con otros proyectos que también se encuentren en el plan de implementación, de modo que el proyecto encaje nuevamente en el plan general de la organización.

Implementar el sistema en entornos de pre-producción. Debe entregar regularmente el sistema en un ambiente de pre-producción para efectuar pruebas y llevar a cabo el control de calidad necesario, así como realizar demostraciones a los involucrados. Cuanta mayor experiencia se obtenga más fácilmente se efectuará la puesta en producción.

Transición

Concluir el proceso de implementación. Para concluir este proceso debe definir una línea base de entrega como referencia, las actividades de la administración de la configuración, y realizar una "última" revisión de software, así como la implementación del flujo de trabajo.

Finalizar la documentación. La mayor parte de la documentación del sistema (operaciones, soporte, visión general, y la documentación al usuario) es generalmente realizada durante esta etapa, debido a que la funcionalidad del sistema se estabiliza en este momento.

Anunciar la implementación. Se debe anunciar el calendario de

implementación de manera anticipada e incluyendo las fechas estimadas de capacitación e instalación. Debe instruir también su equipo de operación, soporte y la comunidad de usuarios según proceda en esta fase.

Capacitar el personal. Capacitar los clientes o usuarios de su proyecto, así como a la administración, equipo de operaciones y equipo de soporte, es siempre parte importante de la implementación. Tome en cuenta que los usuarios pueden necesitar capacitación más allá que aprender a trabajar con su aplicación. Por ejemplo, puede ser que sea la primera vez que algunos usuarios trabajen con una computadora, o con el navegador de internet, entre otros. Del mismo modo, esta puede ser la primera vez que el personal de operaciones trabaje con una nueva tecnología, tal como un servidor de aplicaciones EJB, y por lo tanto se debe capacitar en las nuevas tecnologías para trabajar en su proyecto.

Puesta en producción. En este punto se debe realizar cualquier conversión o migración de datos, y puede ser todo de una vez, un trabajo por lotes o una conversión gradual de los datos, conforme lo requieran los usuarios. Usted puede decidir ejecutar su nuevo sistema en paralelo con su actual sistema varias semanas, para asegurarse de que funciona realmente en producción. También puede optar por implementar el sistema a un subconjunto de su comunidad de usuarios, llamada liberación piloto, para verificar que funciona para un grupo pequeño antes de que "imponga" su sistema a todos. También puede ser necesario entregar una versión del software al departamento de soporte, de manera que este departamento pueda simular los problemas de producción cuando los usuarios los contacten por ayuda.

Tabla 2.6: Fases de la Disciplina de Despliegue
[Fuente: Ambler , S. 2005]

2.2.8. Disciplina de Administración de la Configuración

Gestionar el acceso a los artefactos del proyecto. Esto incluye, además de la traza de versiones de los artefactos, el control de cambios y la gestión de los mismos.

Fases	Actividades
Iniciación	<p>Establecer la configuración del ambiente. Usted tiene que hacer varias:</p> <ul style="list-style-type: none">• La estructura de directorios apropiada, la cual debe seguir los lineamientos corporativos, necesita ser creada para el equipo del proyecto.• Los miembros del equipo del proyecto necesitan tener acceso al folder o directorios del proyecto y cualquier otro software cliente instalados en sus ordenadores• Los miembros del equipo del proyecto también necesitan ser entrenados con los conceptos básicos de CM y las herramientas necesarios.• Su repositorio de CM necesitará ser instalado si este aún esta. <p>Ponga todos los productos del trabajo bajo el control de CM. Cada uno debe poner su trabajo bajo el control de CM en una base regular, verificar las entradas y salidas en cada caso.</p>
Elaboración	Poner todos productos del proyecto sobre el control de CM. Ver Arriba.
Construcción	Poner todos productos del proyecto sobre el control de CM. Ver Arriba.
Transición	Poner todos productos del proyecto sobre el control de CM. Ver Arriba.

Tabla 2.7: Fases de la Administración de la Configuración
[Fuente: Ambler , S. 2005]

2.2.9. Disciplina de Administración del Proyecto

Dirige las actividades que tienen lugar dentro del proyecto, incluyendo gestión de riesgos, dirección del personal y coordinación.

Fases	Actividades
Iniciación	<p>Inicie conformando el equipo. En este punto necesitará alguien con habilidades de modelado para trabajar con los usuarios para identificar los requerimientos iniciales del sistema y con las personas técnicas para identificar una arquitectura potencial.</p> <p>Crear relaciones con sus involucrados del proyecto. El soporte a los usuarios y la participación es crítica para su éxito.</p> <p>Determinar. El proyecto debe ser financiera, técnica, operacional, y políticamente viable. En otras palabras, debe tener sentido desde el punto de vista del negocio, debe ser capaz de construirlo, mantenerlo en funcionamiento una vez que lo ponga en producción, y su organización debe ser capaz de apoyar el proyecto.</p> <p>Desarrollar un cronograma de alto nivel para todo el proyecto. El cronograma del proyecto debe mostrar su proyecto organizado en iteraciones, indicar descripciones de los principales hitos, tareas con dependencias críticas, tanto para su equipo de trabajo como para otros equipos, y su fecha de finalización prevista.</p> <p>Desarrollar un plan detallado de iteración para la siguiente iteración. La planificación detallada se realiza basada en el principio justo a tiempo. Cuando hay eventos importantes, tales como una revisión o una tarea</p>

crítica de alguna dependencia del proyecto, en una próxima iteración el director del proyecto debería trabajar con las personas afectadas antes de tiempo para planificar las actividades adecuadas. Con respecto a la planificación detallada para una iteración, la mejor forma de hacerlo es simplemente agrupar el equipo en el inicio de una iteración y trabajar con ellos para planificar su trabajo: la persona que va a hacer el trabajo es a menudo la mejor adaptada para planearlo.

Manejo del riesgo. Siempre hay riesgos en un proyecto de desarrollo de software: técnicos y organizacionales. A lo largo de la ejecución del proyecto usted deseara identificar estos riesgos y desarrollar estrategias para enfrentarlos, resumiendo los esfuerzos con una lista de riesgos.

Obtener financiamiento y apoyo de los involucrados. Tendrá que demostrar que usted entiende el alcance, que el proyecto es viable, que pueda entender y abordar los riesgos, y que tiene un plan viable para proceder. Usted debe trabajar con los involucrados antes de la revisión de los entregables, o de lo contrario correr el riesgo de falta de aceptación.

Cerrar esta fase. Debe ejecutar la Revisión de los entregables de los objetivos del ciclo de vida(OCV), cuya finalidad principal es hacer que los involucrados formalicen el proyecto.

Elaboración Construya el equipo. Conforme su proyecto tome forma y crezca, necesitará agregar miembros al equipo. Durante esta fase necesitará personas con habilidades de análisis, desarrollo e implementación. Seguramente tendrá que capacitar a su equipo en las nuevas habilidades de desarrollo.

Proteger el equipo. Las políticas de empresa son una realidad y un buen

administrador de proyectos protegen a sus equipos lo mejor posible.

Obtener recursos. Su equipo necesita financiación, instalaciones (por ejemplo salas y cubículos), hardware, software, y así sucesivamente para hacer su trabajo.

Manejo del riesgo. Continúe los esfuerzos de administración del riesgo.

Actualice el plan del proyecto. Continúe las actividades planeadas tal y como las describió.

Cerrar esta fase. Tendrá que ejecutar la revisión del ciclo de vida de la arquitectura (CVA), cuya principal finalidad es demostrar que su arquitectura funciona y que se está enfrentando correctamente los principales riesgos del proyecto.

Construcción Administre el equipo. Continúe desarrollando el equipo (desarrolladores y equipo de pruebas tanto como se necesite), manténgase protegiéndolos y proveyéndoles los recursos que necesitan.

Actualizar su plan de proyecto. Durante la fase de construcción necesitará asegurar que tiene identificadas las principales dependencias involucradas en el desarrollo exitoso de su sistema. Debe considerar las necesidades de sus equipos de operación y soporte, capacitación del usuario final, y el plan de pruebas al sistema piloto.

Cerrar esta fase. Tendrá que ejecutar la revisión de la capacidad operativa inicial (COI), cuya principal finalidad es demostrar que su equipo ha desarrollado un sistema que está potencialmente listo para implementarse en producción.

Transición Administrar el equipo. Debe incluir en el equipo de desarrolladores, encargados de pruebas e implementadores.

Cerrar esta fase. Tendrá que ejecutar la revisión de los productos entregables (PE), cuya principal finalidad es demostrar que su sistema ha pasado las pruebas y es aceptable para los involucrados.

Tabla 2.8: Fases de la Administración del Proyecto
[Fuente: Ambler , S. 2005]

2.2.10. Disciplina del Entorno

Apoyar el resto del esfuerzo asegurando que los procesos, métodos y herramientas están disponibles para el equipo cuando los necesitan.

Fases	Actividades
Iniciación	<p>Configure el entorno de trabajo. Esta será una tarea permanente ya que hay gente que se añade al equipo en el tiempo.</p> <p>Identifique la categoría del proyecto. Muchas organizaciones desarrollan varias versiones de sus procesos de software, por ejemplo uno para equipos pequeños, uno para remplazar sistemas legales, otro para sistemas de plataforma comercial, etc.</p>
Elaboración	<p>Evolucionar el entorno de trabajo. Su proyecto progresa a medida que se entiende la evolución de los requisitos, la estrategia de la arquitectura, y su enfoque general. El resultado es que necesitará evolucionar su entorno instalando nuevas herramientas, o remover las herramientas que ya no necesita.</p> <p>Ajuste de los procesos de materiales. Se debe ajustar AUP para cumplir</p>

con las necesidades del equipo. Esto puede incluir materiales del proceso de AUP (por ejemplo esta página), se debe elegir entre escribir un documento corto que no se quiere hacer, o se debe simplificar eligiendo la cosa correcta en el momento correcto.

Construcción Apoyar al equipo. Miembros del equipo del proyecto necesita ayuda para utilizar y / o la configuración de diversas herramientas para satisfacer sus necesidades. También se necesita ayuda para elegir las plantillas de la documentación adecuada y seguir la guía de su empresa.

Evolucionar el entorno de trabajo. Ver arriba.

Establecer el ambiente de capacitaciones. A medida que progrese en el plan de despliegue o liberación se debe descubrir que se necesita entrenar al usuario, personal de soporte y el personal de operación. Este esfuerzo de capacitación debe requerir espacios de entrenamiento y versiones de entrenamiento del sistema disponibles, frecuentemente en la fase de Transición. Se puede iniciar estableciendo estos pilares del entorno y finalizar en la fase de Construcción.

Transición Configuración de las operaciones y soporte de los entornos. Personal de soporte, y algunas veces personal de operación, frecuentemente se necesita una versión del sistema configurada que se use para simular reportes de defectos en una forma segura.

Recobrar licencias de software. A medida que su proyecto llega a la conclusión puede ser necesario desinstalar las licencias de software los equipos que ya no necesitan el software para que las licencias puedan estar disponibles a los demás dentro de su organización.

Tabla 2.9: Fases de la Disciplina del Entorno
[Fuente: Ambler , S. 2005]

2.3. Ingeniería Web

Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Una posible sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

En términos sencillos, un servicio web es cualquier sistema de software diseñado para soportar interacción máquina a máquina sobre una red.

Esta amplia definición abarca múltiples y diferentes sistemas, pero en general "servicio web" suele referir a clientes y servidores que se comunican usando mensajes XML que siguen el estándar SOAP.

En definitiva, permite comunicación entre diferentes máquinas, con diferentes plataformas y entre programas distintos. Esta comunicación se logra a través de la adaptación de diversos estándares abiertos.

Las especificaciones que definen los servicios web son intencionalmente modulares y esto resulta en que no hay ni un sólo documento que las contenga a todas ellas. Tampoco no hay ni un solo conjunto estable de especificaciones.

2.3.1. Metodología de Modelado UWE

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización . UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos, con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo navegacional, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad.

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

La propuesta de Ingeniería Web basada en UML (UWE (Koch, 2000)) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

2.3.2. Modelado De La Uwe

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de Escenarios Web.

2.3.2.1. Modelo De Lógico-Conceptual

El modelo de lógico-conceptual de la UWE apunta a construir un modelo conceptual de una aplicación Web, procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web. La construcción de este modelo lógico-conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web.

2.3.2.2. Modelo De Navegación

Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. El primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionaran.

2.3.2.3. Modelo De Presentación

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

2.3.2.4. Iteración Temporal

Presenta los objetos que participan en la interacción y la secuencia de los mensajes enviados entre ellos.

2.3.2.5. Escenarios Web

Permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. Junto con el modelo de interacción temporal, los escenarios Web

proveen la representación funcional dinámica del modelo de navegación.

2.3.2.6. Diagramas

Los diagramas usados por UWE, son diagramas UML puro. Entre los más importantes tenemos: Diagramas de estado, de Secuencia, de colaboración y diagramas de Actividad.

2.3.3. Fases De La Uwe

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

2.3.3.1. Captura, Análisis Y Especificación De Requisitos

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipo de la interfaz de usuario.

2.3.3.2. Diseño Del Sistema

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

2.3.3.3. Codificación Del Software

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

2.3.3.4. Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

2.3.3.5. La Instalación O Fase De Implementación

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

2.3.3.6. El Mantenimiento

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

2.4. Lenguaje de Código PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML5.

Lo que distingue a PHP del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML5 y enviándolo al cliente.

El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP.

Lo mejor de utilizar PHP es su extrema simplicidad, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas.

2.5. Sistema de Administración de Base de Datos MYSQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

2.6. Lenguaje HTML5

HTML5 significa HyperText Markup Language, versión 5 es un lenguaje en el que se escriben las páginas Web, que permite hacer enlaces (hipertexto) y que es leído por los navegadores para poder mostrar la información en la pantalla del ordenador de una forma correcta. Por ejemplo, el lenguaje HTML5 indica al navegador cuando un texto tiene un

enlace y también la dirección URL a la que tiene que dirigirse si se hace click sobre el link. Los lenguajes humanos son algo vivo que evoluciona a lo largo del tiempo y al HTML le pasa lo mismo, según ha ido evolucionando Internet se ha ido modificando. Por ello el HTML5 que es la última fase de desarrollo comparte muchas características con HTML4, pero tiene algunas cosas nuevas de gran importancia.

Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto. En este sentido, el HTML5 no se diferencia demasiado de su predecesor. La diferencia principal, sin embargo, es el nivel de sofisticación del código que podremos construir usando HTML5.



3. MARCO APLICATIVO

3.1. Introducción

En el presente capítulo se define el marco del trabajo y las tareas necesarias para desarrollar y construir el Sistema web de seguimiento y control de pacientes internos neonatos ya que se concluyó con toda la investigación de la información preliminar.

Para desarrollar dicho sistema emplearemos todo lo mencionado y nos regiremos en la metodología de desarrollo AUP y siguiendo las fases del modelo UWE

3.2. Fase De Inicio

Para comenzar la fase inicial del proyecto empezaremos a comprender el problema, el entendimiento de los requerimientos, identificar si existen ciertos riesgos relacionados con el proyecto, los cuales nos permitan poder modelar la arquitectura del software.

3.2.1. Modelado Del Negocio

En el modelado de negocios comprenderemos mucho mejor los procesos con los que trabajan los doctores hacia la atención con los pacientes neonatos mostrando así narrativamente los eventos que lo describiremos gráficamente y nos servirán como punto de partida para derivar el modelo de casos de uso.

3.2.2. Modelado De Casos De Uso Del Negocio

En el siguiente diagrama de caso de uso del negocio se observa las principales operaciones que realizan en la Unidad de Cuidados Intensivos Neonatales (UCIN), comenzando con el ingreso de datos de la madre y del recién nacido dándole así al neonato un número de registro del historial clínico, seguidamente el diagnóstico general realizado por el Medico, este para el llenado del historial clínico, de acuerdo a este se verá si es necesaria la internación, y también esta se clasificara por la gravedad de salud.

Al internarse el paciente será actualizado constantemente su historial clínico. Todo este proceso hasta que se le dé el alta al paciente neonato.

En el diagnóstico se verán involucrados todos estos actores.

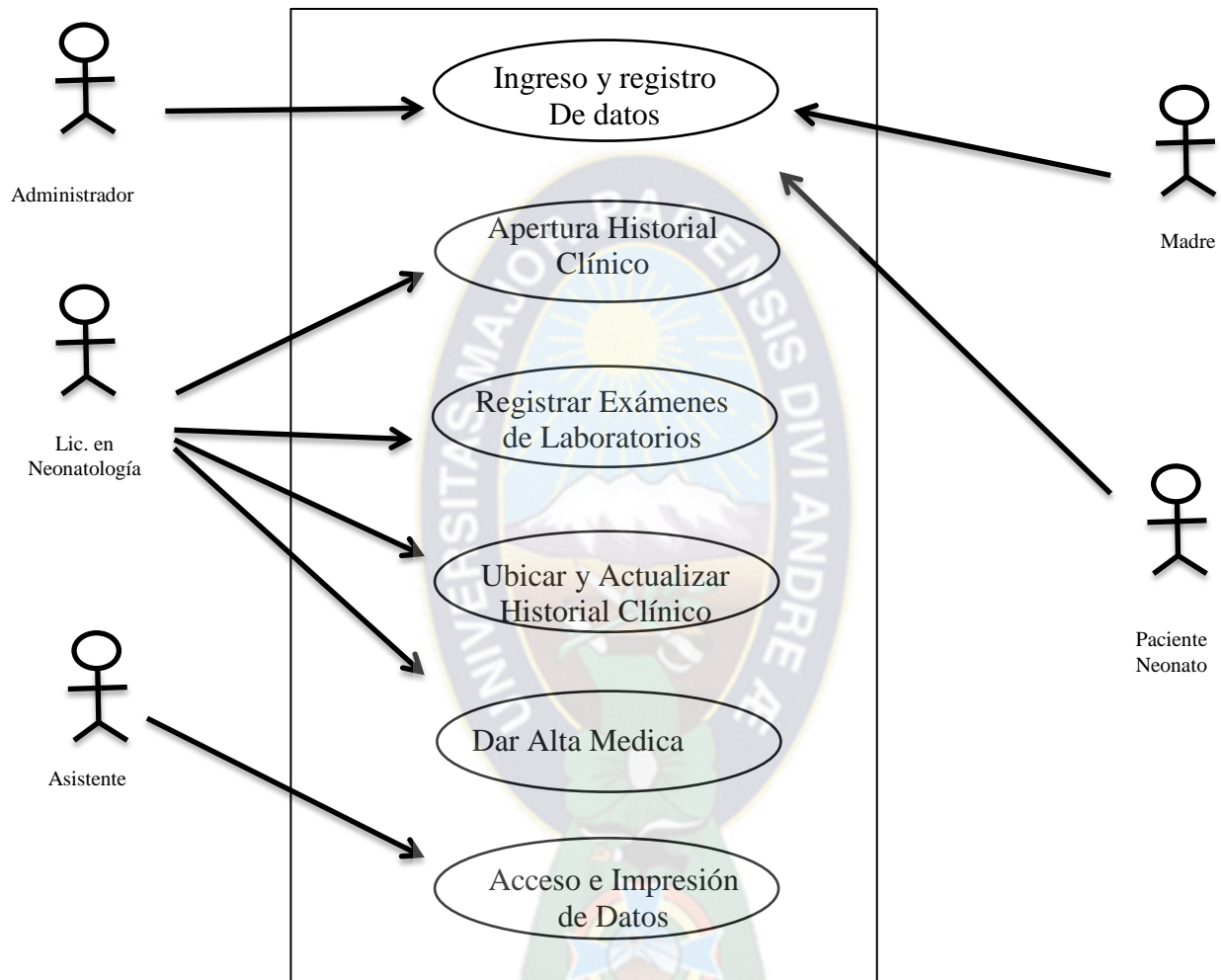


Figura 3.1: Modelado de Casos de Uso del Negocio
[Fuente: Elaboración Propia]

3.2.2.1. Descripción De Actores

A continuación identificaremos a los actores del diagrama de caso de uso del negocio.

ACTOR	DESCRIPCION
Administrador	Es la persona encargada del departamento de administración de sistemas del Hospital Arco Iris, puede acceder a todo el sistema en general.
Lic. en Neonatología	Realiza el registro de datos, la revisión, el diagnóstico de los pacientes neonatos, llenado de datos clínicos, diagnósticos tempranos, él tiene acceso a todas las acciones del sistema como el registro, actualización y ubicación de los pacientes.
Asistente	El asistente solo tendrá acceso a ver e imprimir las historias de los pacientes neonatos.
Madre del Neonato	La madre tiene que proporcionar los datos tanto como de ella y del neonato para poder registrar al paciente y obtener información para el llenado del historial clínico
Paciente Neonato	El Paciente Neonato al no poder valerse por sí mismo es atendido por el Lic. En Neonatología y asistentes.

Tabla 3.1: Descripción de Actores del Casos de Uso del Negocio
[Fuente: Elaboración Propia]

3.2.3. Modelado De Requerimientos

El modelo de requerimientos está formado por una amplia gama de elementos que están basados en el escenario (Casos de uso), orientados a datos (El modelado de datos), basados en clases, orientados al flujo y del comportamiento del mismo. Así mismo, cada uno de

estos elementos mencionados anteriormente estudia el problema desde una perspectiva diferente.

El elemento principal en la descripción de un modelo de requerimientos es el caso de uso. En el contexto de este análisis, un conjunto de casos de uso sirve como base para descubrir los patrones de análisis, el mismo que describe un conjunto pequeño de casos de uso coherentes que describen a su vez una aplicación general.

3.2.3.1. Descripción De Requerimientos A Nivel De Negocios

El requerimiento principal es desarrollar un Sistema Web de Seguimiento y Control de Pacientes Neonatos para el Hospital Arco Iris.

3.2.3.2. Descripción De Requerimientos A Nivel De Usuario

Serán los siguientes:

- U1: Registrar los datos del recién Nacido neonato.
- U2:Llenar el Historial Clínico según diagnóstico del medico
- U3:Asignar sala de internación
- U4:Actualizar el historial Clínico
- U5: Ubicar a un paciente en el sistema

3.2.3.3. Descripción De Requerimientos A Nivel De Sistema

Serán los siguientes:

- S1: El sistema registrara al paciente neonato con todos los datos proporcionados por la madre.
- S2: El sistema mostrara que campos no fueron llenados.
- S3: El sistema controlara las salas de internación.
- S4: El sistema nos permitirá ubicar a cualquier paciente con los datos del mismo.

3.2.3.4. Descripción De Requerimientos A Nivel Técnicos

Tendremos los siguientes requerimientos en software:

- T1: La codificación del sistema web está realizada bajo la plataforma de MySQL y Php.
- T2: Se usara *XAMPP* que es un servidor independiente de plataforma, software libre.

3.3. Fase De Elaboración

En la siguiente fase de elaboración determinaremos todas las soluciones técnicas del proyecto , en la cual se realizaran los modelos de caso de uso, casos de uso extendidos, diagramas de secuencias, estados, clases y diagramas de navegación según la UWE estos a nivel del diseño.

3.3.1. Modelado De Análisis

El modelado de análisis nos reflejara el análisis tomado en el proyecto.

3.3.2. Modelo De Caso De Uso

Este modelado se desarrolla a lo largo de las iteraciones añadiendo así si es necesario nuevos casos de uso y mejorar la descripción de casos de uso que ya existen.

Estos casos de uso representan la interacción entre el sistema y los usuarios.

3.3.3. Diagrama De Caos De Uso De Alto Nivel

A continuación describiremos estos:

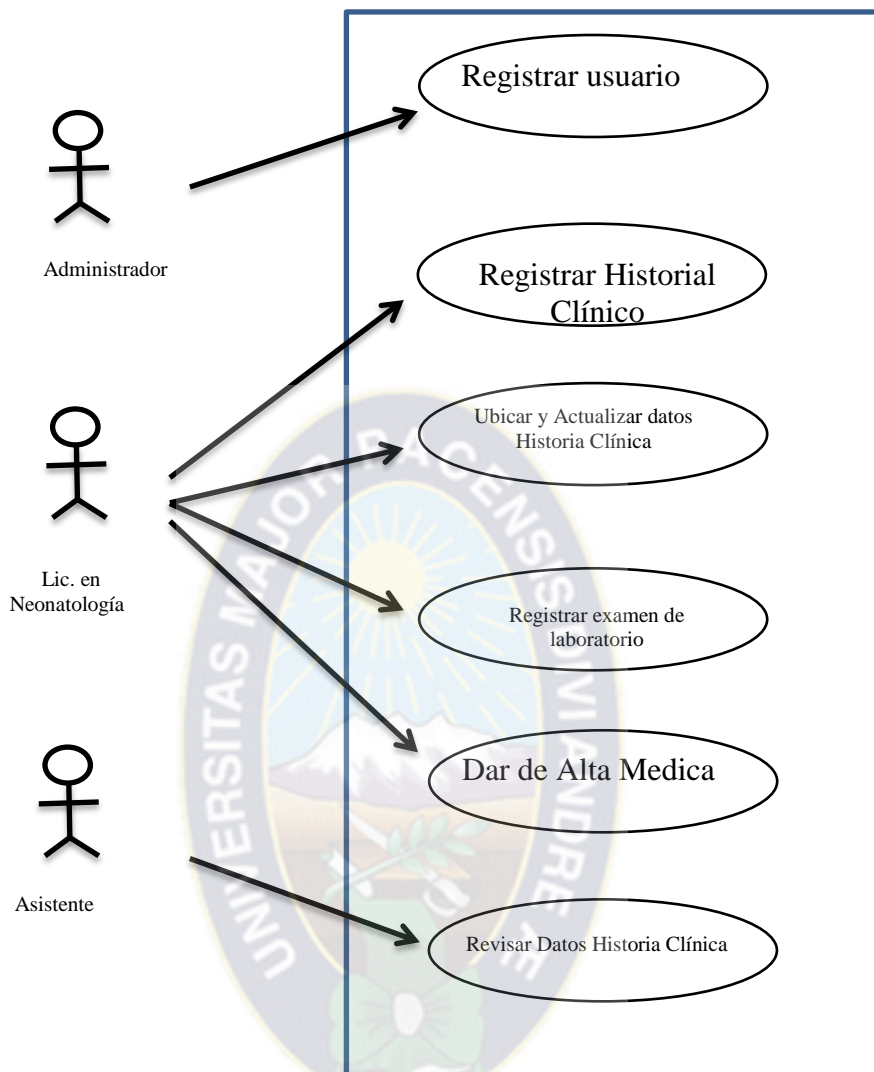


Figura 3.2: Diagrama de Casos de Uso del Sistema Web de Control y Seguimiento de Pacientes

[Fuente: Elaboración Propia]

3.3.4. Descripción De Casos De Uso

Teniendo ya los casos de uso de alto nivel, a continuación describiremos uno por uno con sus respectivos flujos de trabajo.

3.3.4.1. Caso De Uso: Registrar Usuario

Para el registro de usuarios primeramente se debe ingresar al sistema web, los datos será llenado por el administrador, seguidamente se llenara los datos del usuario y a su vez sus datos profesionales como cedula profesional y tipo de profesional.

A continuación mostraremos lo mencionado:

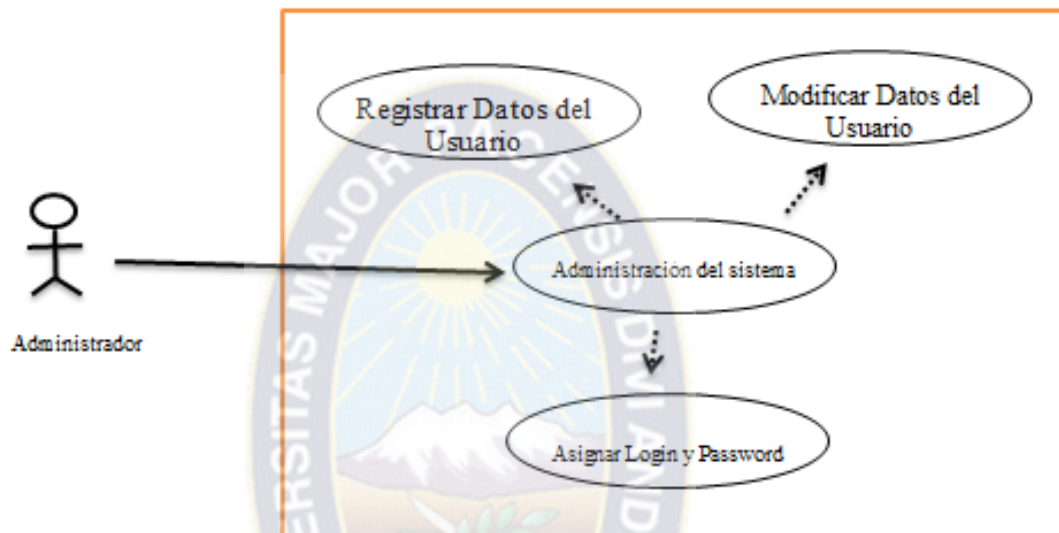


Figura 3.3: Diagrama de Casos de Uso: Registrar Usuario
[Fuente: Elaboración Propia]

Descripción Diagrama de Casos de Uso: Registro Usuario	
Nombre:	Registro de Usuario
Autor:	Wendy Milenka Quiroga Rada
Actores:	Administrador
Descripción:	Permite al administrador registrar, modificar al usuario como también asignarle un Login y Password.

Precondición:	El administrador debe ser usuario del sistema web y cumplir con el rol correspondiente.
Flujo Principal:	<ol style="list-style-type: none"> 1. Autenticación del actor en el sistema web 2. Elegir la opción de Registros, administración del sistema, elegir la opción de registrar, configurar o cambiar contraseña de usuario. 3. Salir del sistema web.
Postcondición:	El sistema web guarda los cambios del administrador.

Tabla 3.2: Descripción de Casos de Uso: Registrar Usuario
[Fuente: Elaboración Propia]

3.3.4.2. Caso De Uso: Registrar Paciente Neonato En Historial Clínico

La apertura del historial clínico consiste en registrar los datos del paciente neonato en el historial clínico, con el respectivo diagnóstico y todos los datos dados por la madre.

A continuación mostraremos lo mencionado:

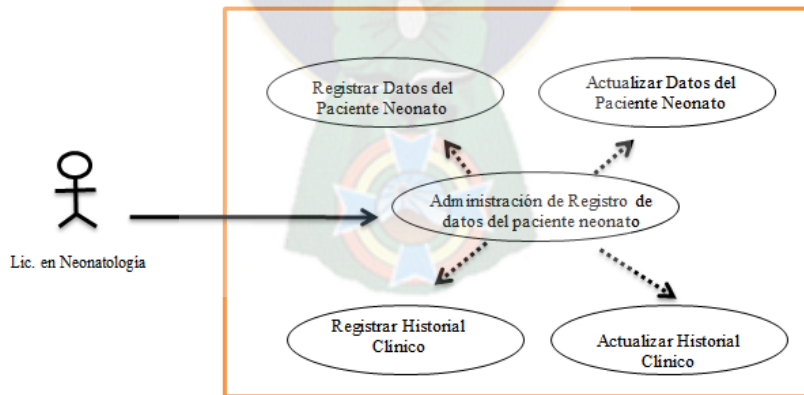


Figura 3.4: Diagrama de Casos de Uso: Registrar Paciente Neonato en Historial Clínico
[Fuente: Elaboración Propia]

Descripción Diagrama de Casos de Uso: Registro datos paciente neonato en historial clínico	
Nombre:	Registro datos paciente neonato en historial clínico
Autor:	Wendy Milenka Quiroga Rada
Actores:	Lic. en Neonatología
Descripción:	Permite al Neonato logo registrar, actualizar los datos del paciente neonato así también su historia clínica llenada con el diagnostico dado por el mismo y la respectiva asignación de sala y cuna.
Precondición:	El Neonato logo debe de ingresar al sistema web con su respectivo Login y Password, seguidamente podrá hacer las modificaciones respectivas en el sistema web en el registro de datos e historial clínico.
Flujo Principal:	<ol style="list-style-type: none"> 1. Autenticación del actor en el sistema. 2. Elegir la opción de registro para pacientes, registrar paciente ahí podrá actualizar al paciente neonato por sus datos personales. 3. Elegir la opción Registros, historial y ahí podrá registrar, editar o buscar una historia clínica. 4. Después de hacer las debidas modificaciones deberá de salir del sistema web.
Postcondicion:	El sistema web guarda todos los datos y actualizaciones del Neonato logo.

Tabla 3.3: Descripción de Casos de Uso: Registrar Paciente Neonato en Historial Clínico
[Fuente: Elaboración Propia]

3.3.4.3. Caso De Uso: Ubicar y actualizar datos Historia Clínica

La ubicación y actualización de las historias clínicas consiste en buscar las historias con los datos personales del paciente neonato y de ahí hacer las modificaciones necesarias a la historia clínica.

A continuación mostraremos lo mencionado:

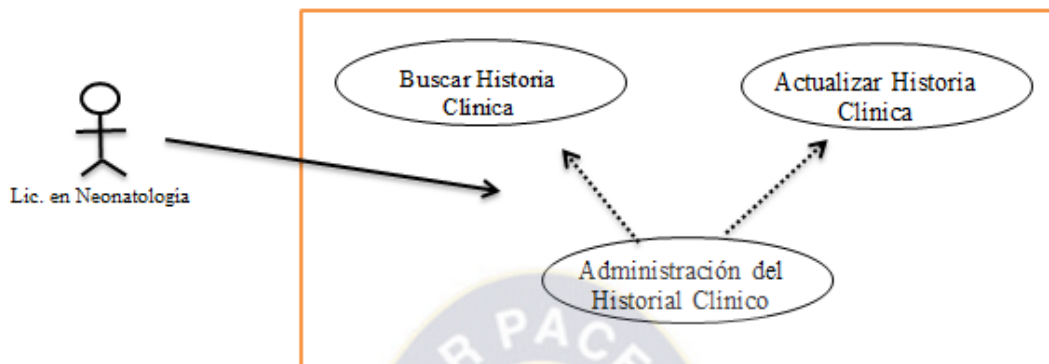


Figura 3.5: Diagrama de Casos de Uso: Ubicar y Actualizar Datos Historial Clínico
[Fuente: Elaboración Propia]

Descripción Diagrama de Casos de Uso: Ubicación y modificación del Historial Clínico	
Nombre:	Ubicación y modificación del Historial Clínico
Autor:	Wendy Milenka Quiroga Rada
Actores:	Lic. en Neonatología
Descripción:	Permite al Neonato logo buscar el historial clínico y hacer las respectivas modificaciones en el mismo.
Precondición:	El Neonato logo debe de ingresar al sistema web con su respectivo Login y Password, seguidamente podrá hacer las modificaciones respectivas en el historial clínico en el sistema web.
Flujo Principal:	1. Autenticación del actor en el sistema.

2. Elegir la opción de registros, ubicar pacientes y en ahí se podrá hacer las modificaciones de los datos del neonato.

3. Elegir la opción Registros, buscar historial y ahí se podrá modificar el historial clínico.

4. Después de hacer las debidas modificaciones deberá de salir.

Postcondicion: El sistema web guarda todos los datos y actualizaciones.

Tabla 3.4: Descripción de Casos de Uso: Ubicar y Actualizar Datos Historial Clínico
[Fuente: Elaboración Propia]

3.3.4.4. Caso De Uso: Registrar Examen De Laboratorio

El registro de laboratorios será realizado por los neonato logos de la UCIN registrándolo en el historial clínico

A continuación mostraremos lo mencionado:

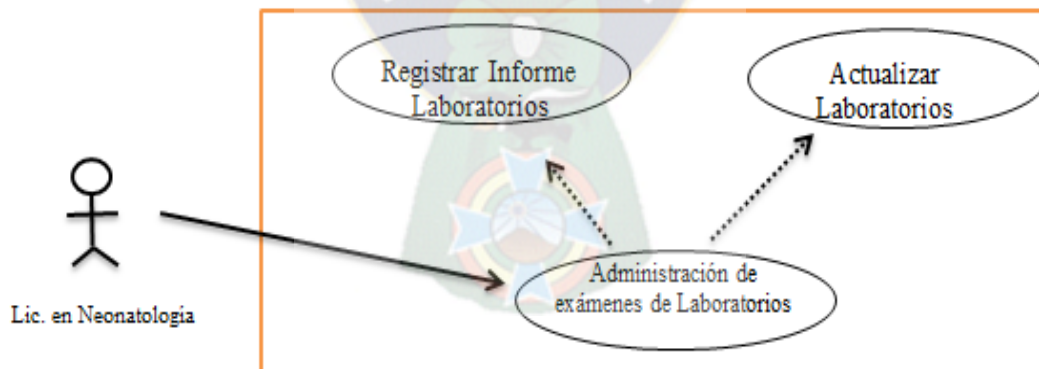


Figura 3.6: Diagrama de Casos de Uso: Registrar Examen de Laboratorios
[Fuente: Elaboración Propia]

Descripción Diagrama de Casos de Uso: Registrar Examen de Laboratorios	
Nombre:	Registrar Examen de Laboratorios
Autor:	Wendy Milenka Quiroga Rada
Actores:	Lic. en Neonatología
Descripción:	Permite al Neonato logo registrar todos los laboratorios tomados al paciente neonato.
Precondición:	El Neonato logo debe de ingresar al sistema web con su respectivo Login y Password, seguidamente podrá registrar los laboratorios tomados al paciente neonato en el historial clínico en el sistema web.
Flujo Principal:	<ol style="list-style-type: none"> 1. Autenticación del actor en el sistema. 2. Elegir la opción de laboratorios, informe de laboratorio y en ahí se podrá hacer los registros de los mismos. 3. Después de hacer los debidos registros deberá de salir del sistema web.
Postcondicion:	El sistema web guarda todos los informes de laboratorios del Neonato logo.

Tabla 3.5: Descripción de Casos de Uso: Registrar Examen de Laboratorios
[Fuente: Elaboración Propia]

3.3.4.5. Caso De Uso: Dar de Alta Médica

La alta médica la dará el neonato logo de acuerdo al diagnóstico, crecimiento y evolución del neonato si así lo ve conveniente.

A continuación mostraremos lo mencionado:

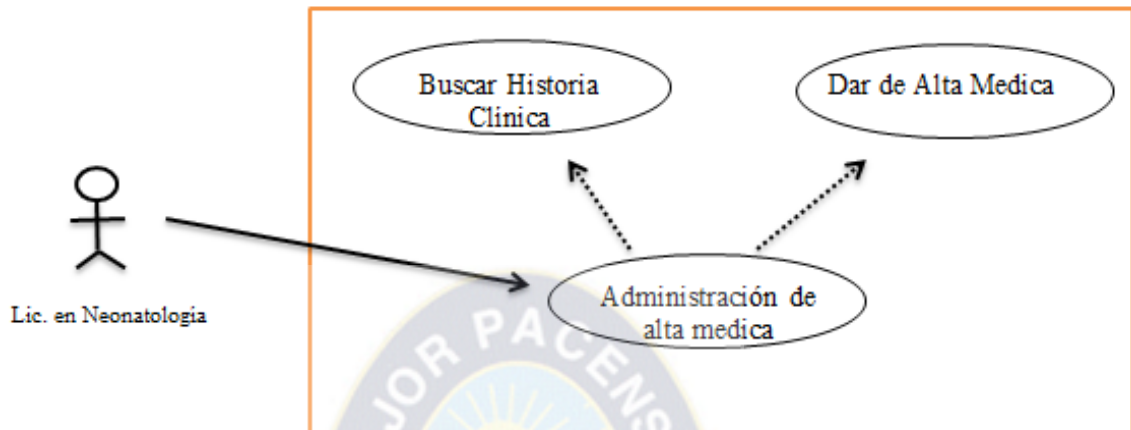


Figura 3.7: Diagrama de Casos de Uso: Dar de Alta Médica
[Fuente: Elaboración Propia]

Descripción Diagrama de Casos de Uso: Dar el Alta Medica	
Nombre:	Dar el Alta Medica
Autor:	Wendy Milenka Quiroga Rada
Actores:	Lic. en Neonatología
Descripción:	Permite al Neonato logo levantar el alta médica de acuerdo al diagnóstico y a la evolución del Neonato.
Precondición:	El Neonatologo debe de ingresar al sistema web con su respectivo Login y Password, seguidamente podrá revisar todo el historial clínico en el sistema web y de ahí procederá a dar la respectiva nota de alta.
Flujo Principal:	1. Autenticación del actor en el sistema.

2. Elegir la opción de informes, dar alta médica.

3. Después deberá llenar la nota de alta con los datos del paciente, las fechas de ingreso, el diagnóstico y peso con el que ingreso y egreso, la evolución que mostro y programas la fecha de su próxima consulta.

Postcondicion: El sistema web guarda los informes de las altas dadas por el Doctor

Tabla 3.6: Descripción de Casos de Uso: Dar de Alta Médica
[Fuente: Elaboración Propia]

3.3.4.6. Caso De Uso: Revisar Datos Historia Clínica

La revisión de las historias clínicas podrá ser vista por todos los asistentes necesarios, restringiéndoles de toda modificación tanto en los datos como el historial clínico.

A continuación mostraremos lo mencionado:

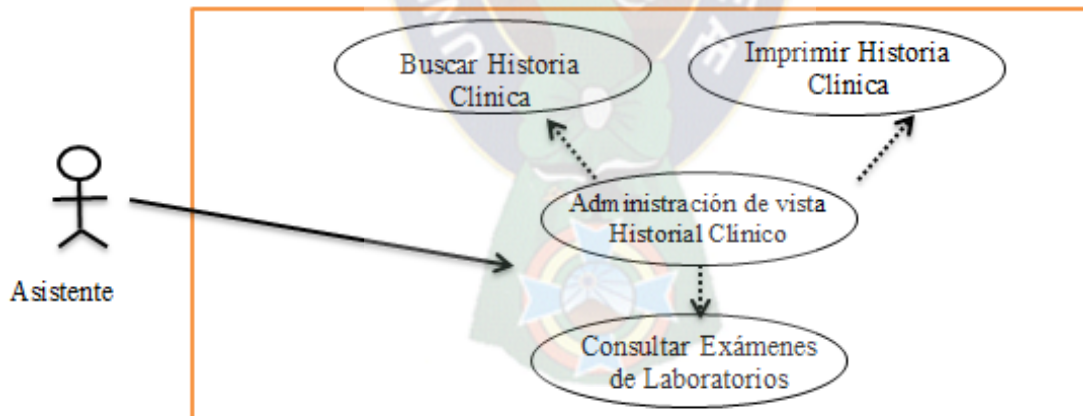


Figura 3.8: Diagrama de Casos de Uso: Revisar Datos Historia Clínica
[Fuente: Elaboración Propia]

Descripción Diagrama de Casos de Uso: Revisión Datos Historia Clínica	
Nombre:	Revisión Datos Historia Clínica
Autor:	Wendy Milenka Quiroga Rada
Actores:	Asistente
Descripción:	Permite al asistente visualizar e imprimir todos los informes con respecto a un paciente neonato.
Precondición:	El asistente está habilitado para el acceso de datos e impresión.
Flujo Principal:	<ol style="list-style-type: none"> 1. Autenticación del actor en el sistema. 2. Elegir la opción que desee visualizar o también imprimir. 3. Salir del Sistema web
Postcondición:	El sistema web genera e imprime reportes.

Tabla 3.7: Descripción de Casos de Uso: Revisar Datos Historia Clínica
[Fuente: Elaboración Propia]

3.3.5. Diagrama De Paquetes

Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

A continuación se muestra un esquema de los módulos que comprende nuestro sistema

Están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido.

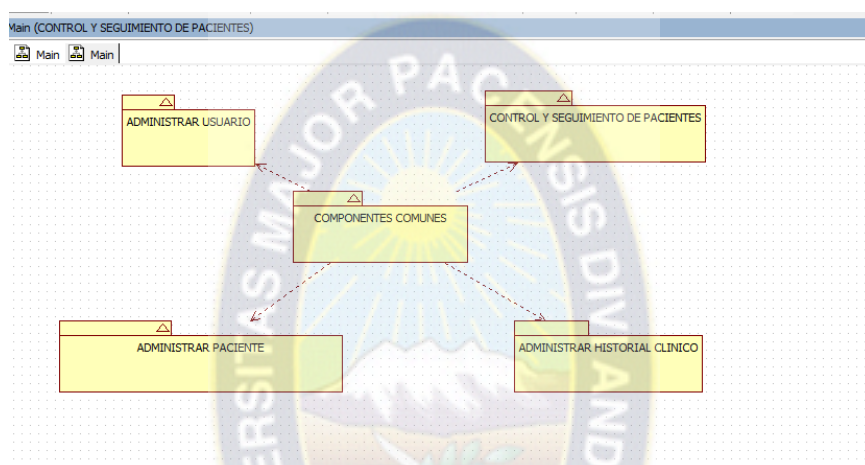


Figura 3.9: Diagrama de paquetes Sistema Web de Seguimiento y Control de Pacientes Internos
[Fuente: Elaboración Propia]

3.3.6. Diagrama de clases

Los diagramas se utilizan generalmente para facilitar el entendimiento de largas cantidades de datos y la relación entre diferentes partes de los datos también para realizar cálculos electrónicos

Un diagrama de clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Nos sirven para ver las relaciones entre las clases que envuelven el sistema, las cuales pueden ser asociativas, de herencia, de uso y de convencimiento. Un diagrama de clases está comprendido por los siguientes elementos:

Clases: atributos, métodos y visibilidad.

Relaciones: herencia, composición, agregación, asociación y uso.

Este es el diagrama principal para el análisis y diseño, en la Figura3.11 se presentan las relaciones entre las clases, atributos y sus operaciones para representar la información del sistema. Después de haber realizado el análisis para la base de datos e identificar todas las entidades que intervienen en el sistema, se elabora el diagrama de clases.



3.3.7. Modelo de navegación

Observaremos el espacio de navegación que especifica los objetivos que pueden ser visitados mediante la navegación de los usuarios, clasificados por el rol que se les asigne.

En el diagrama de navegación se tienen tres usuarios: Un superusuario que en este caso es el encargado de sistemas del hospital, un usuario que en este caso son los médicos y licenciados en neonatología terminando con los asistentes que solo tendrán privilegios mínimos dentro del sistema.

3.3.8. Modelo de navegación de menús

La navegación de menús es un diseño de cómo se presentará nuestro sistema dependiendo del tipo de usuario que se tenga, es decir dependiendo del rol que tenga nuestro usuario.

Se podrá observar las opciones del menú que tiene cada usuario, todos los usuarios deben autenticarse antes de ingresar al sistema, en el caso del usuario administrativo tiene las opciones de realizar ajustes de usuarios.

Los médicos y licenciados en neonatología tienen el acceso a hacer todas las modificaciones en el sistema, y por último los asistentes tienen la opción de observar y recabar los reportes.

3.4. Fase De Construcción

El objetivo de la fase de Construcción consiste en desarrollar el sistema hasta el punto en que está listo para la pre-producción de pruebas. En las etapas anteriores, la mayoría de los requisitos han sido identificados y la arquitectura del sistema se ha establecido. El énfasis es priorizar y comprender los requerimientos, modelado que ataca una solución y, a luego, la codificación y las pruebas del software.

3.4.1. Diseño De Interfaces

El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora, siguiendo así un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de ésta y luego crea una plantilla de pantalla que constituye la base del prototipo de la interfaz de usuario.

3.4.1.1. Autenticación

El usuario que desee acceder al sistema web deberá tener un Login y un Password asignados por el administrador del sistema para poder ingresar al mismo.



Figura 3.11: Autenticación del sistema
[Fuente: Elaboración Propia]

3.4.1.2. Pantalla Principal

Después que el usuario ingresa con su respectivo Login y Password, el sistema web le permitirá el ingreso a la página principal, en la cual dependiendo del rol que desempeña el usuario para acceder a las diferentes funciones del sistema, que se muestra.

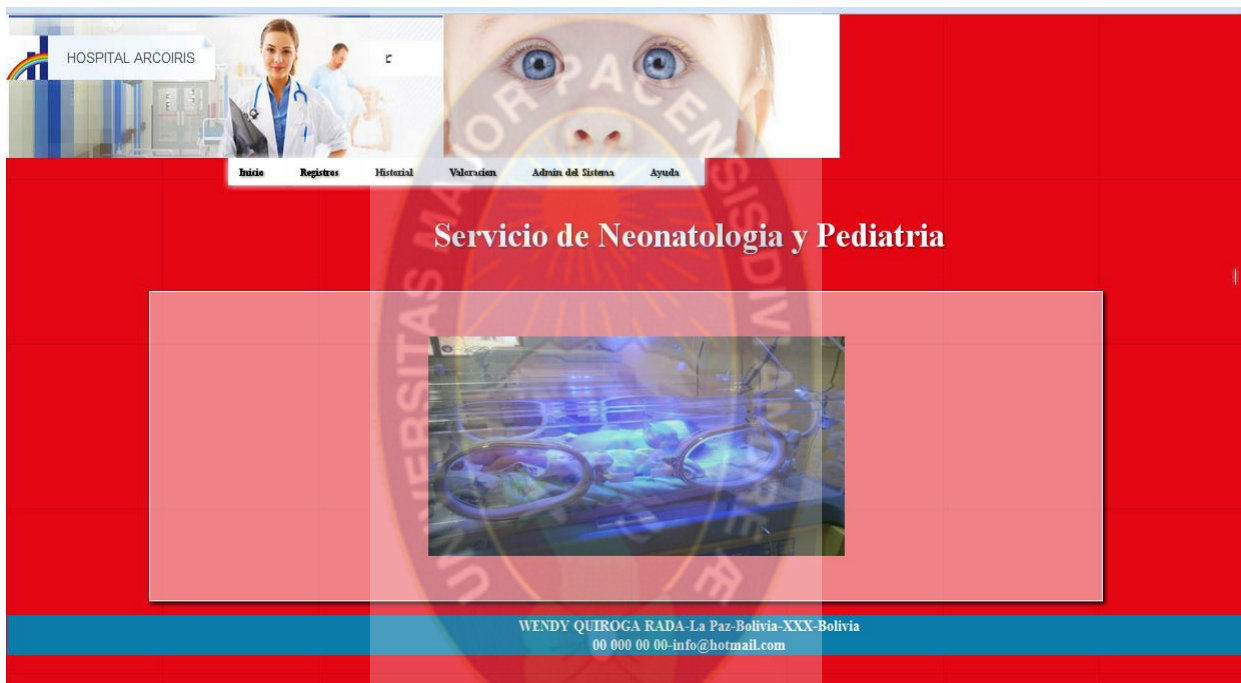


Figura 3.12: Pantalla principal del sistema
[Fuente: Elaboración Propia]

3.4.1.3. Registro De Paciente

Previamente se deberá elegir la opción Registros seguidamente Registrar Paciente en el menú, en este registro se llenaran los datos personales del recién nacido neonato.

Se mostrara en rojo los espacios que faltan llenar como una alerta, luego de completar los datos faltantes se procederá a registrar los datos del paciente o también teniendo la opción de limpiar los datos en caso de equivocarse.

HOSPITAL ARCOIRIS | **ical Center**

Inicio | Registros | Historial | Valoracion | Admin del Sistema | Ayuda

Registro del Recien Nacido

Datos de la Madre:
 Nombres de la Madre:
 Apellidos de la Madre:
 Edad: años
 Cédula:
 Teléfono:

Datos del Recien Nacido:
 Nombres de RN:
 Apellidos de RN:
 Fecha de Nacimiento:
 Sexo:
 Hora:

Reaninacio:
 SI No Mascarina
 Intubacion Manejo Cardioso

Observaciones al Nacer:

Edad Gestacional: Semanas
Peso de Nacimiento: gr.
Talla: cm.
R.P.M.:
Líquido Amniótico:
Perímetro cefálico: cm.
Perímetro torácico: cm.
Perímetro abdominal: cm.
Sala:
Grupo Sanguíneo:
VIH:

WENDY OJIBOGA RADA La Paz-Bolivia-XXX-Bolivia

Figura 3.13: Registro de paciente
 [Fuente: Elaboración Propia]

3.4.1.4. Caso de uso: Actualización Datos de Paciente

Elegimos la opción de Registros seguidamente buscamos al recién nacido con el ci de la madre y nos ubicara al paciente que deseamos actualizar.

De ahí podemos modificar los datos del paciente y después elegimos la opción de guardar o imprimir reporte.



Figura 3.14: Actualización datos del paciente
[Fuente: Elaboración Propia]

3.4.1.5. Caso de uso: Ubicar Paciente

En esta opción ubicaremos al paciente para poder revisar sus datos y así poder imprimir el reporte.

Ingresamos a la opción Registros luego vamos a Ubicar Paciente de acá buscamos al paciente por el apellido del paciente neonato o por la sala que fue registrado.



Figura 3.15: Ubicar paciente
[Fuente: Elaboración Propia]

3.4.1.6. Caso de uso: Crecimiento y Desarrollo

Una vez registrado el paciente se le hará los análisis necesarios y plasmarlos en el sistema.

Entraremos a la opción Valoración seguido de Crecimiento y Desarrollo y buscamos al paciente al cual se le hará la valoración respectiva. Ahí se llenara toda la información en las tablas del crecimiento y desarrollo del recién nacido neonato.

Terminamos registrando o limpiando.

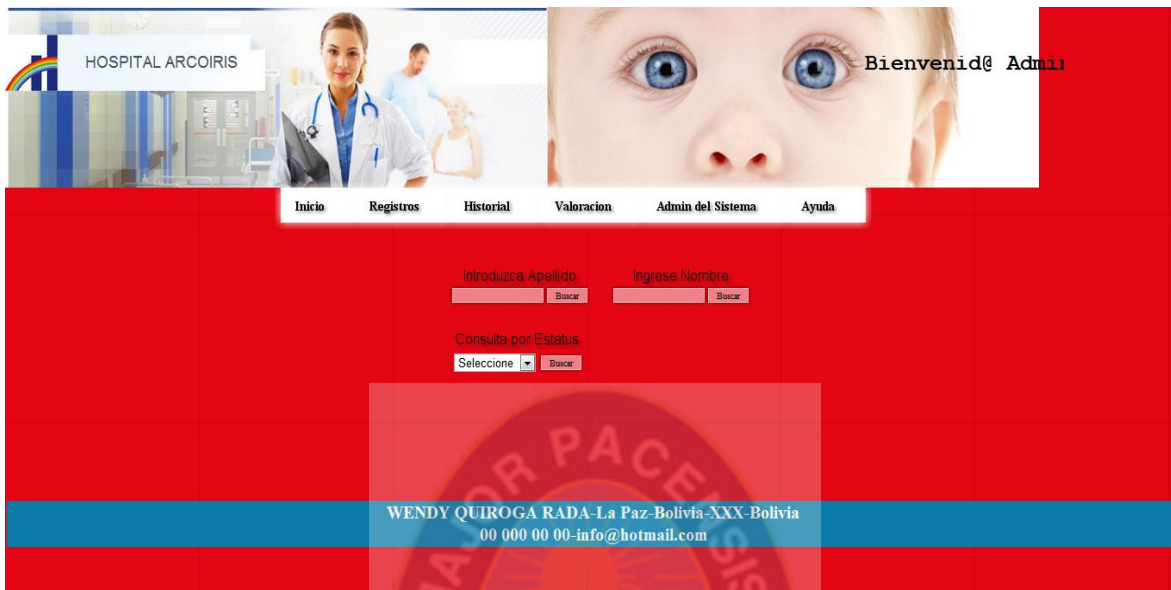


Figura 3.16: Crecimiento y desarrollo
[Fuente: Elaboración Propia]

CRECIMIENTO Y DESARROLLO					
Nombre: IN BABOS	Sexo: MASCULINO	ID Paciente: 0			
Fecha Nac: 00/00/00	Fecha Transf: <input type="text"/>	Peso: <input type="text"/>	VAR: <input type="text"/>		
Fecha: 00-00-00	DIAS: 000	Perimetro Cefálico: <input type="text"/>	Estatura: <input type="text"/>	Ingreso: <input type="text"/>	
Horas: <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
VO: <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Sexo Materno: <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
O2: <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	F	<input type="text"/>	F	<input type="text"/>	F
Veneclisis	Hrs: <input type="text"/>	<input type="text"/>	Hrs: <input type="text"/>	<input type="text"/>	Hrs: <input type="text"/>
	/	<input type="text"/>	/	<input type="text"/>	/
	ugtm	<input type="text"/>	ugtm	<input type="text"/>	ugtm
Farmacos:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
OTRAS IND:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Liberstato Interens:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Ecografía Transfontanelar:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
ROP:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Observaciones:	<input type="text"/>				
	<input type="text"/>				
Nombre del Profesional:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 3.17: Formulario crecimiento y desarrollo
[Fuente: Elaboración Propia]

3.4.1.7. Caso de uso: Cuidados Intermedios

Ya cuando el paciente es internado se procede a llenar los formularios correspondientes como son: Cuidados Intermedios Neonatales, UTI Neonatal, Control de Enfermedades de Neonatología.

Se entrara por medio de la opción Valoración del menú principal, seguidamente elegiremos la opción de cuidados intermedios. Seguidamente se elegirá la opción deseada y se buscara al paciente al cual se le llenara el formulario con sus datos correspondientes o por la sala de internación.



Figura 3.18: Ingreso cuidados intermedios
[Fuente: Elaboración Propia]

CONTROL DE ENFERMERIA DE NEONATOLOGIA

Nombre: <input type="text"/>		Edad: <input type="text"/>		Sexo: <input type="text"/>		Peso anterior: <input type="text"/>		Peso actual: <input type="text"/>		NCHL: <input type="text"/>			
Fecha: <input type="text"/>													
Hrs	Signos Vitales				SO 2%	Oxígeno X Libre	Ingreso VO-SOO	Egresos			Úna	Observación	Control Glucemia
	T	FC	FR	PA				Residual	Otra	Deposita			
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
1													
2													
3													
4													
5													
6													
7													

INGRESOS				Egresos	
TOTAL VO		INGRESOS		Egresos	
Controlado:				Residue	
				Alimento	
TOTAL				Diferencia	
Via Parenteral:				Deposición	

INGRESO VV					
VENOCLEIS	M	T	N	M	
1					
2					
3					
4					
5					

MEDICAMENTOS					
	M	T	N	M	

WENDY QUITROGA RADA, La Pro-Beltrina XXX Beltrina
00 000 00 0000@hospitall.com

Figura 3.21: Formulario Control de enfermería de neonatología
[Fuente: Elaboración Propia]

3.4.1.8. Caso de uso: Terapia Intensiva

De la misma forma se procederá al ingreso de Terapia Intensiva seguidamente de la opción Terapia Intensiva Centro de Enfermería UCIN-UCIP y registraremos los datos correspondientes y proseguiremos a Registrar los datos ingresados por el neonato logo.

3.4.1.9. Caso de uso: Registrar Usuario Medico o Profesional

El superusuario entrara a esta opción de Admin del Sistema y procederá al llenado del formulario para nuevo usuario del sistema web. El mismo procederá al registro del nuevo usuario.

HOSPITAL ARCOIRIS A Medical Center

Inicio Registros Historial Valoracion Admin del Sistema Ayuda

Registro de Usuarios y Profesional

DATOS DEL USUARIO

Nombre:
Login:
Password:
Repetir Password:
Tipo:

ADMINISTRADOR

DATOS DEL PROFESIONAL

Cedula del Profesional:
Nombre:
Tipo:

MEDICO

Registrar Limpiar

WENDY QUIROGA RADA - La Paz - Bolivia - XXX - Bolivia
00 000 00 00 - info@hotmail.com

Figura 3.24: Registro de usuarios y profesional
[Fuente: Elaboración Propia]

3.4.1.10. Caso de uso: Configuración de Usuario y Cambio de Contraseña

El superusuario ingresara a la opción De Admin del Sistema y elegirá Configuración de Usuarios de ahí buscara al médico por su Login o nombre para proceder a la modificación ya sea de nombre, Login, o Password.

Lo mismo para el cambio de contraseña.



Figura 3.25: Modificación de usuario
[Fuente: Elaboración Propia]

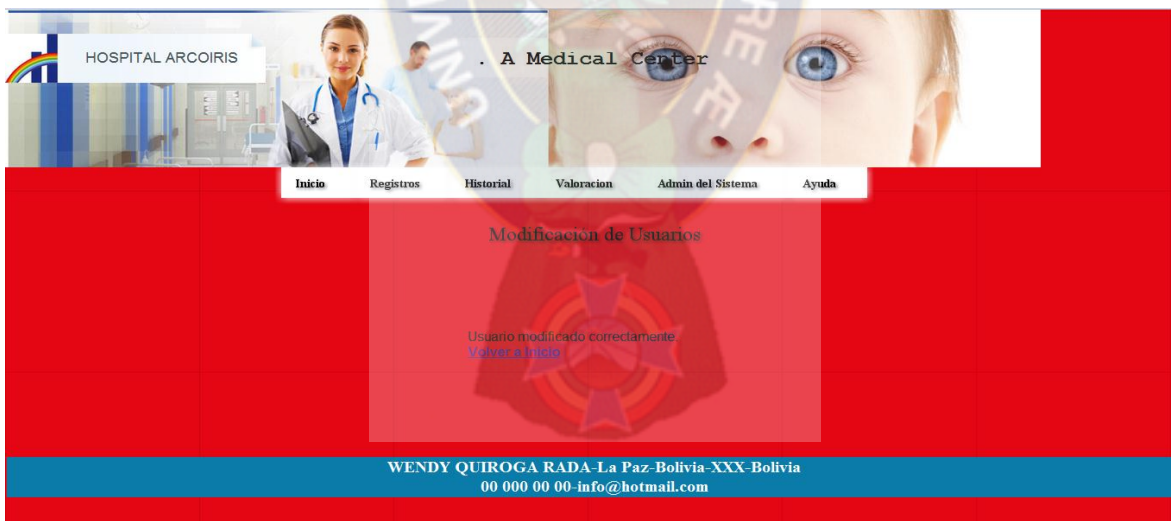


Figura 3.26: Verificación modificación de usuario
[Fuente: Elaboración Propia]



Figura 3.27: Cambio contraseña
[Fuente: Elaboración Propia]

3.5. Fase De Transición

Esta fase se enfoca en liberar el sistema a producción. Para finalizar esta fase se completa con la aprobación y visto bueno del diseño e implementación del sistema, por parte de los usuarios.

3.6. Políticas De Implementación

Las especificaciones del diseño del sistema web sirven como base para la construcción del nuevo sistema.

El proceso de activar el nuevo sistema de información y retirar el viejo, el cambio puede ser rápido o lento, dependiendo del método a usarse. Los cuatro métodos son:

- Corte y cambio o direct cutover.
- Operación paralela o parallel operation.

- Operación piloto o pilot operation.
- Operación de fase o phased changeover.

3.6.1. Corte y Cambio

El cambio del sistema viejo al nuevo ocurre tan pronto el nuevo sistema es operacional. Usualmente es el método menos costoso, puesto que se debe operar y mantener un sistema a la vez. Por otro lado, es el método con mayor riesgo, pues no se puede usar el sistema viejo para ayudar a resolver los problemas que ocurran, o para verificar que todos los detalles de los datos sean correctos.

3.6.2. Operación Paralela

Ambos sistemas operan por un tiempo específico. Los datos son entradas a ambos sistemas y el output (salida) generado por el sistema opera correctamente, se deja de usar el sistema viejo.

La ventaja mayor es su bajo riesgo, si el nuevo sistema no trabaja bien, se puede usar el sistema viejo hasta que se realicen los cambios apropiados. Sin embargo, este método es el más costoso, pues hay que darle servicios a ambos sistemas. Los usuarios deben trabajar en los dos sistemas y a veces hace falta ayuda temporaria. Este método no es practico so los dos sistemas son incompatibles o realizan funciones diferentes.

3.6.3. Operación Piloto

Se implementa todo el nuevo sistema en un área de la compañía.

El grupo que usa el nuevo sistema se conoce como área o grupo piloto. El sistema viejo continúa operando en toda la organización, incluyendo el área piloto. Después de probar que el sistema trabaja correctamente en el área piloto, se implanta en toda la organización, normalmente usando el método corte y cambio. La operación piloto reduce el riesgo de

fallas del sistema comparando con corte y cambio: y es menos costoso que la operación paralela.

3.6.4. Operación De Fase

Se implementa el nuevo sistema en fases o módulos, en vez de implantar todo el sistema a la vez.

Cada subsistema se puede implantar usando uno de los otros métodos.

Una ventaja es que el resto se limita solo al modelo implantado. Este método no es posible si no se puede separar el sistema en módulos. No se debe confundir el método operación de fase con la operación piloto. En la operación de fase se da parte del sistema a todos los usuarios, mientras que la operación piloto provee el sistema completo, pero solo a un grupo de usuarios.

3.6.5. Aplicación del método corte y cambio

Para el presente trabajo se aplicara el metodom corte cambio por las siguientes razones:

- Toma menos tiempo y esfuerzo.
- El nuevo sistema esta funcionando inmediatamente.

4. CALIDAD DE SOFTWARE Y SEGURIDAD

4.1. Introducción

Empezaremos hablando de la calidad de software, que es el desarrollo de software basado en estándares con la funcionalidad y rendimiento total que satisface los requerimientos del cliente.

A la conclusión del Sistema Web de Control y Seguimiento de Pacientes Internos se debe medir la calidad del producto, hay dos tipos de medición: la directa y la indirecta.

En el presente proyecto se aplicara las medidas por la norma ISO 9126.

4.2. Características Propuestas por ISO-9126

ISO 9126 es un estándar internacional para la evaluación del software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software.

La normativa define seis características de la aplicación, estas seis características son divididas en un número de sub- características, las cuales representan un modelo detallado para la evaluación de cualquier sistema informático.

El modelo establece diez características, seis que son comunes a las vistas interna y externa y cuatro que son propias de la vista en uso.

A continuación se describen las características y su características propias de este estándar que se encuentran dentro de las vistas interna y externa, las cuales usaremos para evaluar el software.

Funcionalidad: Capacidad del software de proveer los servicios necesarios para cumplir con los requisitos funcionales.

Subcaracterísticas:

Idoneidad.- Hace referencia a que si el software desempeña las tareas para las cuales fue desarrollado.

Exactitud.- Evalúa el resultado final que obtiene el software y si tiene consistencia a lo que se espera de él.

Interoperabilidad.- Consiste en revisar si el sistema puede interactuar con otro sistema independiente.

Seguridad.- Verifica si el sistema puede impedir el acceso a personal no autorizado.

Fiabilidad: capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.

Subcaracterísticas:

Madurez.- Se debe verificar las fallas del sistema y si muchas de estas han sido eliminadas durante el tiempo de pruebas o uso del sistema.

Recuperabilidad.- Verificar si el software puede reasumir el funcionamiento y restaurar datos perdidos después de un fallo ocasional.

Tolerancia a fallos.- Evalúa si la aplicación desarrollada es capaz de manejar errores.

Usabilidad: esfuerzo requerido por el usuario para utilizar el producto satisfactoriamente.

Subcaracterísticas:

Aprendizaje.- Determina que tan fácil es para el usuario aprender a utilizar el sistema.

Comprensión.- Evalúa que tan fácil es para el usuario comprender el funcionamiento del sistema

Operatividad.- Determina si el usuario puede utilizar el sistema sin mucho esfuerzo.

Atractividad.- Verifica que tan atractiva se ve la interfaz de la aplicación.

Eficiencia: relación entre las prestaciones del software y los requisitos necesarios para su utilización.

Subcaracterísticas:

Comportamiento en el tiempo.- Verifica la rapidez en que responde el sistema

Comportamiento de recursos.- Determina si el sistema utiliza los recursos de manera eficiente

Mantenibilidad: esfuerzo necesario para adaptarse a las nuevas especificaciones y requisitos del software.

Subcaracterísticas:

Estabilidad.- Verifica si el sistema puede mantener su funcionamiento a pesar de realizar cambios.

Facilidad de análisis.- Determina si la estructura de desarrollo es funcional con el objetivo de diagnosticar fácilmente las fallas.

Facilidad de cambio.- Verifica si el sistema puede ser fácilmente modificado

Facilidad de pruebas.- - Evalúa si el sistema puede ser probado fácilmente

Portabilidad: capacidad del software ser transferido de un entorno a otro.

Subcaracterísticas:

Capacidad de instalación.- Verifica si el software se puede instalar fácilmente

Capacidad de reemplazamiento.- Determina la facilidad con la que el software puede reemplazar otro software similar.

Adaptabilidad.- El software se puede trasladar a otros ambientes

Co-Existencia.- El software puede funcionar con otros sistemas

Cada una de las características debe ser evaluada dentro del software basándonos en pruebas de funcionamiento, medición de rendimiento y pruebas con usuarios que harán uso del sistema.

4.2.1. Confiabilidad

La confiabilidad de un sistema es un elemento importante en su calidad general. Para determinar esta se toma en cuenta las fallas que se producen en el sistema en un determinado tiempo, así también es el grado en que el sistema responde bajo las condiciones definidas durante un intervalo de tiempo dado. Primeramente se considera la confiabilidad de cada módulo independiente. Para ello se requiere el modelo del sistema, que se puede observar en la figura 4.1.

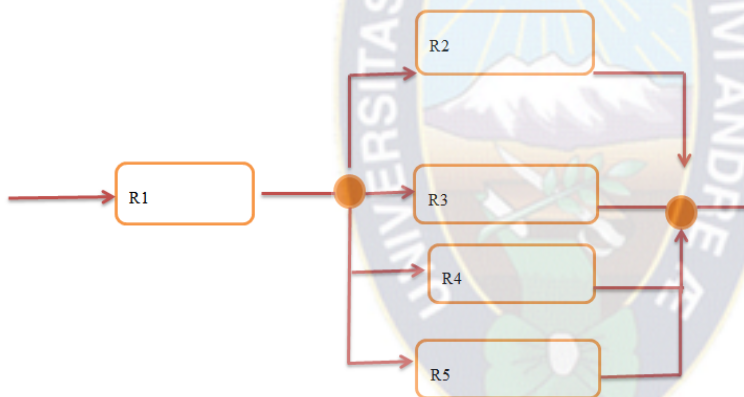


Figura 4.1:Modelo del Sistema Web de Control y Seguimiento de Pacientes
[Fuente: Elaboración Propia]

Tomando en cuenta la relación de: $\{1\} R(t) = e^{-\lambda t}$

Dónde :

$R(t)$ = Confiabilidad de un componente o subsistema t.

$e^{-\lambda t}$ = Probabilidad de falla de un componente o subsistema en el tiempo t.

T = Tiempo de trabajo sin fallas.

λ = Tasa constante de fallas.

t =Periodo de operación de tiempo

Se realiza el cálculo de la confiabilidad de cada módulo del sistema con (1).

MODULO	λ	T	R(t)
1 Autenticación del usuario	0.02	2 Hrs.	0.96
2 Registro del historial del paciente	0.03	3 Hrs.	0.91
3 Actualización datos del paciente	0.04	3 Hrs.	0.89
4 Ubicación del historial clínico	0.03	3 Hrs.	0.91
5 Seguimiento del paciente interno	0.04	3 Hrs.	0.89

Tabla 4.1:Calculo de Confiabilidad
[Fuente: Elaboración Propia]

El modelo de sistema en la figura nos muestra una conexión compuesta donde se tiene inicialmente una conexión en serie y posteriormente una conexión en paralelo.

Por lo tanto realizamos los cálculos correspondientes tenemos:

$$(2) R = R_i * R_S$$

Dónde: $R_i = R_1 = 0.96$

$$(3) R_s(t) = 1 - \{[1 - R_2(t)] * [1 - R_3(t)] * \dots * [1 - R_{n-1}(t)] * [1 - R_n(t)]\}$$

Reemplazando en (3) tenemos:

$$R_s(t) = 1 - [(0.09) * (0.11) * (0.09) * (0.11) * (0.08)]$$

$$R_s(t) = 0.9999922$$

Reemplazando en (2) tenemos:

$$R = 0.96 * 0.9999922$$

$$R\% = 96\%$$

Con este resultado se podría concluir que el Sistema Web de Seguimiento y Control de Pacientes Internos Caso Hospital Arco Iris, presenta una confiabilidad del 96% entonces se afirma que es un sistema confiable.

4.2.2. Funcionalidad

Esta no se mide directamente, por tanto es necesario evaluar un conjunto de características y capacidades del sistema. Nuestro sistema debe ser capaz de proveer las funciones que cumplen con las necesidades explícitas e implícitas cuando es utilizado en las condiciones especificadas por el cliente.

Para el cálculo de la funcionalidad utilizaremos la métrica de punto función, para esto se debe determinar cinco características de dominios de información y se proporciona las cuentas en la posición apropiada a la tabla.

Los valores de dominio de información se definen de la siguiente forma:

- Número de entradas de usuario

- Número de salidas de usuario
- Número de peticiones de usuario
- Numero de archivos
- Numero de interfaces externas

Para calcular puntos función(PF), se utiliza la siguiente relación:

$$(1) PF = cuenta\ total * (0.65 + 0.01 * \sum F_i)$$

Dónde:

PF =Medida de funcionalidad

Cuenta total = Es la suma de todas las entradas obtenidas en :N° de entradas, N° de salidas, N°de peticiones, N° de archivos y N° de interfaces externas.

$\sum F_i$ =Son los valores de ajuste de complejidad según las respuestas a preguntas destacadas en la siguiente tabla.

Tomando en cuenta estos pasos ahora encontraremos el punto función.

4.2.2.1. Número De Entradas De Usuarios

Es la información que llega desde el exterior, tiene una sola dirección del exterior al interior.

Se puede observar en la tabla

N°	ENTRADAS DE USUSARIOS
1	Pantalla de ingreso al sistema
2	Registro de usuario

3	Registro de datos del paciente
4	Registro de historial clínico
5	Registro crecimiento y desarrollo
6	Registro cuidados intermedios
7	Registro UTI neonatal
8	Registro control de enfermedades
7	Registro cuidados intensivos
9	Registro laboratorios
10	Registro de alta medica

Tabla 4.1: Entradas de Usuario
[Fuente: Elaboración Propia]

4.2.2.2. Número de salidas de usuario

Es la información elaborada por el sistema que es transmitida al usuario, también actualizan algunos archivos, tiene una sola dirección del exterior al interior. Se puede observar en la tabla las salidas de usuario.

Nº	SALIDA DE USUARIO
1	Detalle del usuario
2	Detalle de los datos del paciente
3	Detalle de historial clínico

4	Detalle de crecimiento y desarrollo
5	Detalle de cuidados intensivos
6	Detalle de UTI neonatal
7	Detalle de control de enfermedades
8	Detalle de terapia intensiva
9	Detalle de laboratorios
10	Detalle de alta medica

Tabla 4.2:Salida de Usuario
[Fuente: Elaboración Propia]

4.2.2.3. Número De Peticiones De Usuario

Es una entrada interactiva que produce la generación de alguna respuesta de software inmediata en forma de salida interactiva. Se puede observar las peticiones de usuario en la figura

N°	PETICIONES DE USUARIO
1	Autenticación usuario
2	Modificación de usuario
3	Modificación de paciente
4	Modificación de historial clínico

5	Modificación de crecimiento y desarrollo
6	Modificación de cuidados intensivos
7	Modificación de UTI neonatal
8	Modificación de control de enfermedades
9	Modificación de cuidados intensivos
10	Listado de pacientes internos

Tabla 4.3: Peticiones de Usuario
[Fuente: Elaboración Propia]

4.2.2.4. Número De Archivos

Nos referiremos a cada archivo maestro lógico, es decir es un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente. Se puede observar el número de archivos en la tabla

Nº	ARCHIVOS
1	Usuarios
2	Expediente
3	Historial
4	Paciente
5	Patología

6	Profesional
---	-------------

Tabla 4.4:Archivos
[Fuente: Elaboración Propia]

4.2.2.5. Número De Interfaces Externas

Son todas las interfaces legibles por la máquina que se utilizan para transmitir información a otro sistema. Una vez que se ha recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad. Se puede observar las interfaces externas en la figura

N°	INTERFACES EXTERNAS
1	Internet
2	Intranet

Tabla 4.5:Interfaces Externas
[Fuente: Elaboración Propia]

4.2.2.6. Ponderación

Ahora reuniremos todos los datos encontrados, para realizar una ponderación.

Para utilizar puntos de función es necesario elegir un criterio de ponderación, que en este caso utilizaremos el factor medio, Como se puede observar en la tabla

PARAMETRO DE MEDICIÓN	DE CUENTA		FACTORES DE PONDERACION				
			SIMPLE	MEDIO	COMPLEJO	=	TOTAL
N° de entradas de usuario	10	*	3	4	6	=	44

N° de salidas de usuario	10	*	4	5	7	=	65
N° de peticiones de usuario	9	*	3	4	6	=	40
N° de archivos	6	*	7	10	15	=	110
N° de interfaces externas	2	*	5	7	10	=	14
Cuenta total							273

Tabla 4.6: Factores de ponderación
[Fuente: Elaboración Propia]

La cuenta total de los puntos de función se debe ajustar en función a las características ambientales del sistema, en la tabla 4.7 se obtiene los valores de ajuste de la complejidad, según las respuestas a las siguientes respuestas evaluados entre 0 y 5

N°	FACTORES DE COMPLEJIDAD	SIN INFLUENCIA	INCIDENTAL	MODERADO	MEDIO		SIGNIFICATIVO	ESSENCIAL	
		0	1	2	3	4	5		
1	Requiere el sistema copias de seguridad y de recuperación fiables							X	5
2	Se requiere comunicación de datos							X	5
3	Existe funciones de procesamiento			X					2

	distribuido							
4	Es critico el rendimiento			X				2
5	Se ejecutara el sistema con un entorno operativo existente.					X		4
6	Requiere el sistema entrada de datos interactivo					X		3
7	Facilidad operativa					X		4
8	Se actualiza los archivo maestros de forma interactiva			X				2
9	Son complejas las entradas, las salidas, los archivos o las peticiones					X		3
10	Procesamiento interno complejo					X		4
11	Diseño del código reutilizable					X		3
12	Facilidad de instalación						X	5
13	Soporta múltiples instalaciones en diferentes sitios						X	5
14	Facilidad de cambios					X		4
FACTOR DE AJUSTES DE COMPLEJIDAD								51

Tabla 4.7: Valores de ajuste de la complejidad
[Fuente: Elaboración Propia]

Reemplazando los datos en la fórmula de punto función, es decir en (1) tendremos:

$$PF = 273 * (0.65 + 0.01 * 51)$$

El punto función del Sistema Web de Control y Seguimiento de Pacientes Internos Caso: Hospital Arco Iris es

$$PF = 316.68$$

Luego, comparando los valores de funcionalidad de sistemas con el punto función máximo que se pueda alcanzar es:

$$PF = 273 * (0.65 + 0.01 * 70)$$

$$PF = 368.55$$

Por tanto la funcionalidad del Sistema Web de Control Y Seguimiento de Pacientes Internos Caso: Hospital Arco Iris será:

$$Funcionalidad = \left(\frac{316.68}{368.55} \right) * 100$$

$$Funcionalidad = 85.92\%$$

4.2.3. Mantenibilidad

Para calcular la mantenibilidad del sistema utilizaremos las medidas directas proporcionadas por la IEEE 982.1 – 198 el cual sugiere un índice de madurez del sistema, que consiste en los cambios que producen en cada versión del producto para lo cual se tiene la siguiente relación.

$$(1)IMS = [M_T - (F_a + F_b + F_c)]/M_T$$

Dónde:

M_T = Numero de módulos en la versión actual.

F_a = Numero de módulos en la versión actual que se han cambiado.

F_b = Numero de módulos en la versión actual que se han añadido.

F_c = Numero de módulos en la versión anterior que se han borrado en la versión actual.

Si el valor del IMS se aproxima a 1, el sistema empieza a estabilizarse.

Entonces los valores encontrados con:

$$M_T = 6$$

$$F_a = 1$$

$$F_b = 0$$

$$F_c = 0$$

Reemplazando estos valores en (1):

$$IMS = [6 - (1 + 0 + 0)]/6$$

$$IMS = 0.83 * 100$$

$$IMS = 83\%$$

Por tanto puede indicarse que el sistema tiende a estabilizarse en un 83% y el 17% indica que aún no es estable a los cambios.

4.2.4. Portabilidad

Es la capacidad del software para ser transferido de un ambiente de operaciones a otro. La portabilidad del software se enfoca en tres aspectos: a nivel de aplicaciones, a nivel de sistema operativo y a nivel de hardware.

A demás de ver estos tres aspectos, también podemos considerar la facilidad de instalación, ajuste y adaptación al cambio. Dado por la siguiente formula:

$$(1)GP = 1 - \left[\frac{CT}{CRD} \right]$$

Dónde:

GP = Grado de portabilidad.

CT = Costo de transportar.

CRD = Costo de re-desarrollo.

Si $GP > 0$, La portabilidad es más rentable que el re-desarrollo.

Si $GP = 1$, la portabilidad es perfecta.

Si $GP < 0$, El re-desarrollo es más rentable que portable.

Reemplazando en (1) tenemos:

$$GP = 1 - [75/2000]$$

$$GP = 0.9625$$

Por lo tanto, el sistema puede transportarse de un entorno a otro en un 96% más rentable que el re-desarrollo.

4.2.4.1. Nivel de aplicaciones

El software es desarrollado en el lenguaje PHP con su gestor de base de datos MySQL, en cuanto a la aplicación es portable ya que el sistema puede ser distribuido en CDs.

4.2.4.2. Nivel de sistema operativo

Para los sistemas operativos de Windows 2000, XP, Vista, 7 es portable nuestro sistema web.

4.2.4.3. Nivel de hardware

A nivel de hardware es portable para todas las computadoras mayores o iguales a la tecnología Pentium IV.

4.2.5. Usabilidad

Grado en el que el software es fácil de usar, y bien reflejado por la: facilidad de comprensión, facilidad de aprendizaje y operatividad. Para medir la usabilidad del sistema se usara las siguientes tres métricas.

4.2.5.1. La complejidad de la descripción

Dada por la fórmula: $(1)x = A/B$

Dónde:

A = Número de funciones (casos de uso) o tipos de funciones descritas en la descripción del producto.

B = Número total de funciones (casos de uso).

Reemplazando en (1) tenemos:

$$x = 8/10$$

$$x = 0.8$$

Por lo tanto, el sistema presenta un 80% de entendimiento por parte de los usuarios finales respecto a la capacidad del producto.

4.2.5.2. Consistencia operacional

Dada por la fórmula: (2) $x = 1 - A/B$

Dónde:

A = Numero de instancias de operaciones con comportamiento inconsistente.

B = Número total de operaciones.

Reemplazando en (2) tenemos:

$$x = 1 - 3/11$$

$$x = 0.72$$

Por lo tanto, el sistema presenta un 72% de no instancias de operaciones con comportamiento inconsistente.

4.2.5.3. Consistencia operacional en el uso

Dada por la fórmula: (3) $x = 1 - A/B$

Dónde:

A = Numero de funciones que el usuario encontró inaceptablemente inconsistentes según son expectativas.

B = Numero de funciones usadas por el usuario durante el periodo de prueba.

Reemplazando en (3) tenemos:

$$x = 1 - 2/11$$

$$x = 0.89$$

Por lo tanto, el usuario encuentra un 10% del sistema inaceptable en el periodo de prueba. De acuerdo a los resultados obtenidos anteriormente el usuario se encuentra satisfecho con la consistencia operacional del uso del sistema.

4.3. Seguridad de Software

Los problemas de seguridad de un sistema web que pueden venir de la configuración de las herramientas que se utilizaron para su desarrollo o pueden ser producto de una falla en el diseño lógico, a menudo es la segunda falla la que ocasiona problemas en el funcionamiento del sistema.

4.3.1. Amenazas.

Existen diversas amenazas, a continuación se mencionan las más comunes:

- Ingreso de usuario no valido.
- Control de acceso roto.
- Administración de sesión y autenticación rota.
- Desbordamiento de buffer.
- Inyección de código.
- Manejo de errores inadecuado.
- Almacenamiento inseguro.
- Administración de configuración insegura.

4.3.2. Guías de seguridad

Estos son algunos principios de seguridad para el diseño de aplicaciones web.

- Validar todas las entradas y salidas.
- Mantener un esquema de seguridad simple.
- Manejar las fallas y errores de forma adecuada.
- Utilizar solo componentes de confianza.
- Controlar las excepciones.

4.3.3. Tipos de seguridad para sistemas web .

Hay cuatro tipos de seguridad en los sistemas web:

- Seguridad en el cliente.
- Seguridad en el servidor.
- Seguridad en las comunicaciones.
- Seguridad en la aplicación.

4.3.3.1. Seguridad en el cliente

Uno de los mecanismos de seguridad que se implementan son las validaciones por el lado del cliente.

Existen mecanismos de validación provistas por las herramientas que utilizamos para hacer la aplicación, en el caso PHP se tienen los controladores de validación para la información introducida por el cliente, estas validaciones son realizadas antes de que la información introducida llegue al servidor, esto evita que se envíen datos incorrectos al servidor, además se ahorra tiempo, ya que si la información es incorrecta simplemente no se envía al servidor.

4.3.3.2. Seguridad en el servidor

La validación del lado del cliente no es suficiente, también deben realizarse otro tipo de controles por el lado del servidor, ya sea del servidor de aplicaciones o del servidor de base de datos.

➤ Seguridad en el servidor de aplicaciones.

Un servidor de aplicaciones proporciona muchos servicios y no todos son necesarios para el funcionamiento de la aplicación web.

Es conveniente deshabilitar lo que no se necesite en el servidor de aplicaciones, para ello debe configurarse adecuadamente el servidor de aplicaciones.

➤ Seguridad en el servidor de base de datos.

Existen muchos problemas a nivel de base de datos, uno de ellos es la Inyección MySQL (lenguaje de consulta estructurada), son los ataques realizados contra base de datos.

En este caso, un usuario utiliza debilidades en el diseño de base de datos o de la página web para extraer información o más aun, para manipular información dentro de la base de datos. Una forma de evitar este tipo de debilidades es restringiendo los caracteres que el usuario introduce, por ejemplo las comillas, los puntos y coma y otros.

También está el acceso no autorizado a la información. Esto podría solucionarse asignando correctamente los roles que tiene cada usuario en el sistema. El acceso de los usuarios a los formularios web es encargado desde la base de datos, cuando un usuario se autentica el sistema se le da acceso solo a los formularios asignados para él.

5. ANÁLISIS DE COSTO/BENEFICIO

5.1. Introducción

Es una parte muy importante de todo análisis de sistemas. El propósito, es mostrar a los usuarios del nuevo sistema, al igual que a otros grupos de administradores de la organización, que los beneficios que se espera obtener con el nuevo sistema superan a los costos esperados.

En las secciones siguientes examinaremos diversos aspectos de los cálculos de costo/beneficio:

- Análisis de costos.
- Análisis de beneficios.
- Como expresar los ahorros.
- Análisis de riesgo.

5.2. Análisis De Costos

Se debe calcular todos los costos anticipados asociados con el sistema. Para determinar el costo total del proyecto se tomara en cuenta los siguientes costos:

- Costo del software desarrollado.
- Costo de implementación del sistema.
- Costo de la elaboración del proyecto.

5.2.1. Costo Del Software Desarrollado

Para determinar el costo del software desarrollado, se utilizara el modelo constructivo COCOMO II, orientado a los puntos de función.

Estimación de puntos de función:

PARAMETRO DE MEDICION	CUENTA	FACTOR PONDERACION	DE TOTAL
Nº de entradas de usuario	11	4	44
Nº de salidas de usuario	13	5	65
Nº de peticiones de usuario	10	4	40
Nº de archivos	11	10	110
Nº de interfaces externas	2	7	14
Cuenta total			273

Tabla 5.1: Calculo de punto funciono ajustado
[Fuente: Elaboracion Propia]

Calculo de valores de ajuste de la complejidad tomando los valores de la tabla 5.2, en la cual se determina la complejidad.

$$\text{factor de ajuste} = (0.65 + 0.01 * 51)$$

$$\text{factor de ajuste} = 1.16$$

FACTORES DE COMPLEJIDAD	VALOR
Requiere el sistema copias de seguridad y de recuperación fiables.	5

Se requiere comunicación de datos.	5
Existen funciones de procesamiento distribuido.	2
Es crítico el rendimiento.	2
Se ejecutara el sistema con un entorno operativo existente y fuertemente utilizado.	4
Requiere el sistema entrada de datos interactiva.	3
Facilidad operativa.	4
Se actualiza los archivos maestros de forma interactiva.	2
Son complejos las entradas, las salidas, los archivos o las peticiones.	3
Procesamiento interno complejo.	4
Diseño del código reutilizable.	3
Facilidad de instalación.	5
Soporta múltiples instalaciones en diferentes sitios.	5
Facilidad de cambios.	4
Total	51

Tabla 5.2:Calculo de punto función ajustado
[Fuente: Elaboracion Propia]

El cálculo de los puntos de función se basa en la fórmula:

$$PF = Cuenta\ total * Factor\ de\ ajuste$$

$$PF = 273 * 1.16$$

$$PF = 316.68$$

Conversión de los puntos de función a KDLC:

Ahora convertiremos los PF a miles de líneas de código. Para ello veremos la tabla 5.3.

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
ANSI Basic	5	64
Java	6	53
ANSI Cobol	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

Tabla 5.3: Conversión de puntos función a KLDC
[Fuente: Elaboración Propia]

$$LDC = PF * Factor\ LCD/PF$$

$$LCD = 316.68 * 36$$

$$LDC = 11400.48$$

$$KLDC = 11.40$$

Aplicando las formulas básicas del esfuerzo, tiempo calendario y personal requerido.

Las ecuaciones del COCOMO tienen la siguiente forma:

$$E = a_b(KLDC)b_b$$

$$D = c_b(E)d_b$$

Dónde:

E: Esfuerzo aplicado en personas por mes.

D: Tiempo de desarrollo en meses.

KLDC : Número estimado de líneas de código distribuidas(en miles)

PROYECTO DE SOFTWARE	a_b	b_b	c_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Sema-acoplado	3	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Tabla 5.4: Coeficientes a_b y b_b
[Fuente: Elaboracion Propia]

En la Tabla 5.4 se muestra los tipos de proyecto de software. Como este es un proyecto intermedio, en tamaño y complejidad, se elige sema- acoplado.

$$E = 3 * (11.40)^{1.12}$$

$$E = 45.79$$

$$D = 2.5 * (45.79)^{0.35}$$

$$D = 9.53$$

El personal requerido se obtiene con la siguiente formula:

$$\text{Numero programadores} = E/D$$

$$\text{Numero programadores} = 45.79/9.53$$

$$\text{Numero programadores} = 4.80 = 5$$

El salario de un programador aproximadamente es de 400\$, cifra que se tomara en cuenta para la estimación siguiente:

$$\text{Costo software} = \text{Numero de programadores} * \text{Salario de programador}$$

$$\text{Costo software} = 5 * 400$$

$$\text{Costo software} = 2000\$$$

5.2.2. Costo De La Implementación Del Proyecto

Como el Hospital Arco Iris ya cuenta con sus servidores de aplicaciones y base de datos completamente instalados y configurados, por lo que el costo de implementación es cero.

5.2.3. Costo De Elaboración Del Proyecto

Se refiere a los costos de estudio del sistema, en la etapa de análisis estos costos se representa en la siguiente tabla:

DESCRIPCION	COSTO TOTAL(\$us)
Análisis y diseño del proyecto	250
Bibliografía	40
Material de escritorio	30
Otros	30
Total	350

Tabla 5.5: Costo de elaboración del proyecto
[Fuente: Elaboracion Propia]

5.2.4. Costo Total

El costo total es la suma de software de desarrollo y el costo de elaboración del proyecto, que se puede observar en la siguiente tabla:

DESCRIPCION	COSTO TOTAL(\$us)
Costo de software de desarrollo	2000
Costo de implementación	0
Costo de elaboración del proyecto	350
Total	1850

Tabla 5.6: Costo total del proyecto
[Fuente: Elaboracion Propia]

5.3. Análisis de Beneficios

Es mucho más difícil calcular los beneficios de un nuevo sistema de información que calcular su costo. La labor más grande al llevar a cabo un cálculo de costo-beneficio será acorralar a los usuarios y hacer que identifiquen beneficios tangibles que pueden medirse y calcularse de manera cuantitativa.

Los beneficios para el presente proyecto son de tipo intangible. Para realizar el análisis de beneficios se utilizaran cinco criterios de evaluación. Los cuales permiten evaluar el sistema diseñado respecto al sistema anterior estos son:

- Incremento de velocidad en los procesos.
- Capacidad con el volumen de información.
- Control de procesos.
- Integración de la información.
- Información para la toma de decisiones.

5.4. Valor Neto Actual

El valor neto actual (VAN), es un indicador financiero que mide los flujos de los futuros ingresos y egresos que tendrá un proyecto, para determinar, si luego de descontar la inversión inicial, nos quedaría alguna ganancia. Si el resultado es positivo, el proyecto es viable.

Basta con hallar VAN de un proyecto de inversión para saber si dicho proyecto es viable o no.

El VAN también nos permite determinar cuál proyecto es el más rentable entre varias opciones de inversión. Incluso, si alguien nos ofrece comprar nuestro negocio, con este indicador podemos determinar si el precio ofrecido está por encima o por debajo de lo que ganaríamos de no venderlo.

La fórmula del VAN es:

$$VAN = BNA - Inversion$$

Donde el beneficio neto actualizado(BNA) es el valor actual del flujo de caja o beneficio neto proyectado, el cual ha sido actualizado a través de una tasa de descuento.

La tasa de descuento (TD) con la que se descuenta el flujo neto proyectado, es la tasa de oportunidad, rendimiento o rentabilidad mínima, que se espera ganar; por lo tanto, cuando la inversión, resulta mayor que el BNA (VAN negativo o menor que 0) es porque no se ha cumplido con dicha tasa. Y cuando el BNA es igual a la inversión (VAN igual a 0) es porque se ha cumplido con dicha tasa. Y cuando el BNA es mayor que la inversión es porque se ha cumplido con dicha tasa y además, se ha generado una ganancia o beneficio adicional.

$$VAN > 0 \xrightarrow[\text{mayor a } 0]{} \textit{El proyecto es rentable}$$

$$VAN = 0 \xrightarrow[\text{igual a } 0]{} \textit{El proyecto es rentable, tambien por que ya esta sumada la ganancia}$$

$$VAN < 0 \xrightarrow[\text{menor a } 0]{} \textit{El proyecto no es rentable}$$

Entonces para hallar el VAN se necesitan:

- Tamaño de la inversión.
- Flujo de caja neto proyectado.
- Tasa de descuento.

Veamos un ejemplo:

El proyecto tiene una inversión de 1200 \$us. Y una tasa de descuento (TD) de 10%, por defecto.

Hallando el VAN:

$$VAN = BNA - Inversion$$

$$VAN = \frac{1450}{(1 + 0.14)^1} + \frac{1450}{(1 + 0.14)^2} + \frac{1450}{(1 + 0.14)^3} + \frac{1450}{(1 + 0.14)^4} + \frac{1450}{(1 + 0.14)^5} - 1450$$

$$VAN = 1425.69 - 1450$$

$$VAN = 151.69$$

5.5. Tasa interna de retorno

La TIR es la tasa de descuento (TD) de un proyecto de inversión que permite que el BNA sea igual a la inversión (VAN igual a 0). La TIR es la máxima TD que puede tener un proyecto para que sea rentable, pues una mayor tasa ocasionaría que el BNA sea menor que la inversión (VAN menor que 0).

Entonces para hallar la TIR se necesitan:

- Tamaño de inversión.
- Flujo de caja neto proyectado.

Veamos un ejemplo:

El proyecto tiene una inversión de 1200.

Para hallar la TIR hacemos uso de la fórmula del VAN, solo que en vez de hallar el VAN (el cual reemplazamos por 0), estaríamos hallando la tasa de descuento:

$$VAN = BNA - Inversion$$

$$0 = \frac{1200}{(1+i)^1} + \frac{1200}{(1+i)^2} + \frac{1200}{(1+i)^3} + \frac{1200}{(1+i)^4} + \frac{1200}{(1+i)^5} - 1200$$

$$i = 10\%$$

$$TIR = 10\%$$

Si esta tasa fuera mayor, el proyecto empezaría a no ser rentable, pues el BNA empezaría a ser menor que la inversión. Y si la tasa fuera menor (como en el caso del ejemplo del VAN donde la tasa es de un 10%), a menor tasa, el proyecto sería cada vez más rentable, pues el BNA sería cada vez mayor que la inversión.



6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Las conclusiones a las que se llegó en el presente Proyecto de Grado y conforme a las actividades definidas para el análisis e implementación del Sistema Web de Seguimiento y Control de Pacientes Internos Caso: Hospital Arco Iris, se detallan a continuación:

- Se implementó el Sistema Web de Seguimiento y Control de Pacientes Internos para el hospital Arco Iris, este se concluyó de manera satisfactoria, dicho sistema optimizara el registro y manipulación de los historiales clínicos así de esta manera mejora la atención a los pacientes neonatos.
- Se implementó los módulos de registro modificación y actualización de historiales clínicos de pacientes neonatos, los cuales se almacenan en la base de datos del sistema.
- Se proporciona información actualizada del historial del paciente.
- Se implementó digitalmente los formularios de la Unidad de Cuidados Intensivos Neonatales.
- Se ha reducido el tiempo empleado el tiempo empleado para búsqueda de información requerida sobre historiales clínicos.
- Se ha establecido un medio adecuado para determinar el estado actual del paciente.
- Se usó la metodología AUP para el desarrollo del sistema, comprobando que la ser amplia se genera gran cantidad de información, además de las iteraciones que se adoptan en esta metodología hicieron posible la realización del producto.
- El Sistema Web de Seguimiento y Control de Pacientes Internos, paso por varias fases antes de tener un resultado final de acuerdo a los requerimientos del usuario.

Por lo ya mencionado el Sistema Web de Seguimiento y Control de Pacientes Internos para el Hospital Arco Iris, ha cumplido satisfactoriamente con los objetivos propuestos en la fase inicial.

6.2. Recomendaciones

Para ampliar el presente proyecto de grado, se recomienda lo siguiente:

- Se deberá incorporar normas y políticas de uso del sistema.
- Se deberá cambiar la contraseña semanalmente o mensualmente para dar mejor seguridad al sistema.
- Se recomienda integrar el Sistema Web de Seguimiento y Control de Pacientes Internos con el sistema general del Hospital Arco Iris para llevar un mejor control de ambas instancias ya que están estrechamente relacionadas.
- Implementar módulos más detallados acerca de la información de los médicos y sus especialidades.



BIBLIOGRAFÍA

TEXTO

- [Caro, 1996] Caro, T. (1996). Ingeniería de Requisitos de Software. Madrid: Servicio de Publicaciones de la Facultad de Informática, Universidad Politécnica.
- [Kendall, 1997] Kendall, K. & (1997). Análisis y Diseño de Sistemas. New York: Services of New England Inc.
- [Fowler & Kendall, 1997] Fowler, M. & Kendall, S. (1997). UML Gota a Gota. México DF. : Wesley Longman.
- [Prentice, 1999] Prentice, L. (1999). UML y Patrones Introducción al Análisis y Diseño Orientado a Objetos.
- [Piatini, 1999] Piatini, M. (1999). Fundamentos y Modelos de Base de Datos (2da edición). México DF. Alfaomega.

[Booch & Jacobson , 1999]

Booch, G., J., R., & Jacobson. (1999).
Proceso Unificado de Desarrollo de
Software.

[Minguez & Garcia, 2000]

Minguez, D., & Garcia, E. (2000).
Metodología para el Desarrollo de
Aplicaciones Web: UWE.

[Idaberto, 2000]

Idaberto, C. (2000). Administración de
Recursos Humanos. Bogota: ME Graw
HILL.

[Mendoza, 2008]

Mendoza, M. S. (2008). Analiysis
Service. Perú: Megabyte.

[Rumbaugh, Jacobson & Booch, 2000]

Rumbaugh, J., Jacobson, I., & Booch, G.
(2000). El lenguaje Unificado de Modelo.
Madrid: Pearson.

[Pressman, 2006]

Pressman, R. (2006). Ingeniería de Software un Enfoque Practico (sexta edición ed.), México DF.

SITIOS DE INTERNET

[Ambler, 2000]

Ambler, S. (12 de marzo de 2010). Metodologías Proceso Ágil. Recuperado el 15 de Junio de 2011, de Metodologías del Proceso Ágil :
www.agilemodeling.com/essays/agileModelingRUP.html

[Copley, 2010]

Copley, S. (25 de enero de 2010). Implementación del Nuevo Sistema. Recuperado el 5 de Noviembre de 2011, de
<http://www.igcseict.info/theory/8/implement/>

[ICT.2011]

Academia Latinoamericana de Seguridad Informática. (28 de enero de 2011). Implementación y Plan de Seguridad.

Recuperado el 1 de Noviembre de 2011,
de

[http://piramidedigital.com/Documentos/ICT
/pdictseguridadinformaticaimplementacion](http://piramidedigital.com/Documentos/ICT/pdictseguridadinformaticaimplementacion)

[IEEE, 2011]

IEEE, Norma IEEE 9000 estándar.
Recuperado el 15 noviembre del 2011, de
<http://www.ieee.org.com>

[Koch, 2011]

Unit of Programming and software
engineering. (11 de junio de 2011). UWE-
UML. Based Web Engineering.
Recuperado el 10 de agosto de 2011, de
<http://uwe.pet.ifi.lmu.de/examples.html>

[AISI, 2007]

Asociación de Investigación de Software
Inteligente AISI. (2007). Ingenierías Web
Recuperado el 4 de agosto de 2001, de
<http://www.AISI.html>