



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

DISEÑO Y DESARROLLO DE SISTEMA DE TELEMEDICINA CON PROTOTIPO FUNCIONAL PARA EL ÁREA DE OBSTETRICIA.

Tesis para optar al título de:
Ingeniero Civil en Informática

Profesor Patrocinante:
Sr. Raimundo Vega Vega
Estadístico
Master en Informática
Doctor en Informática

Profesores Co- Patrocinante:
Dr. Francisco Guerra Bellone
Médico Cirujano
Gineco-Obstetra

Profesor Informante:
Sr. Jorge Arancibia Muñoz
Ingeniero Civil Electrónico
Magíster Ingeniería en Informática (c).

LEONARDO J. FAÚNDEZ MARTÍNEZ
NICOLÁS E. HERNÁNDEZ NUÑEZ
VALDIVIA - CHILE

2007



Universidad Austral de Chile

Instituto de Informática

Valdivia 19 de enero de 2007

Sr.

Juan Pablo Salazar F.

Director

Escuela de Ingeniería Civil en Informática

Presente

Estimado Director,

Informo a usted que he revisado el trabajo de titulación "DISEÑO Y DESARROLLO DE SISTEMA DE TELEMEDICINA CON PROTOTIPO FUNCIONAL PARA EL ÁREA DE OBSTETRICIA", de los alumnos egresado Sr(s) Leonardo J. Faúndez Martínez y Nicolás Hernández Nuñez

Por lo anterior, pude comprobar que los objetivos del trabajo de titulación se cumplieron a cabalidad, dando como resultado una aplicación que soluciona un problema concreto en el área de la salud. La especificación de requisitos, análisis y diseño de la aplicación se hizo siguiendo estándares y desarrollando artefactos propios de la Ingeniería de software, lo que permite observar con claridad el desarrollo de la aplicación.

La presentación del trabajo de titulación pudo haber sido mejor logrado como también haber innovado en la problemática planteada.

Por los fundamentos dado, he resuelto calificar el trabajo de titulación con nota 6,2 (seis como 2).

Reciba un cordial saludo,

Dr. Raimundo Vega

Académico

Instituto de Informática

Dirección : General Lagos 2086 – Campus Miraflores- Valdivia – Chile

Fono: 56 63 221427

Fax: 56 63 293115

email: instituto@inf.uach.cl

De: Francisco Guerra Bellone
Médico Cirujano

A: Juan Pablo Salazar Fernández
Director Escuela de Informática

MOTIVO:

INFORME TRABAJO DE TITULACIÓN

Nombre del Trabajo de Titulación:

**DISEÑO Y DESARROLLO DE SISTEMA DE TELEMEDICINA CON
PROTOTIPO FUNCIONAL PARA EL ÁREA DE OBSTETRICIA**

Nombre del Alumno:

- Leonardo J. Faúndez Martínez
- Nicolás E. Hernández Núñez

Nota:

6,8

.....
(en números)

Seis coma ocho

(en letras)

FUNDAMENTO DE LA NOTA:

El trabajo desarrollado cumple con los objetivos propuestos, dando solución a los problemas de almacenamiento de fichas clínicas de pacientes y exámenes en un medio centralizado, que hasta la fecha se encontraba en Sistemas heterogéneos y difíciles de integrar.

Una de las funcionalidades más notables, fundamento principal de la calificación, es la posibilidad de definir nuevos exámenes sin la intervención de programadores, lo que hace robusto y dinámico al Sistema implementado, permitiendo a los propios médicos, con los privilegios suficientes, incorporar nuevas funcionalidades oportunamente.

Por otro lado, la posibilidad de acceder a módulos comunes a todas las especialidades es de vital importancia para la publicación oportuna de comunicados de carácter general o información relevante del quehacer médico.



Valdivia, 28 marzo 2007.-

DE: Jorge Arancibia M.

A: DIRECTOR ESCUELA INGENIERIA CIVIL INFORMATICA

MOTIVO: INFORME TRABAJO DE TITULACION

Nombre de Trabajo de Titulación: **DISEÑO Y DESARROLLO DE
SISTEMA DE TELEMEDICINA CON PROTOTIPO FUNCIONAL
PARA EL ÁREA DE OBSTETRICIA**

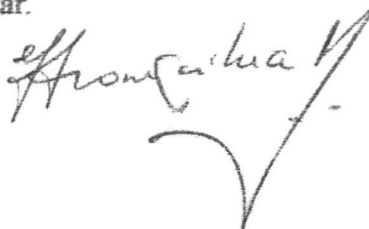
Nombre de Alumnos: LEONARDO J. FAÚNDEZ MARTÍNEZ
NICOLÁS E. HERNÁNDEZ NUÑEZ

Nota: 5,6

Cinco coma seis.

Por cumplimiento de objetivos propuestos y dar satisfacción de un sistema requerido.
El trabajo aborda con originalidad la solución a nivel de prototipo y es adecuada la
interpretación de datos y las conclusiones.

Es todo lo que puedo informar.

A handwritten signature in dark ink, appearing to read 'Jorge Arancibia M.', followed by a large, stylized checkmark or flourish.

I Agradecimientos

En primer lugar queremos agradecer a nuestras familias, pilar fundamental sobre el que se sostienen nuestras vidas, nada sería sin ellas, nada valdría sin ellas.

En segundo lugar queremos agradecer a nuestra Universidad, a su equipo docente, nuestros respetados profesores. Ellos nos han entregado la base que nos define como profesionales, una llave maestra de las puertas del futuro.

Finalmente, agradecer a la otra mitad del equipo, mi gran amigo, esas interminables noches de trabajo no habrían dado frutos sin su apoyo. Los logros ahora son nuestros, tomémoslos como referente para fijarnos nuevas metas, que nos permitan avanzar mucho más en este eterno camino de la vida.

II Índice de contenidos

I	Agradecimientos.....	2
II	Índice de contenidos.....	3
III	Índice de figuras.....	5
IV	Índice de tablas.....	7
V	Resumen.....	8
VI	Summary.....	9
1.	Introducción.....	10
	1.1 Objetivos.....	11
	1.1.1 Objetivo general.....	11
	1.1.2 Objetivos específicos.....	11
2.	Avances en Telemedicina.....	13
	2.1 Factibilidad de la implementación.....	14
3.	Proceso de desarrollo.....	15
	3.1 Ciclo de vida.....	15
	3.1.1 Prototipo parchado.....	15
	3.2 Especificación de requisitos.....	16
	3.2.1 Descripción problema y situación actual.....	16
	3.2.2 Área de obstetricia.....	16
	3.2.3 Requisitos funcionales.....	19
	3.2.4 Requisitos no funcionales.....	19
	3.2.5 Requerimientos de hardware y software.....	20
	3.2.5.1 Análisis de herramientas y plataformas.....	20
	3.2.5.2 Requerimientos de software.....	24
	3.2.5.3 Requerimientos de hardware.....	26
	3.3 Análisis.....	27
	3.3.1 Sistema general.....	27

3.3.1.1 Diagrama entidad relación.....	28
3.3.1.2 Modelo de datos para sistema genérico.....	29
3.3.1.3 Diagramas de flujo de datos.....	31
3.4 Diseño.....	41
3.4.1 Generalización.....	42
3.4.2 Usuarios del sistema.....	44
3.4.2.1 Privilegios y perfiles.....	45
3.4.3 Diseño de interfaces.....	48
3.5 Aseguramiento de la calidad del software.....	50
3.6 Implementación, validación y prueba.....	55
3.6.1 Implementación	55
3.6.2 Validación y prueba.....	56
4. Conclusiones.....	58
5. Glosario.....	60
6. Referencias.....	65
7. Anexos.....	66
7.1 Estimación de costo esfuerzo según COCOMO II.....	66
7.2 Descripción de tablas y sus campos.....	78
7.3 Detalles de codificación y programación.....	83
7.3.1 Funciones y procedimientos.....	83
7.3.2 Estructura del sistema.....	86
7.3.3 Reglas generales de programación y diseño.....	87
7.3.4 Concepto de secciones, componentes y plantillas.....	88
7.4 Módulo interconsultas.....	90
7.4.1 Modelo de datos.....	92
7.4.1.1 Descripción de las tablas y sus campos.....	92
7.4.1.2 Estructura y relaciones entre tablas.....	93

7.4.2 Diagrama de flujo de datos.....	93
7.4.2.1 Diagrama nivel 0.....	93
7.4.2.2 Diagrama nivel 1.....	94
7.4.2.3 Diagramas nivel 2.....	94
7.5 Guía de operación.....	97

III Índice de figuras

Figura 3.1: Diagrama de contexto del sistema general.....	27
Figura 3.2: Diagrama entidad-relación del sistema general.....	28
Figura 3.3: Modelo de datos del sistema genérico.....	29
Figura 3.4: Diagrama nivel 0.....	31
Figura 3.5: Diagrama nivel 1, módulo validación.....	32
Figura 3.6: Diagrama nivel 1, módulo telemedicina.....	33
Figura 3.7: Diagrama nivel 2, módulo noticias.....	34
Figura 3.8: Diagrama nivel 2, módulo foro.....	35
Figura 3.9: Diagrama nivel 2, módulo publicaciones.....	36
Figura 3.10: Diagrama nivel 2, módulo fichas clínicas.....	37
Figura 3.11: Diagrama nivel 2, módulo administración episodios y exámenes.....	39
Figura 3.12: Diagrama nivel 2, módulo administración secciones y componentes....	40
Figura 3.13: Esquema básico de relaciones entre bases de datos.....	41
Figura 3.14: Diagrama de relaciones básicas	42
Figura 3.15: Modelo general del sistema telemedicina.....	42
Figura 3.16:Modelo detallado.....	43
Figura 3.17: Relación entre módulos, estructura del sistema.....	43
Figura 7.1: Resumen factores de escala COCOMO II.....	67
Figura 7.2: Resumen drivers de coste COCOMO II.....	69
Figura 7.3: Pantalla principal COCOMO II.....	73

Figura 7.4: Detalle de los resultados	75
Figura 7.5: Distribución global del proyecto.....	76
Figura 7.6: Diagrama entidad-relación del módulo interconsultas.....	91
Figura 7.7: Modelo de datos del módulo interconsultas.....	92
Figura 7.8: Diagrama nivel 0, módulo interconsulta.....	93
Figura 7.9: Diagrama nivel 1, módulo interconsulta.....	94
Figura 7.10: Diagrama nivel 2, solicitud de consulta.....	94
Figura 7.11: Diagrama nivel 2, procesar interconsulta.....	95
Figura 7.12: Pantalla principal de acceso al sistema telemedicina.....	97
Figura 7.13: Pantalla registro de usuarios.....	98
Figura 7.14: Pantalla principal de acceso a la especialidad, módulo noticias.....	99
Figura 7.15: Pantalla módulo de fichas clínicas, buscador de pacientes.....	100
Figura 7.16: Pantalla módulo de fichas clínicas, datos del paciente.....	101
Figura 7.17: Pantalla módulo de fichas clínicas, historial del paciente.....	101
Figura 7.18: Pantalla módulo de fichas clínicas, agregar episodio.....	102
Figura 7.19: Pantalla módulo de fichas clínicas, agregar examen.	102
Figura 7.20: Pantalla módulo de fichas clínicas, examen para completar.	103
Figura 7.21: Pantalla módulo de administración, validar usuario.....	103
Figura 7.22: Pantalla módulo de administración, menú opciones.....	104
Figura 7.23: Pantalla módulo de administración, episodios.....	104
Figura 7.24: Pantalla módulo de administración, editar episodios.....	104
Figura 7.25: Pantalla módulo de administración, crear episodios.....	105
Figura 7.26: Pantalla módulo de administración, crear examen.....	106
Figura 7.27: Pantalla módulo de administración, crear secciones.	106
Figura 7.28: Pantalla módulo de administración, crear componentes.....	107
Figura 7.29: Pantalla módulo de administración, plantilla examen.....	107

IV Índice de Tablas

Tabla 3.1: Tabla genérica de privilegios y perfiles..... 45

Tabla 3.2: Privilegios del administrador del sistema..... 46

Tabla 3.3: Privilegios administrador de especialidad..... 46

Tabla 3.4: Privilegios médico especialista..... 47

Tabla 3.5: Privilegios médico general..... 47

Tabla 3.6: Privilegios de Secretario (a)..... 48

Tabla 7.1: Riesgos..... 71

Tabla 7.2: Costo por módulo..... 72

Tabla 7.3: Resultados por fase..... 76

Tabla 7.4: Plan de trabajo..... 77

V Resumen

El proyecto consiste en el diseño de una solución Web que sirve de apoyo a los distintos procesos internos involucrados en el área de medicina, factibles de ser automatizados u optimizados gracias al uso de tecnología informática. Entre ellos se destacan: apoyo a la atención médica y diagnósticos, almacenamiento centralizado de fichas clínicas de pacientes, imágenes radiográficas y ecográficas, información de investigación, estudios de laboratorio y documentación de manejo interno. Para tal efecto, se ha diseñado una plataforma de comunicación en tiempo real y asíncrono.

Se implementó un módulo para el área de Obstetricia, el que fue puesto a prueba en el Hospital Base de Valdivia, con el objeto de recabar la mayor cantidad de información posible sobre el quehacer médico y así lograr el diseño de un sistema integral que pueda ser adoptado por cualquier entidad hospitalaria en el ámbito provincial, regional o nacional. Dicho módulo fue realizado en lenguaje PHP y base de datos Oracle 8i (base de datos utilizada en el Hospital Base de Valdivia), se han implementado fichas clínicas digitales usando como plantilla fichas estándar del Ministerio de Salud Chileno, el sistema además permite el despliegue de imágenes, facilitando así el acceso a éstas desde localidades apartadas.

Es importante mencionar que el sistema fue construido a partir de módulos, los que se integrarán en las distintas etapas de desarrollo, es decir, se trata de un proyecto de integración de tecnologías, y no de programación y desarrollo de protocolos de comunicación y/o seguridad a bajo nivel.

VI Summary

This project consist of the design of a Web solution, that it serves as a support to the internal processes involved into medical field, feasible of being automated or being optimized thanks to the use of computer science technology. Among them they stand out: support to the medical attention and centralized clinical card diagnoses, storage of patients, x-ray images, information of investigation, studies of laboratory and documentation of internal handling. For such effect, a platform of communication in real time and asynchronous has been designed.

One has implemented a module for the area of Obstetrics, the one that was put on approval in the Hospital Base of Valdivia, with the intention of successfully obtaining the greater amount of possible information on the medical task and thus to obtain the design of an integral system that can be adopted by any hospitable organization in the provincial, regional or national scope. This module was made in language PHP and data base Oracle 8i (data base used in the Hospital Base of Valdivia), has implemented digital clinical cards using as it soles standard cards of the Chilean Ministry of Health, the system in addition allows to the unfolding of images, facilitating therefore the access to these from separated localities.

It is important to mention that the system was constructed from modules, those that was integrated in the different stages from development, that is to say, one is a project of integration of technologies, and not of programming and development of communication protocols and/or security at low level.

1. Introducción

En los últimos años, la medicina en Chile ha dado gran importancia a la atención primaria en salud, generando políticas que entregan recursos para infraestructura, capacitación de personal, mejorar la gestión y calidad de servicio, logrando una mayor satisfacción para los usuarios.

A nivel nacional, particularmente en las comunas de la provincia de Valdivia, la atención primaria en salud, adolece de la falta de médicos especialistas, en consecuencia, los diagnósticos realizados por el personal disponible en esos centros hospitalarios no tiene el respaldo de estos. Por otra parte, la comunicación con los hospitales base se realiza a través de documentos impresos, telefónicamente o por correo electrónico en un medio informal. La información del área, tal como: registros médicos, radiografías, ecografías, exámenes, informes, respuestas a interconsultas, etc. se encuentran en sistemas y formatos distintos, difíciles de integrar, haciendo casi imposible su disponibilidad en forma remota y / o cuando es requerida.

Esta problemática implica trasladar pacientes a centros especializados, lo que conlleva el desarraigo del paciente de su núcleo familiar, retraso en el intercambio de informes que responden a interconsultas o exámenes solicitados, lo cual, genera un retardo en la toma de decisiones médicas, aumento de costos, traba de la gestión en salud y riesgo vital para el paciente en casos de urgencia.

Considerando lo anteriormente expuesto, se propone el desarrollo de una plataforma virtual que permita una mayor conectividad entre los diferentes niveles de atención de salud utilizando Internet. De esta forma contar con especialistas que ayuden en forma remota a las acciones de salud en la atención primaria. Tener en línea la posibilidad de envío de informes médicos (exámenes imagenológicos, de laboratorio, etc.), solicitud y respuestas de interconsultas, posibilidad de foros y Chat, acceso a las guías y normas, etc.

Para finalizar, la motivación que llevó al desarrollo del trabajo fue la de aportar, mediante los conocimientos adquiridos durante los años de estudio, a la población local en un área de gran relevancia como es el Sistema de Salud. Con esto, se reafirma el compromiso social, propuesto como competencia para el Ingeniero Civil en Informática y se comparte la misión propuesta a nivel de Facultad de Ciencias de la Ingeniería.

1.1. Objetivos

1.1.1. Objetivo general

Diseñar una plataforma de software que permita a los profesionales de la salud apoyar la atención de pacientes en forma local y remota disponiendo de un medio formal para el almacenamiento de información interna e intercambiar opiniones y experiencias sobre temas de interés.

1.1.2. Objetivos específicos

- Investigar y recopilar información referida a la atención hospitalaria.
- Investigar herramientas informáticas adecuadas para la realización del proyecto.
- Diseñar e implementar una solución informática que permita automatizar –en parte- el sistema de atención hospitalaria al nivel del profesional de salud con herramientas tales como: fichas electrónicas en línea y acceso a información centralizada; permitiéndole mejorar y agilizar dicho proceso.
- Diseñar e implementar un sistema mediante el cual los profesionales de la salud puedan apoyar la atención hospitalaria en forma remota, realizando diagnósticos y / o interconsultas.

- Diseñar e implementar un sistema central de noticias que permita mantener un registro de las políticas y normas a adoptarse en los distintos Centros de Salud, con la finalidad de agilizar su difusión.
- Diseñar e implementar un entorno Web para almacenar información y documentación, en un medio centralizado y accesible desde cualquier computador perteneciente a la red de salud.
- Validar prototipo funcional del sistema para el área de Obstetricia.

2. Avances en Telemedicina

Telemedicina significa Medicina practicada a distancia, incluye diagnóstico y tratamiento, como también la educación médica. Es un recurso tecnológico que posibilita la optimización de los servicios de atención en salud, ahorrando tiempo y dinero y facilitando el acceso a zonas distantes para tener atención de especialistas. Otra de las utilidades que presta el uso de la transmisión de datos médicos sobre redes adecuadas, es la educación, donde los alumnos de medicina y enfermería pueden aprender semiología remotamente, apoyados por su profesor y con la presencia del paciente. Así podemos definir los siguientes servicios, que telemedicina presta:

- Servicios complementarios e instantáneos a la atención de un especialista (obtención de una segunda opinión).
- Diagnósticos inmediatos por parte de un médico especialista en un área determinada.
- Educación remota de alumnos de las escuelas de enfermería y medicina.
- Servicios de archivo digital de exámenes radiológicos, ecografías y otros.
- Almacenamiento de fichas clínicas de pacientes y exámenes asociados, permitiendo el acceso remoto.

Todo esto se traduce en una disminución de tiempo entre la toma de exámenes y obtención de resultados, o entre la atención y el diagnóstico certero del especialista, reduciendo costos de tiempo y dinero.” [REF, 04].

2.1. Factibilidad de implementación

En la actualidad existe un número limitado de aplicaciones que abordan la telemedicina desde un punto de vista integral, que puedan ser usados por todos los profesionales de la medicina proveyendo un entorno de trabajo profesional, con canales de comunicación orientados al mundo médico, eliminando total o parcialmente el uso de medios de comunicación convencionales tales como teléfono, fax, e-mail, etc.

Gran parte de las soluciones existentes en el mercado apuntan a problemáticas específicas y solamente se usan en algunos Centros de Salud. Siendo totalmente desconocidas para otros y de un elevado costo en la mayoría de los casos.

Como ejemplo se cita el sistema Meditux (<http://sourceforge.net/projects/meditux>), desarrollado en Servlets Java, aunque actualmente el proyecto está inactivo se dispone de su 7ª versión. Este sistema (con licencia GNU) esta orientado específicamente al apoyo de las Unidades de Tratamiento Intensivo (UCI).

Si bien existe una serie de aplicaciones con fines similares, son pocas las que cumplen los requerimientos de los centros de salud nacionales, ya sea por que están diseñados para fines específicos o simplemente porque a nivel cultural son diferentes.

Aunque el concepto de Telemedicina que ha estado en boga desde hace ya varios años, y si bien es cierto, algunos países desarrollados ya disponen de herramientas software similares a la propuesta, se puede concluir que aún no se ha llegado a una solución definitiva que pueda ser adoptada por cualquier entidad hospitalaria en el ámbito nacional.

La factibilidad de llevar a cabo el proyecto de desarrollo de una aplicación de telemedicina esta dada por el Esfuerzo y los requerimientos tecnológicos, tales como calidad de servicio en transmisión de datos, resolución en los equipos de captura de video, audio etc. En cuanto al Esfuerzo que demanda la aplicación se utilizó la metodología de Punto de Función para determinar la métrica del total de línea de código. A partir de ahí se hicieron las estimaciones con COCOMO para determinar el esfuerzo (ver anexo 7.1).

3. Proceso de Desarrollo

En cuanto al proceso de desarrollo, lo primero que se decidió fue el Ciclo de Vida a seguir. Para esta etapa se optó por el método que mejor se ajustaba a la naturaleza del Sistema y a la forma de trabajo del equipo.

3.1. Ciclo de vida

El Ciclo de Vida de un producto corresponde a las etapas por las cuales pasa hasta alcanzar su desarrollo final, y está dada por el tipo de problema abordado, la cultura de la organización y la experiencia de los desarrolladores. En este caso en particular, se utilizaron prototipos, debido a que en un principio los usuarios no tenían claros los requerimientos. De los cuatro enfoques básicos, se optó por el “Prototipo parchado” [KEN97], debido a que la recolección de información inicial en la etapa de análisis de requerimientos fue minuciosa. Las modificaciones posteriores se basaron principalmente en cambios de las interfaces, formas de despliegue, ordenamiento, optimización de funciones e implementación de módulos anexos (noticias, foros, etc.).

3.1.1. Prototipo parchado

Se conoce como “Prototipo parchado” a un sistema que tiene las características necesarias, pero que es ineficiente. Los usuarios pueden interactuar con él, acostumbrándose a la interfaz y a las entradas-salidas disponibles, sin embargo, la recuperación de información no es la óptima debido a que la programación de los diferentes módulos que lo componen no fue rigurosa, ya que el objetivo de este modelo es primar la funcionalidad por sobre la eficiencia [KEN97]. El prototipo partió de un modelo básico que se mejoró paulatinamente hasta lograr el producto final que se traduce en el sistema Telemedicina.

3.2. Especificación de Requisitos

3.2.1. Descripción problema y situación actual

Para realizar el análisis de requerimientos se recopiló información de acuerdo a metodologías indicadas en [ING01], destacando lo siguiente:

- Entrevistas en terreno a personal médico involucrado.
- Estudio del funcionamiento interno del establecimiento, deduciendo modelos de datos que abstraen la interacción entre usuarios y la información manipulada.
- Revisión de documentos médicos.
- Conocimiento empírico, adquirido durante los años de estudio, en el diseño de sistemas e implementación de aplicaciones.

La información recopilada incluyó exámenes de Obstetricia, observaciones realizadas por los profesionales del área, requerimientos y funcionalidades del Sistema.

3.2.2. Área de obstetricia

Al analizar el sistema de fichas clínicas de obstetricia, se observó una serie de problemas asociados a la falta de automatización de los procesos, identificándose un conjunto de datos por cada paciente: nombre, rut, código de paciente, grupo sanguíneo, etc, los cuales fueron empleados para definir las características del sistema de telemedicina genérico y se diferencian entre especialidades principalmente en el contenido de sus exámenes.

Dentro del sistema tradicional se utilizan fichas clínicas en papel, lo que implica un deterioro progresivo de la información a través del tiempo y una replicación innecesaria de datos tanto del paciente como del historial clínico, lo cual se incrementa significativamente en caso de embarazos múltiples. Esta consta de 2 partes:

- Primer trimestre.
- Segundo / tercer trimestre.

La ficha contiene para cada etapa del embarazo la siguiente información:

- Datos personales del paciente.
- Datos del médico solicitante.
- Descripción de consultas anteriores.

El examen del primer trimestre proporciona:

- Información del útero.
- Anexo derecho.
- Anexo izquierdo
- Douglas
- Conclusiones

Para el segundo y tercer trimestre considera:

- Información del feto.
- Placenta.
- Líquido amniótico.
- Observaciones.
- Conclusiones.

Junto con la información antes descrita, se realizaron visitas al hospital Base de Valdivia y Centros de Salud alejados del núcleo provincial, para identificar problemas y falencias que ayudaron a definir los requerimientos locales. Los problemas se pueden resumir en:

- Falta de automatización: el personal administrativo debe ingresar manualmente los datos de cada consulta al sistema en uso, lo que provoca lentitud y carga de trabajo variable, debido a que este se encuentra aislado y sin posibilidad de ser consultado en línea desde otros computadores de la red interna del hospital, por lo que para acceder a él es necesario recurrir al operador y tener conocimientos específicos sobre la aplicación.

- Falta de coordinación: inherentes al sistema “Lápiz y Papel”, entre: proveedor y administrador del sistema, solicitante de información y operador manual del sistema, oficina de recepción de pacientes y administrativos de obstetricia y finalmente la más importante falencia: descoordinación y desconexión total entre la base de datos general del Hospital Regional de Valdivia y el sistema utilizado actualmente para el área de obstetricia.
- Falta de comunicación: los problemas de comunicación con las demás áreas dificultan la realización de interconsultas, derivación de pacientes a otras especialidades, intercambio de fichas clínicas entre establecimientos hospitalarios, actualización y revisión de datos personales del paciente y los funcionarios de los centros de salud.
- Falta de actualización: el actual sistema de obstetricia se usa desde el año 1993, por consiguiente, es poco moderno, de ambiente gráfico-operacional hostil para los usuarios e incompatible con los sistemas implementados dentro del establecimiento en años posteriores, tales como el registro de pacientes o las imágenes digitales generadas por los ecógrafos. Otra problemática asociada a esta obsolescencia es la imposibilidad de analizar los datos desde plataformas externas a las que manejan información en forma nativa, incluso acceder a ésta remotamente ya que el equipo en el cual se encuentra es antiguo y sin actualizar. Además, el programa se ejecuta en el sistema operativo MS-DOS, no empleando una base de datos relacional ni mecanismos que resguarden la información almacenada a nivel del software o hardware.

Toda esta información permitió definir los requisitos funcionales y no funcionales del sistema.

3.2.3. Requisitos funcionales

- Ficha digital para obstetricia enmarcada en el episodio “embarazo”, con exámenes correspondientes al primero, segundo y tercer trimestre de gestación. Requerimiento que se puede generalizar, interpretándolo como la capacidad de almacenar exámenes, episodios o fichas clínicas para cada paciente, similar al historial físico que almacena cada establecimiento en sus dependencias.
- Página de noticias, que proporcione información de hechos importantes en forma inmediata y masiva, relacionada con el quehacer del sistema público de salud.
- Despliegue y almacenamiento de publicaciones y documentos.
- Herramientas que permitan la comunicación entre usuarios.
- Seguridad de la información de cada paciente (restricción de acceso).

3.2.4. Requisitos no funcionales

- Interconectar la base de datos del hospital con la del sistema Telemedicina, para evitar la duplicidad de información.
- Implementar algoritmo para relacionar dependencia entre los exámenes, episodios y especialidades, considerando niveles de jerarquía.
- Capacidad de modificar las fichas creadas, dando lugar a actualizaciones que permitan mejorar el desempeño, responder a nuevos requerimientos, o agregar otros datos para almacenar.
- Sistema de registro que permita identificar automáticamente la especialidad a la que pertenece el usuario, de acuerdo a su identificador, permaneciendo vigente todo el tiempo que se mantenga la conexión al servidor.
- Obtención automática de fecha, hora y usuario responsable de la creación o modificación de fichas, episodios y exámenes; historial de cambios.
- Despliegue y distribución de información en pantalla de manera ordenada y de fácil manejo.

- Ordenamiento de búsquedas, en base a criterios personalizados por los usuarios, por ejemplo, fecha o código.
- Ordenamiento personalizado de componentes y secciones que conforman los exámenes.
- Ajuste automático de contenidos en base a la resolución de la pantalla.
- Almacenamiento de imágenes ecográficas asociadas a los exámenes, en tamaño y resolución original.
- Creación de plantillas para secciones de exámenes que pueden ser comunes a una o más especialidades.
- Creación de perfiles para discriminar niveles de seguridad en el acceso a los diferentes recursos del sistema.
- Módulo de administración que permita activar o desactivar cuentas de usuario.
- Indentación automática de contenidos de exámenes en base a secciones y componentes, para mejorar orden y legibilidad.

3.2.5. Requerimientos de hardware y software

Para desarrollar el sistema se optó por el ambiente Web, principalmente por la portabilidad de la plataforma, es decir, su fácil implementación en diversos sistemas operativos usados en la actualidad. Lo anterior permitiría que a mediano plazo, este pueda ser incorporado en cualquier lugar que cuente con una conexión a Internet (cobertura que aumenta exponencialmente debido a las tecnologías inalámbricas).

3.2.5.1. Análisis de herramientas y plataformas

Se adoptó el lenguaje PHP para la programación, ya que al ser Open Source y tener una estructura similar a C++ facilita la investigación y solución de problemas.

Para elaborar los primeros prototipos se utilizó base de datos Mysql junto con la herramienta PhpMyadmin, por su fácil configuración y reducido tamaño. Posteriormente se migró a Oracle, al igual que el Hospital Regional de Valdivia (dueño de la licencia),

lográndose un entorno potente, con información actualizada y de uso común para el personal hospitalario. A continuación se listan las características de las herramientas utilizadas:

EasyPHP

“EasyPHP permite usar toda la potencia y flexibilidad que ofrece el lenguaje PHP y un eficiente uso de bases de datos. Incluye Apache, MySQL y ejecución de código PHP, así como herramientas de desarrollo para sitios Web” [REF, 03].

Se puede concluir que EasyPHP es una excelente herramienta para programar en lenguaje PHP bajo un entorno Windows; instala, configura y ejecuta automáticamente las aplicaciones necesarias, evitando la complejidad al hacerlo en forma individual, lo que ahorra tiempo en esta tarea.

Apache

“El servidor HTTP Apache es un software de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parchado").” [REF, 04].

Ventajas:

- Utilizado a nivel mundial por prestigiosas empresas.
- Funciona sobre sistemas y arquitecturas variadas, ofreciendo capacidad de rendimiento superior a muchos de sus competidores.
- Es un software libre y su código fuente está disponible para la comunidad que lo va mejorando constantemente.

Desventajas:

- El proceso de instalación 'nativo' no es sencillo, debido a que es necesario descargar el código fuente de la Web, compilarlo y seguir una serie de pasos a nivel de usuario avanzado. Sin embargo, todas las distribuciones de Linux y otros sistemas UNIX lo incorporan por defecto.
- La configuración no es trivial. Se requiere conocimientos específicos para manipular su compartimiento si se desea explotar todas sus funcionalidades.

PHP

“PHP (Hypertext Pre-processor) es un lenguaje de programación usado para la creación de contenido para sitios Web. Inicialmente conocido como PHP Tools o Personal Home Page Tools. Se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web [REF, 01]. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+.” [REF, 06].

Ventajas:

- Muy sencillo de aprender.
- Similar en sintaxis a C y PERL
- Soporta orientación a objeto, clases y herencia.
- El análisis léxico para recoger las variables lo hace de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Buen soporte de acceso a diferentes base de datos.
- Comprobación de parámetros en el servidor y no en el cliente (como se hace con javascript) de forma que se evitan los chequeos locales, previniendo la inserción de órdenes maliciosas dentro de una consulta de datos solicitud al servidor.

Desventajas:

- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La orientación a objetos es aún deficiente para aplicaciones de gran envergadura.

MySQL

Aunque carece de elementos propios de las bases de datos relacionales, es de amplio uso por los desarrolladores Web con contenidos dinámicos por su simplicidad, aquellos elementos que faltaban en su concepción inicial han sido incorporados paulatinamente, tanto por el staff de desarrollo interno como por desarrolladores de software libre. La licencia de MySQL es GPL.

Oracle

Según lo expuesto en [REF07] "Desde 1977, Oracle ha provisto tecnología, aplicaciones, y servicios que entregue a las compañías la información más actualizada y más exacta posible". Versátil, multiplataforma, robusta y poderosa, son algunas de las características de Oracle. Es considerado por muchos como la base de datos más potente que existe, por ello, es utilizado por las empresas e instituciones más colosales a nivel mundial. El valor de adquisición de las licencias para su utilización comercial no es una inversión menor (inalcanzable a veces para pequeñas y medianas empresas).

Oracle al ser una base de datos de gran envergadura y de uso a gran escala, posee un staff de personas e instituciones que se dedican a probarlo y aplicarlo a distintos campos científicos, dichos grupos de trabajo constantemente elaboran publicaciones relacionadas con su uso y la solución de problemas, motivo por el cual es sencillo conseguir material de alto nivel relacionado con la plataforma en cuestión.

Entre las desventajas principales están: elevado costo de licencias, requiere un servidor potente y dedicado para optimizar su desempeño, sus interfaces de administración por defecto son poco amistosas y poco intuitivas.

Dreamweaver

“Dreamweaver es un editor WYSIWYG (acrónimo de What You See Is What You Get) de páginas web, creado por Macromedia (actualmente Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación Web, por sus funcionalidades, integración con otras herramientas como Macromedia Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Frontpage. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas.

Ventajas:

- Facilidad de uso (intuitivo)
- Integración nativa con otras tecnologías para el diseño de aplicaciones Web.
- WYSIWYG (Lo que se ve, es lo que se obtiene como resultado)

Desventajas:

- Elevado costo.
- Elevado consumo de recursos de máquina.

3.2.5.2. Requerimientos de Software

Debido a que el sistema considera en su gran mayoría el almacenamiento y acceso a la información, se determinó que la mayor carga de trabajo recae en la base de datos, por lo que se considera a esta como una componente crítica debido al volumen de datos que se contempla (27 centros de salud y 61 postas en la Provincia de Valdivia), por lo que los requerimientos en este aspecto están supeditados al motor de base de datos utilizado. Específicamente Oracle 8i tiene los siguientes requerimientos:

Requerimientos del servidor

- Sistema Operativo Linux/Unix o Windows. Se probó Debian Knoppix 3.3 (kernel 2.4.22), Redhat 3AS (kernel 2.4.21-37.EL). Además, para el desarrollo se trabajó sobre Windows 2k Server y Windows XP Profesional.

Para el servidor Linux con motor Oracle 9.2.0.4 fue necesaria la instalación del parche p3006854_9204_LINUX, que corrige problemas con el instalador (OracleInstaller).

- Servidor Web Apache. El Sistema fue desarrollado en Apache v.1.3.33 sobre Windows 2k Server e instalado para pruebas en Linux con Apache v2.0.

Para el desarrollo fue necesaria la utilización de una serie de herramientas, consiguiéndose excelentes resultados en la siguiente configuración:

- Lenguaje PHP. El Sistema fue desarrollado en PHP v4.3.10 sobre Windows 2k Server e instalado para pruebas en Linux con PHP v4.3.2 y v5.1.2.
- Módulo oci8. Para el desarrollo y pruebas se utilizó oci8 v1.2.2 como módulo de PHP.
- Para el servidor Linux se requirió instalar Oracle Instant Client. El desarrollo se realizó sobre la versión instantclient-sdk-linux32-10.2.0.2-20060331.
- Para el servidor Windows 2k Server fue necesario instalar el cliente de Oracle9i.

Requerimientos del cliente

Se entiende como cliente, al computador remoto, que accede a los contenidos provistos por el sistema. Los requerimientos mínimos se listan a continuación:

- Sistema Operativo Mac OS, Linux/Unix o Windows.
- Navegador Web gráfico (Browser).
- Compatibilidad para ejecutar Javascript.

3.2.5.3. Requerimientos de Hardware

Requerimientos del servidor (según requerimientos mínimos de hardware [REF07])

- 1 CPU de 1Ghz, recomendable ambiente multiprocesador. Dependiendo del Sistema Operativo puede variar la arquitectura; se probó en x86, pero eventualmente podría instalarse en SPARC, PowerPC, o cualquier otra que soporte motor Oracle.
- 256MB de RAM en ambiente de pruebas, recomendable de 1 a 2GB para producción, dependiendo del crecimiento proyectado, normalmente de acuerdo a los requerimientos mínimos de instalación de los motores Oracle [REF07].
- 10GB de espacio en disco para motor Oracle y base de datos, viéndose incrementado dependiendo de la estimación de crecimiento proyectada por el establecimiento [REF07]. El espacio considerado es adicional al Sistema que ya está en funcionamiento en el Hospital Regional de Valdivia, cuyos requerimientos son independientes de esta estimación.
- Como se trata de un Sistema productivo crítico en su área, es recomendable:
 - o Utilización de controladora RAID para el resguardo de la información ante una falla de disco, capacidad que no dispone el actual sistema.
 - o Utilización de fuentes de poder redundantes, para tolerancia a fallos y disminución de los tiempos de recuperación del sistema hardware.
 - o Respaldo de energía. Utilización de UPS para el resguardo de la base de datos ante un corte abrupto, ya que este puede corromper los datos.
 - o Condiciones ambientales adecuadas para la máquina, requerimientos de ventilación, temperatura y humedad sujetos a recomendaciones del fabricante de la máquina.
- Tarjeta de red de 100Mbps Full Duplex, recomendable de 1000Mbps.

- Conexión a la red interna del Hospital y enlace seguro hacia Internet.
Recomendable la utilización de un corta fuegos, permitiendo solamente solicitudes por el puerto 80.

Requerimientos del cliente

Se requiere de una máquina con capacidad de navegar en Internet, mínimo:

- 1 CPU de 200Mhz
- 64MB de RAM
- Disco de 1GB, solo para Sistema Operativo y caché de páginas Web.
- Tarjeta de red de 100Mbs Full Duplex.

Conexión a red interna del Hospital o a Internet si se conecta desde otro lugar.

3.3. Análisis

3.3.1. Sistema general

Basándose en los requerimientos generales determinados en la especificación de requisitos, se deduce un diseño lógico propuesto en la figura 3.1 como un diagrama de contexto básico:

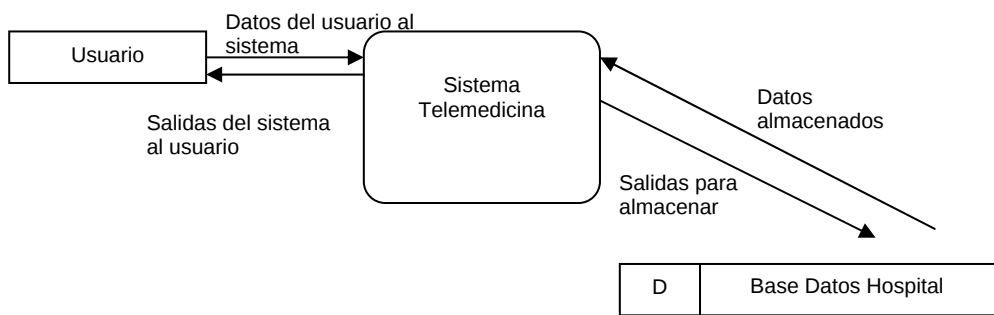


Figura 3.1: Diagrama de contexto del sistema general.

La base de datos consideró información de las tablas actuales del hospital y la información del sistema telemedicina. En este último se considera los datos de la ficha clínica de pacientes y de los módulos de comunicación, almacenamiento y despliegue de información tales como: foro, chat, video, publicaciones de documentos y noticias;

finalmente, a modo de resguardar la confidencialidad propia de los centros de salud, se consideró un módulo de validación de usuarios, con distintos niveles de privilegios en base a perfiles, lo que se tradujo posteriormente en un módulo de administración general, capaz de configurar cuentas de usuarios, crear o modificar fichas clínicas y exámenes.

3.3.1.1. Diagrama Entidad Relación

Los modelos entidad-relación (o diagramas de Chen), han sido desarrollados para describir conceptos, definiéndose una entidad como cualquier “objeto” sobre el que se tiene información, en otras palabras, es el paso necesario para abstraerse al mundo real. Otra de sus características importantes es la posibilidad de identificar un gran número de instancias para las entidades (correspondientes a los tipos de usuarios), que aseguran la completitud del programa y de la base de datos relacional resultante del análisis [REF,05].

El diagrama de entidad-relación de la figura 3.2, fue utilizado para modelar de mejor manera las formas relevantes dentro del sistema, sirvió como guía inicial para establecer un modelo de datos coherente y como base para la creación de los flujos de datos entre los módulos programados al indicar las relaciones entre ellos.

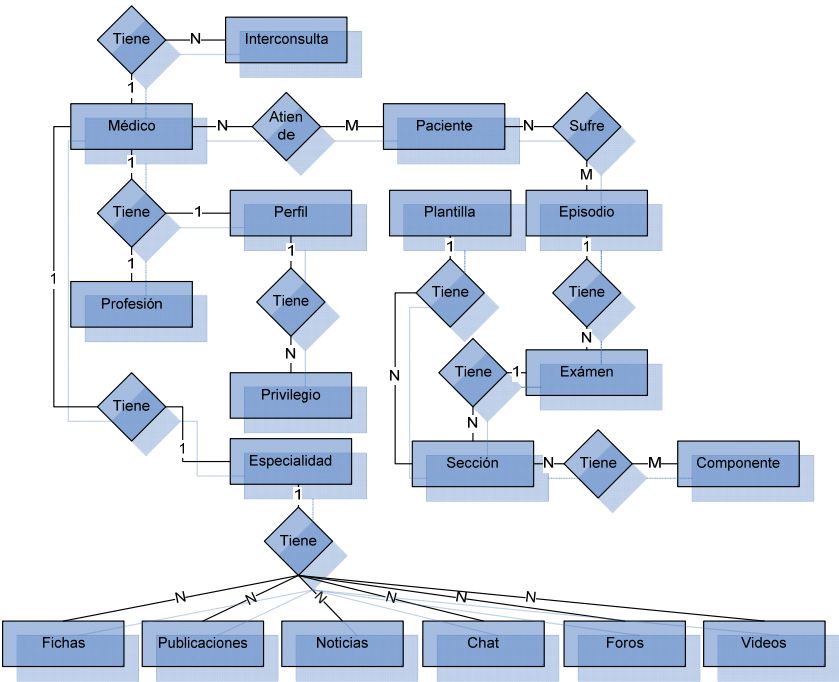


Figura 3.2: Diagrama entidad-relación del sistema general.

Explicación:

- Un médico atiende a varios pacientes, realiza interconsultas médicas, tiene un perfil de usuario, además de tener una profesión y una especialidad.
- Cada perfil posee privilegios.
- Cada especialidad tiene videos, chat, noticias, foros y publicaciones.
- Los pacientes son atendidos por más de un médico y sufren más de un episodio.
- Los episodios tienen exámenes.
- Los exámenes poseen secciones.
- Las secciones incluyen otras secciones o componentes y se configuran en base a una plantilla.
- Un componente pertenece a una o mas secciones.

3.3.1.2 Modelo de datos para Sistema Genérico

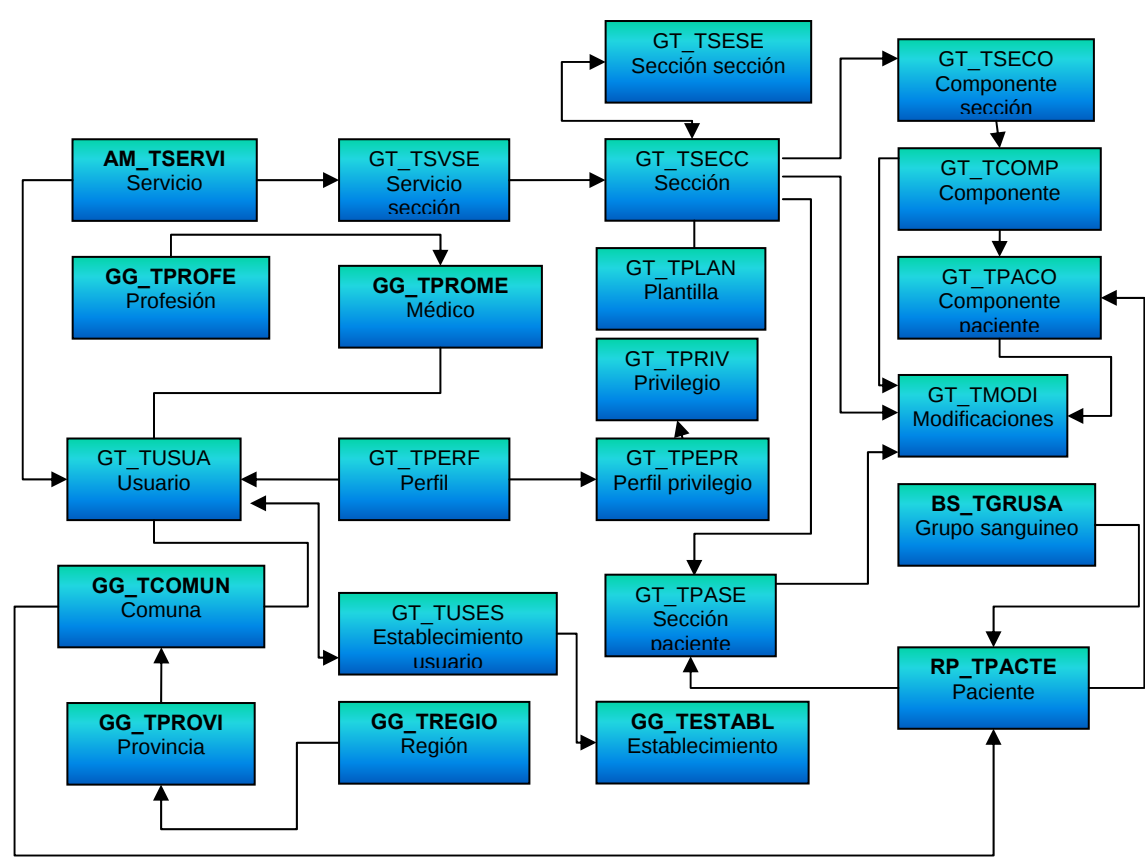


Figura 3.3: Modelo de datos del sistema genérico.

La descripción de las tablas y sus campos será detallada en el anexo 7.2, correspondiente a “Descripción de las tablas y sus campos”.

Estructura y relaciones entre tablas

En la figura 3.3, una especialidad (AM_TSERVI) tiene asociada una o más secciones padre (GT_TSECC) que pueden ser del tipo indicado por la plantilla (GT_TPLAN). Para nuestro caso, las secciones padre serían los episodios.

Una sección padre por defecto es de tipo "S. secc" (Sección formada por otras secciones: solo puede contener secciones), donde cada una de estas es un examen.

Para poder asociar secciones padre a una especialidad existe la tabla GT_TSVSE, que contiene varios códigos de secciones padre para cada especialidad (episodios). Así mismo, por cada sección padre o "S. secc" existe una o más secciones hija que se relacionan en la tabla GT_TSESE. Cada componente (GT_TCOMP) se asocia con una o más secciones en la tabla GT_SECO.

Tablas de datos fichas clínicas: Cuando un médico agrega episodios y exámenes en la ficha de un paciente, éstos son almacenados en la tabla GT_TPASE.

Por otro lado, los componentes (datos) se almacenan en la tabla GT_TPACO.

Tablas comunes fichas clínicas: Todas las modificaciones de estructura y datos (secciones y componentes), incluida la creación, se almacenan en la tabla GT_TMODI.

Tablas de acceso: Cuando un usuario ingresa al sistema, se valida con su dirección de correo electrónico (usuario) y su contraseña. Ambos almacenados en la tabla GT_TUSUA. Si su cuenta está activa se consulta su perfil (tabla GT_TPERF) y se despliega la ventana principal de la especialidad a la que pertenece (noticias).

El perfil define si se trata de un médico, alumno, administrador, etc., donde cada uno tiene asociada una lista de privilegios GT_TPEP que indican a qué nivel tiene acceso (GT_TPRIV).

3.3.1.3. Diagramas de Flujo de Datos

Para modelar el sistema fue necesario describir el flujo de la información; haciendo un análisis progresivo, se ha obtenido un diagrama de flujo de datos general. Se comenzará por el diagrama nivel 0, figura 3.4.

Diagrama nivel 0

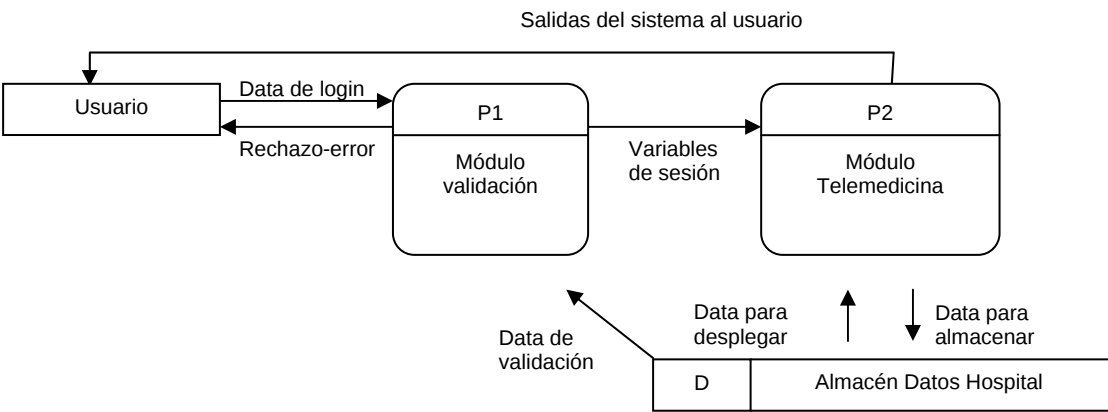


Figura 3.4: Diagrama nivel 0.

El usuario ingresa sus datos, los que son verificados en el módulo de validación (usuario y contraseña), si son correctos, se generan las variables de sesión, en las que se destaca la “especialidad” a la que pertenece e identificación de los privilegios que posee, para dar paso al módulo telemedicina. Dicho bloque corresponde al ambiente de trabajo del usuario, en el se generan: fichas clínicas, publicaciones, noticias y conversaciones.

Diagramas de nivel 1

Módulo de validación

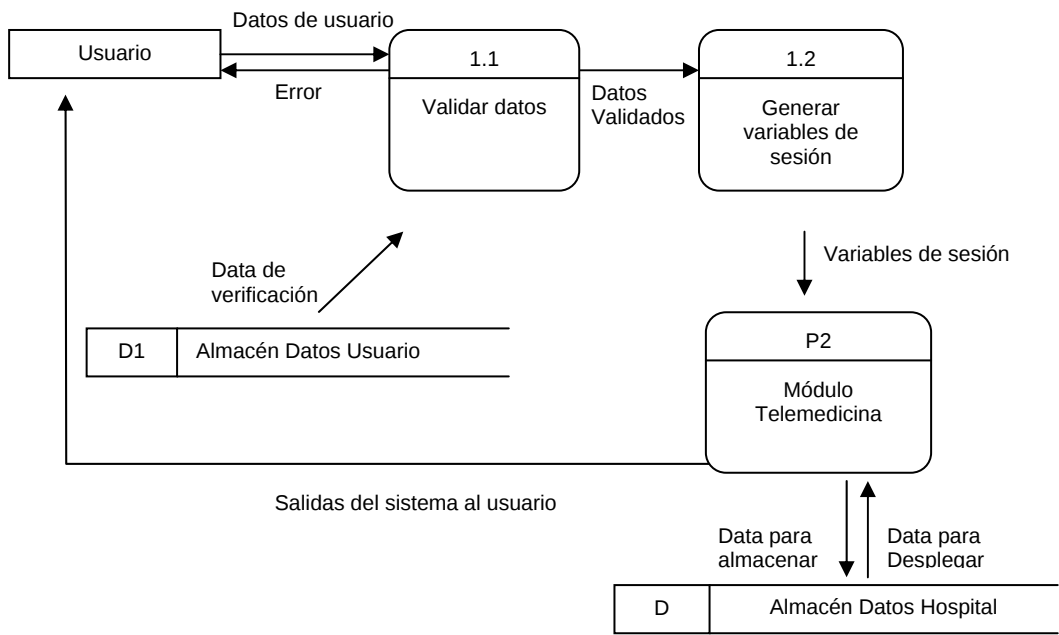


Figura 3.5: Diagrama nivel 1, módulo validación.

En la figura 3.5, el usuario ingresa sus datos, estos son validados respecto de la data existente en la tabla de usuarios, si existen, son traspasados al proceso 1.2 para generar las variables de sesión (en definitiva determinar la especialidad del usuario), dichas variables de sesión son enviadas al módulo Telemedicina. Las contraseñas son encriptadas bajo el algoritmo md5.

Módulo Telemedicina

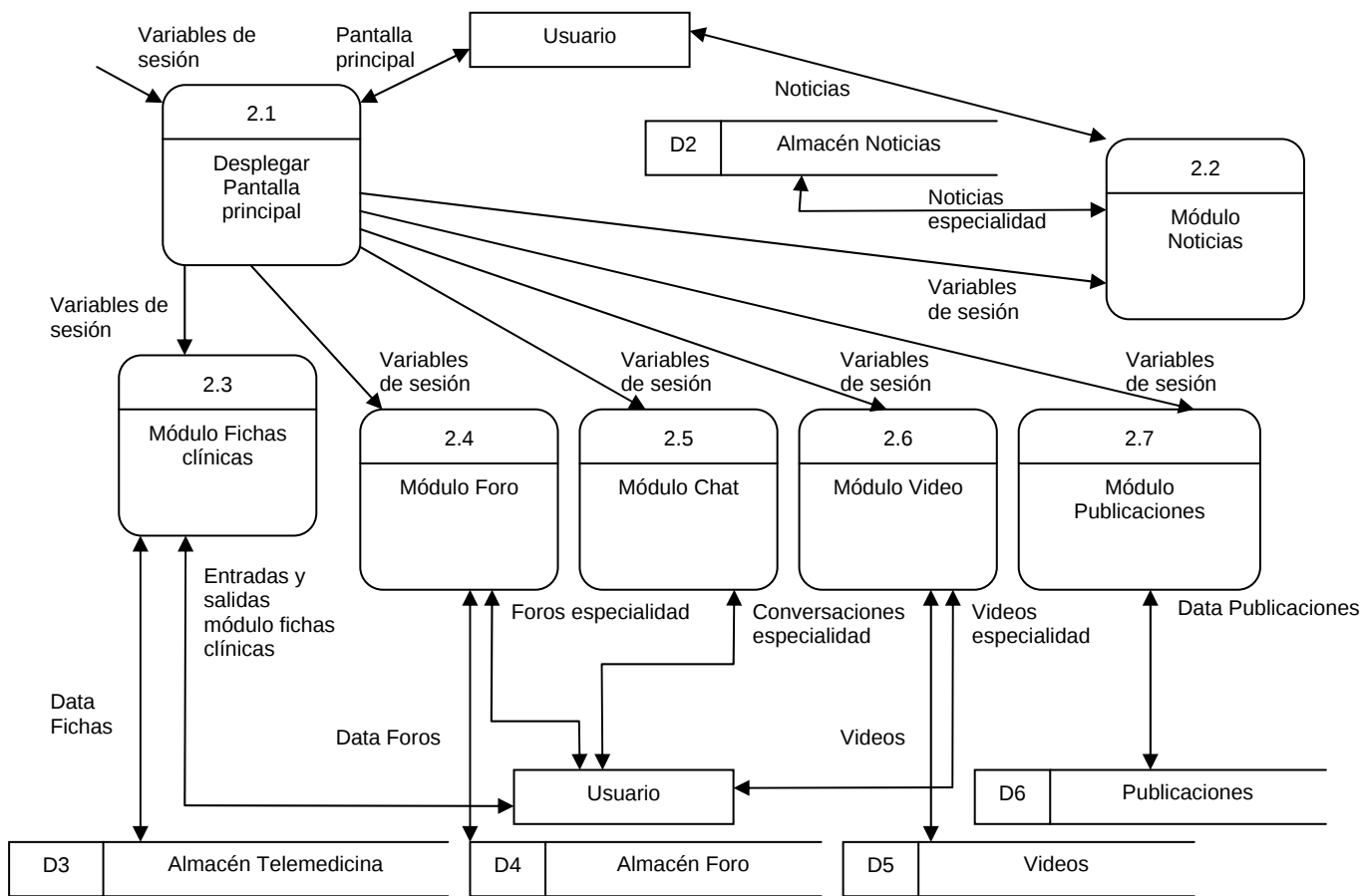


Figura 3.6: Diagrama nivel 1, módulo telemedicina.

En la figura 3.6 se aprecia que al definirse las variables de sesión (principalmente su especialidad), se despliega automáticamente el módulo de noticias (discriminando por especialidad). La cabecera de las páginas Web permite dirigirse a los módulos restantes y operar en ellos según el perfil del usuario activo, así se interactúa con las distintas tablas, con fines de almacenamiento y lectura de los datos manipulados.

El módulo de Chat, foro, etc. no se explican en profundidad, debido que existe abundante documentación respecto al flujo de información involucrada. Encontrándose incluso, sistemas bajo licencia GPL que pueden ser integrados en cualquier aplicación Web.

Diagramas de nivel 2

Módulo Noticias

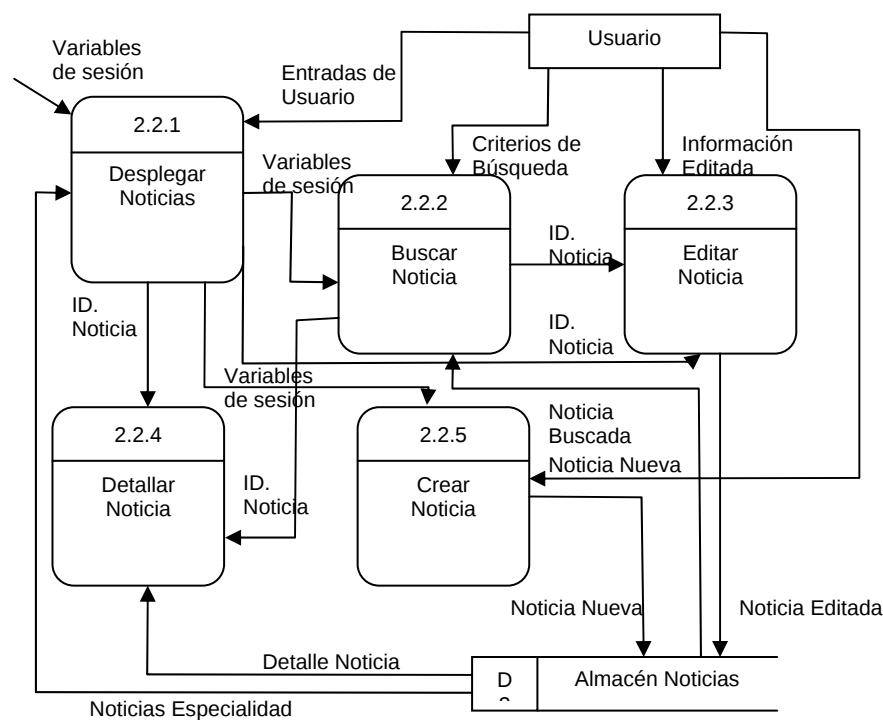


Figura 3.7: Diagrama nivel 2, módulo noticias.

El proceso Desplegar Noticias, según figura 3.7, será el primero que se desarrollará en el módulo noticias, éste tomará como discriminantes a las variables de sesión, desplegando solo lo que corresponda a la especialidad del usuario o las noticias definidas por sus autores como generales. El usuario podrá desde allí crear, ver, buscar, editar, ordenar noticias creadas (dependiendo de los permisos que se le otorguen al inicializar la sesión). Al crear nuevas noticias o editar las existentes, se permite subir una imagen que acompañará la nota, en caso de no subir una, el sistema utilizará una imagen por defecto.

En general, cada modificación a la base de datos queda registrada, almacenándose el identificador del usuario que realizó los cambios, al igual que la fecha y hora, lo que ocurre en cada módulo programado.

El usuario toma parte de los procesos, ingresando la información que le solicite el sistema, tal como: criterios de búsqueda, modificaciones, nuevas noticias, etc., el sistema le responde desplegando en pantalla sus solicitudes.

Módulo Foro

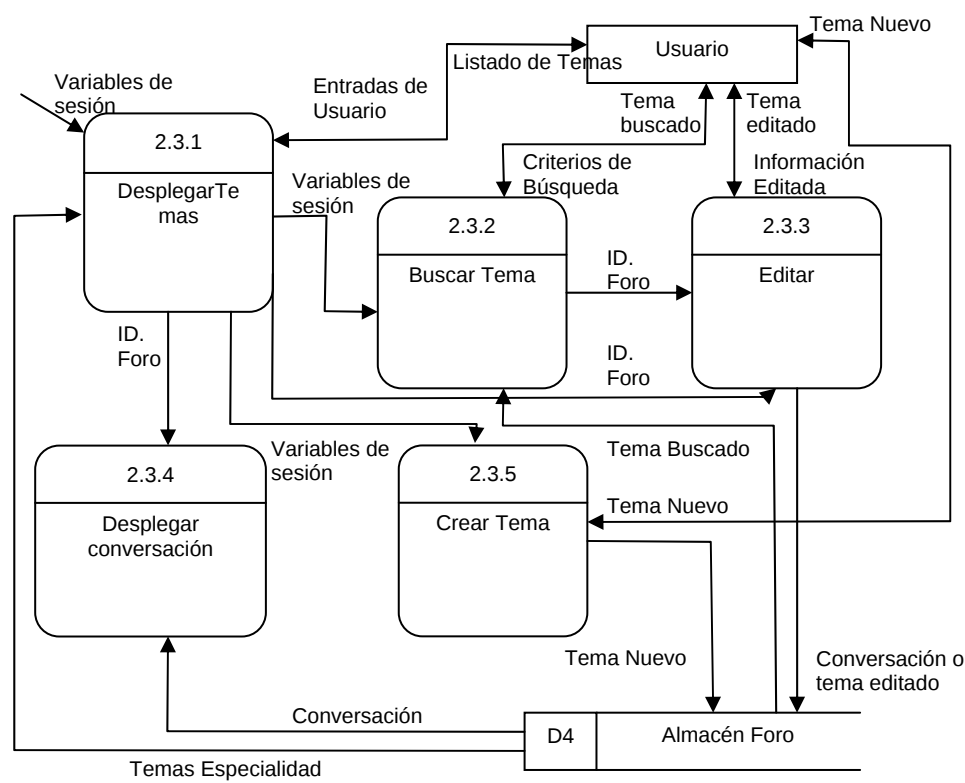


Figura 3.8: Diagrama nivel 2, módulo foro.

Al igual que en el módulo anterior, la figura 3.8 muestra que lo primero en desplegarse son los temas que correspondan a la especialidad del usuario, dicho usuario podrá buscar temas, editarlos, desplegar conversaciones (foros) y crear nuevos temas (foros), en este módulo no se manipulan documentos de ningún tipo.

Módulo Publicaciones

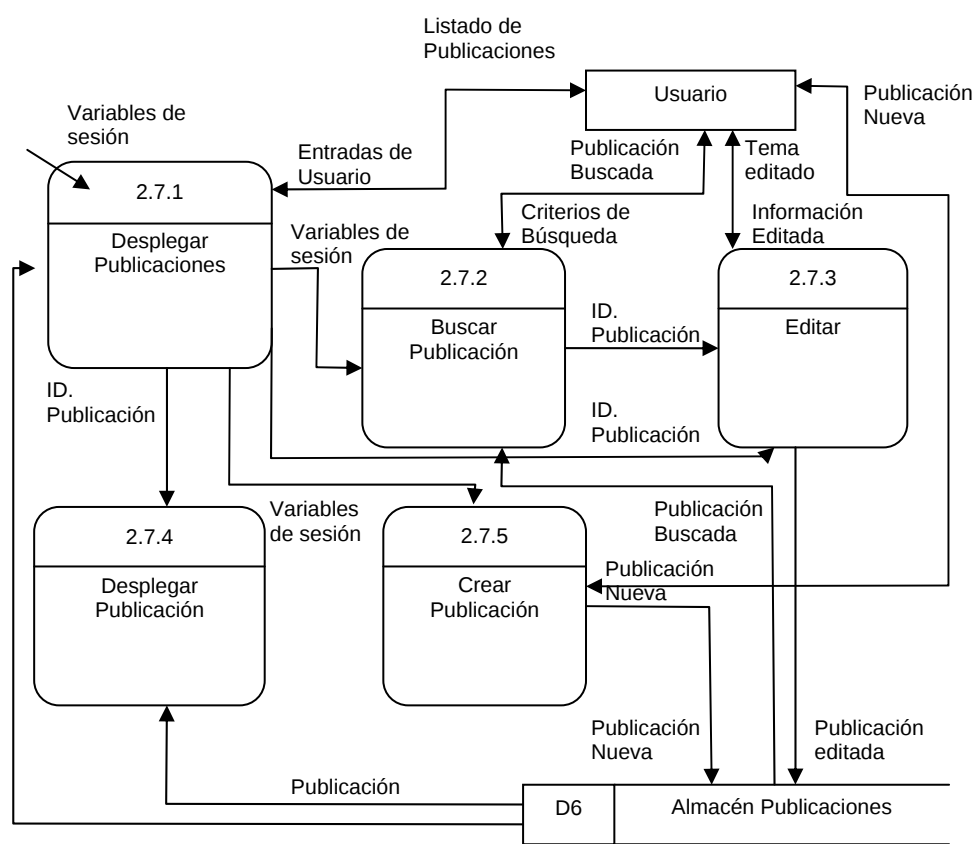


Figura 3.9: Diagrama nivel 2, módulo publicaciones.

En la figura 3.9, la lógica es similar a los dos módulos anteriores, con una única diferencia: incorporar la posibilidad de agregar archivos (que son las publicaciones), en formatos de texto, planilla electrónica o imagen, pudiendo ser descargados desde el servidor.

Módulo Fichas clínicas

Es el módulo principal del sistema implementado, corresponde al desarrollo central de la tesis; justificado en capítulos anteriores, cuyo flujo de datos, figura 3.10, es la siguiente:

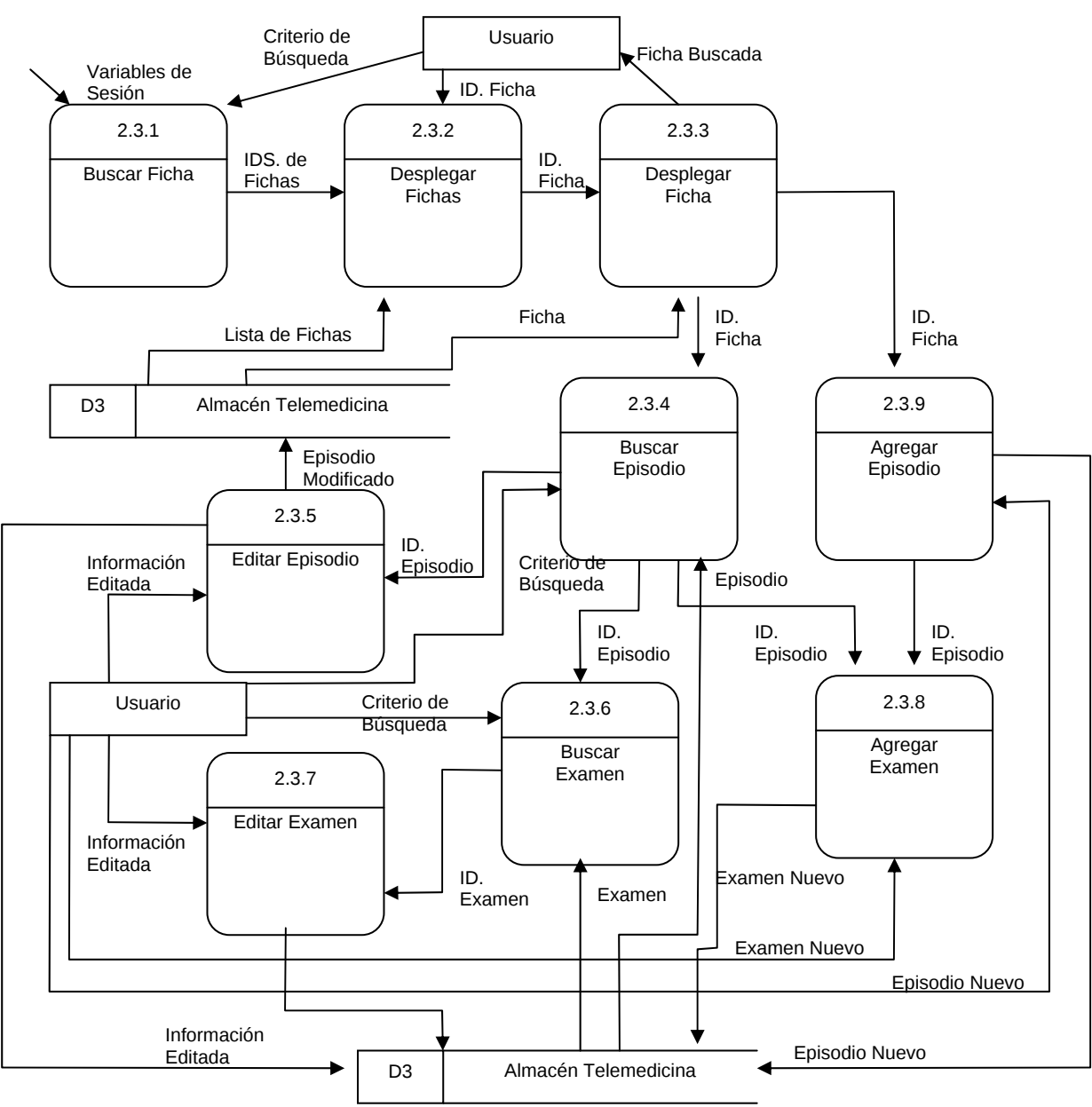


Figura 3.10: Diagrama nivel 2, módulo fichas clínicas

El flujo de datos comienza con el proceso buscar fichas, mediante un criterio de búsqueda, determina una lista de coincidencias, el usuario escoge entre ellas para pasar al proceso desplegar ficha, que selecciona en la base de datos la ficha que corresponda al identificador señalado por el usuario; la ficha desplegada contiene solo a la información del paciente. Existen 2 posibles caminos a partir de aquí, el primero es buscar un episodio existente, el segundo es crear un nuevo episodio. Al buscar un episodio existente, el usuario podrá editarlo, agregar un nuevo examen o buscar uno ya existente, a su vez, el usuario podrá editar un examen existente. En caso de que se esté creando un nuevo episodio, el proceso siguiente será el de agregar un examen.

Cada episodio y examen que despliegue el sistema, estará basado en la plantilla original definida por el administrador del sistema, mediante el módulo de administración, el cual será analizado posteriormente. Los accesos a los distintos menús, la posibilidad de editar o cambiar datos existentes así como los accesos, serán determinados según la información contenida en las variables de sesión, configurada cada vez que el usuario ingresa al sistema y mantenida para cada módulo.

Módulo Administración

Nace en el análisis de requerimientos, la necesidad de administrar el sistema de manera sencilla, provocar cambios en la forma y tipo de exámenes o episodios; del mismo modo gestionar permisos, configuraciones o simplemente agregar nuevas aplicaciones. Se crea de este modo un nuevo módulo, al que tienen acceso solo los administradores del sistema de información hospitalario, por medio de una dirección Web al servidor y validado con un nombre de usuario y contraseña.

Este módulo se divide en 2 subsistemas: el módulo de Episodios-Exámenes y el módulo de Secciones y Componentes.

Es importante recordar que este módulo se encarga de administrar la estructura del sistema, no la información contenida en sus tablas.

Episodios y exámenes

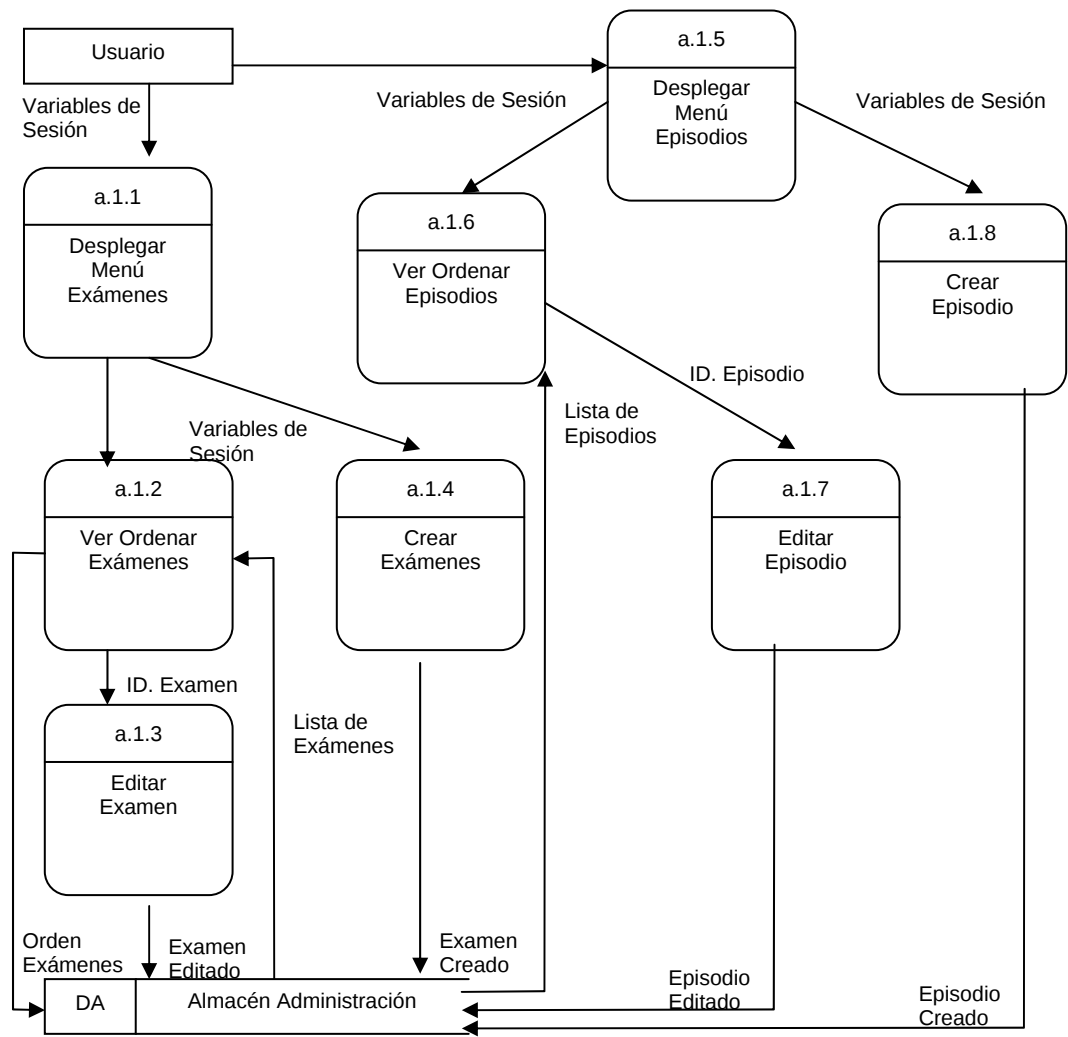


Figura 3.11: Diagrama nivel 2, módulo administración episodios y exámenes.

En la figura 3.11, el usuario se valida de igual manera que en el sistema de fichas, este proceso genera variables de sesión que se utilizan para determinar los privilegios del usuario. Una vez validado, puede seguir 2 caminos: exámenes y episodios.

En el proceso desplegar menú exámenes, puede optar por crear una nueva plantilla de exámenes o editar una plantilla existente, manipulándose entre sus procesos las variables de sesión y los identificadores de exámenes.

En el proceso desplegar menú episodios, podrá crear un nuevo episodio o modificar uno creado con anterioridad, al igual que en los exámenes, se emplean las variables de sesión y los identificadores de episodios.

Secciones y componentes

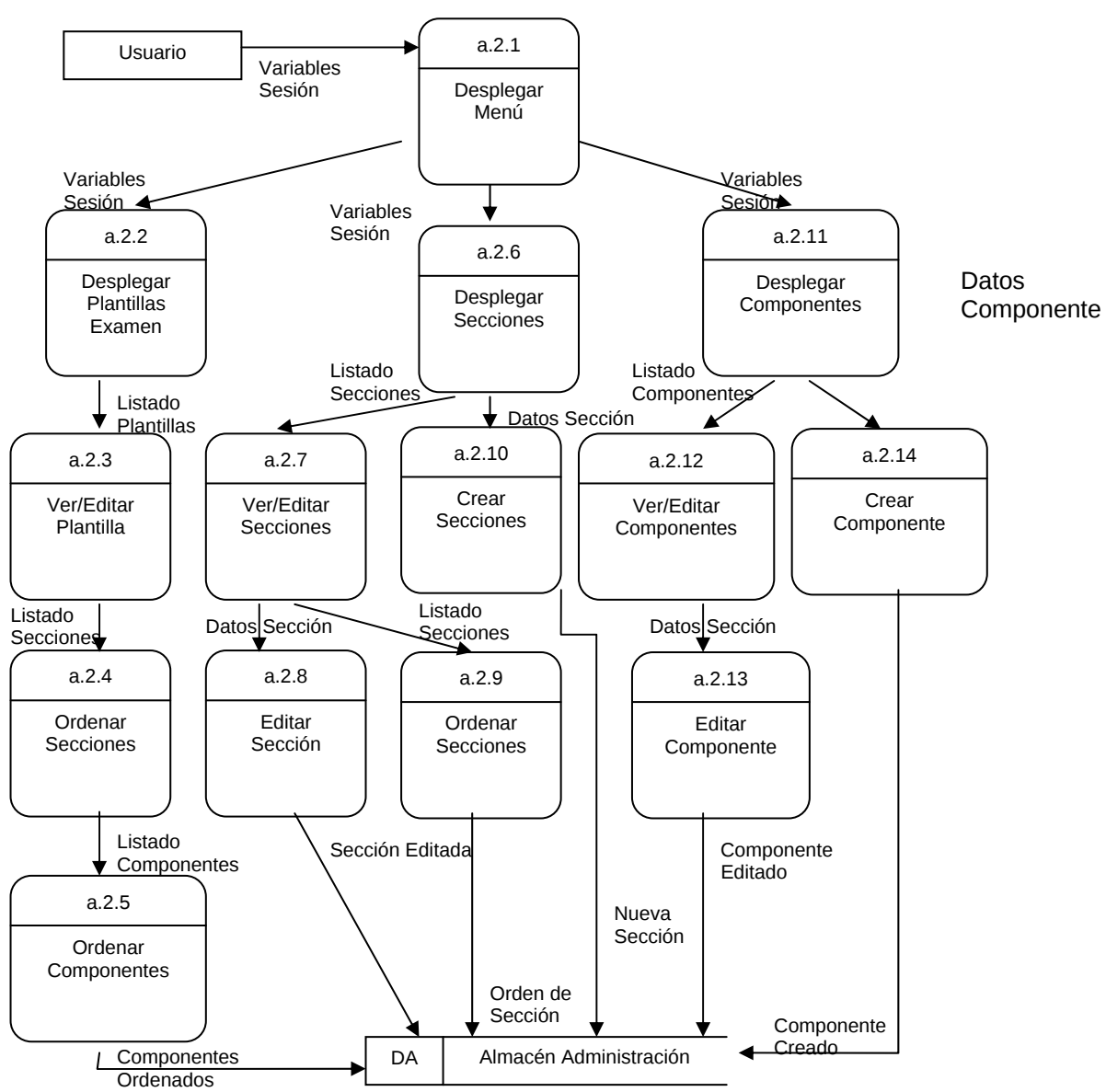


Figura 3.12: Diagrama nivel 2, módulo administración secciones y componentes.

En el módulo 3.12, lo primero que se despliega es el menú con las opciones, el usuario elije entre plantillas, secciones o componentes.

El propósito de este módulo es definir las estructuras básicas que conforman los exámenes, tales como: secciones y componentes, además (de acuerdo a la figura 2.16),

de establecer su orden al desplegarse, de modo tal, que no será necesario programar nuevos exámenes, puesto que basta con definir episodios junto a los exámenes que lo componen, a su vez, a los exámenes se le definen secciones que los forman y los componentes que incorpora cada sección, generándose una plantilla válida para ese examen en todo el sistema.

3.4. Diseño

Al terminar el análisis de la problemática e información, se diseña un sistema genérico soportado por una base de datos normalizada hasta su cuarta forma normal [REF,08], lográndose el esquema básico propuesto en la figura 3.13, que relaciona la información del hospital, con los datos que almacena el módulo de telemedicina.

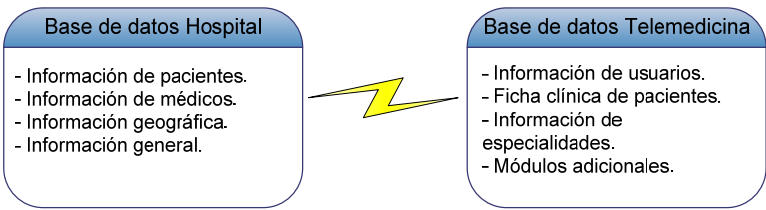


Figura 3.13: Esquema básico de relaciones entre bases de datos.

Posteriormente se contactó con un operador de la base de datos, generando una serie de modificaciones en la estructura interna de datos originalmente concebida, para hacerla compatible con lo que genéricamente usa el recinto hospitalario en estudio.

Con las revisiones realizadas sobre el prototipo inicialmente propuesto, por parte de los profesores encargados y tesisistas del área de obstetricia, se logra el diagrama 3.14 identificando las relaciones e información compartida entre la ficha digital para obstetricia, el sistema actual del hospital regional y la información del entorno en el cual se ejecuta la aplicación, por ejemplo: fecha, hora, resolución de pantalla, etc.

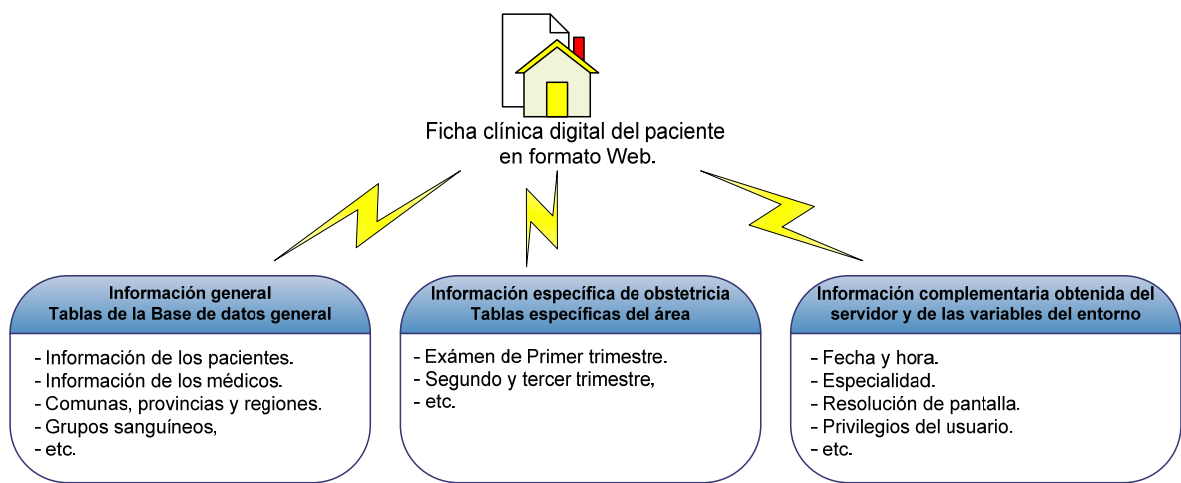


Figura 3.14: Diagrama de relaciones básicas e información compartida en el sistema.

3.4.1 Generalización

Determinada la estructura del sistema para obstetricia, se abstrae la información específica del área en estudio e implementa un modelo genérico, que puede ser utilizado para incorporar nuevas especialidades, junto a sus fichas clínicas específicas y funcionalidades anexas como: foro, chat, noticias, publicaciones y herramientas administrativas o de mantenimiento. El modelo general de resume en la figura 3.15:

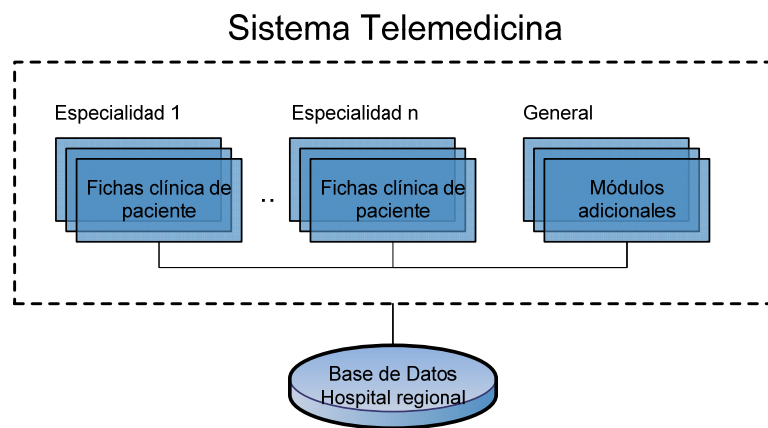


Figura 3.15: Modelo general del sistema telemedicina.

En la figura 3.15, se indican las interacciones entre distintas especialidades, módulos generales (noticias, foro, chat, publicaciones, etc) e información contenida en la base de datos del hospital, la cual es accedida con permiso de “solo lectura”. El modelo en detalle tiene la estructura planteada en la figura 3.16.

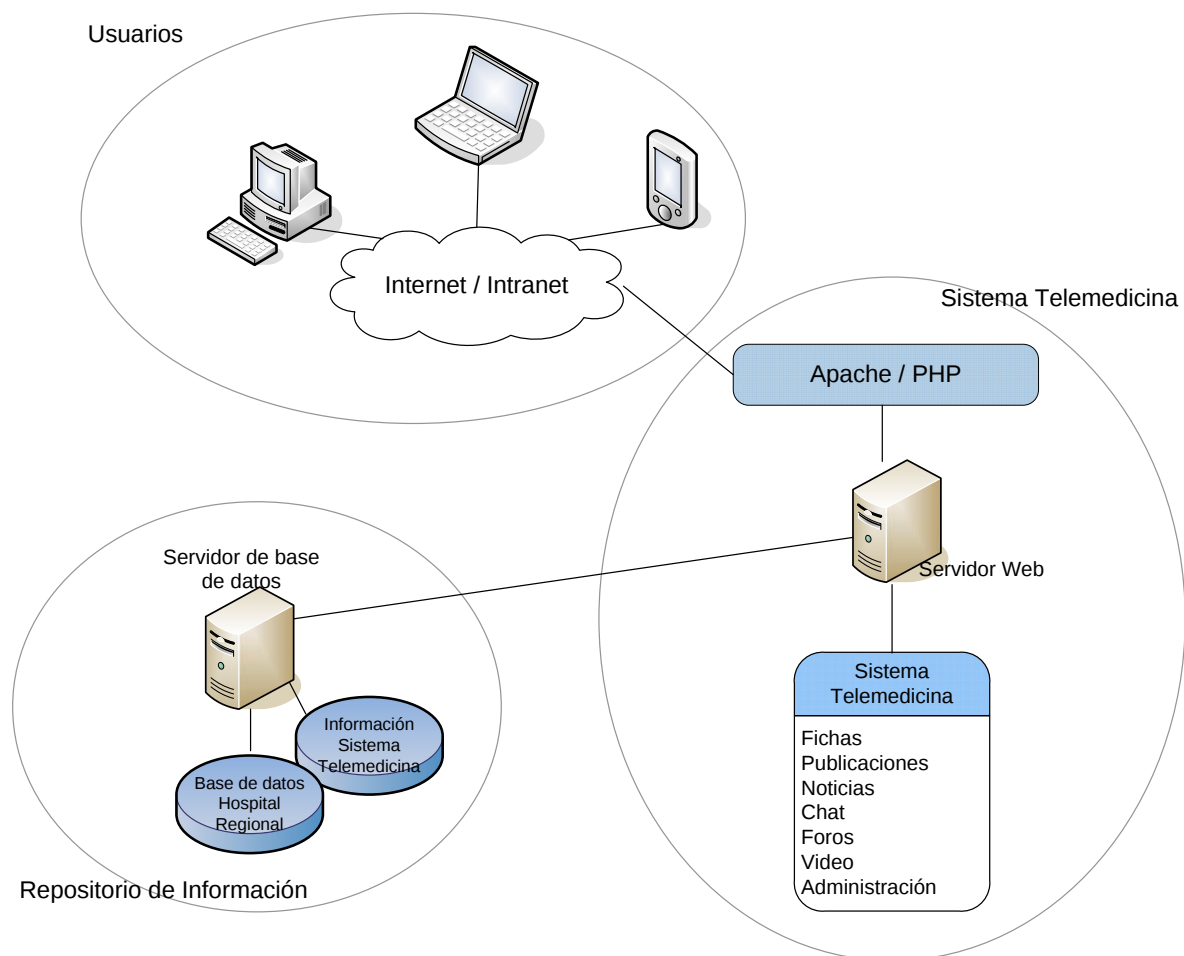


Figura 3.16: Modelo detallado.

La estructura jerárquica y flujo de información del sistema Web, se muestra en la figura 3.17, donde cada uno de los bloques desde el tercer nivel corresponde a un directorio en el servidor, que contiene los archivos asociados a sus funcionalidades.

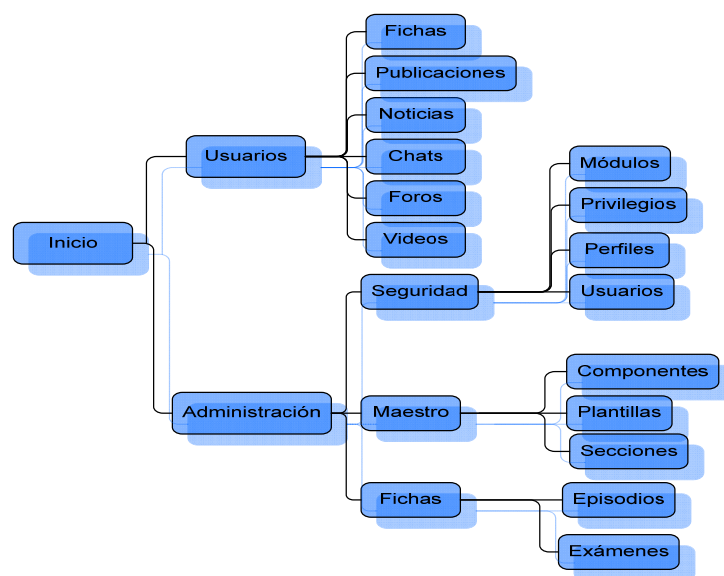


Figura 3.17: Relación entre módulos, estructura del sistema.

Como punto de partida en el momento de utilizar el programa, y en base a los privilegios asociados a cada perfil, los usuarios tienen la posibilidad de ingresar en un modo normal de operación o como administrador.

Una vez determinada la información relevante y necesaria para el sistema, además de las herramientas más apropiadas para la elaboración del software, se inició la etapa de diseño.

El sistema en cuestión utilizaría la base de datos del Hospital Regional de Valdivia (Oracle), de la que además obtendrá parte de la información que contiene, como son los siguientes datos: información general de pacientes, grupos sanguíneos, grupos crónicos (relacionado a las enfermedades crónicas presentes en los pacientes), códigos y nombres de comunas, etc., a parte, el sistema Telemedicina utilizará sus propias tablas, las que se analizarán con detalle junto con las aplicaciones que dependen y / o interactúan con ellas.

3.4.2 Usuarios del sistema

Es necesario determinar quienes serán los actores en el sistema, así, se podrán ordenar de acuerdo a perfiles y permisos, estableciendo los niveles de acceso al sistema por grupos, además de la estructuración del módulo de validación junto a la información necesaria y las tablas que deberán crearse en la base de datos del hospital.

En el área de obstetricia participan los siguientes roles definidos:

Secretaria: actualmente es la encargada de traspasar la información del papel al sistema computacional de fichas, de manipular y operar dicha aplicación.

Médico: es quién realiza el o los exámenes, llena la ficha y entrega un diagnóstico del paciente. Se diferencian 2 tipos de médicos: el especialista y el general, siendo el primero quien tiene la responsabilidad más alta en el sistema, puesto que es él quien tiene el conocimiento del área médica en cuestión.

Jefe de área o encargado de la especialidad: es quien administra la especialidad o área de la medicina de la cual es especialista, gestiona lo ocurrido en el establecimiento, incluyendo infraestructura y personal.

Administrador del sistema: es el técnico o ingeniero a cargo del sistema de información del hospital, quien mantiene el servidor central y la base de datos general. Normalmente es un equipo de profesionales.

Paciente: es la persona que acude al recinto hospitalario como cliente. Aunque el sistema se ha diseñado pensando en el paciente como un conjunto de datos, podría en futuras actualizaciones considerársele como un usuario de nivel bajo y así pueda revisar su historial médico por Internet, mediante una clave de acceso. Esto queda para posteriores versiones, ya que se necesita un análisis mayor, no solo en cuanto a software (es casi trivial agregar un nuevo tipo de usuario), si no más bien un debate de las repercusiones que pueda traer el que el historial médico de una persona esté disponible a través de un medio público.

3.4.2.1 Privilegios y perfiles

Cada usuario tiene asociado un perfil, compuesto de uno o más privilegios o permisos, los cuales acotan el acceso a los diferentes módulos, siendo estos: Crear, Visualizar, Editar y Eliminar. Esto se puede ver en la tabla 3.1.

Tabla 3.1: Tabla genérica de privilegios y perfiles.

Módulo \ Privilegio	Crear	Visualizar	Editar	Eliminar
Cuentas de usuario				
Módulos, privilegios y perfiles				
Componentes, secciones y plantillas				
Fichas Clínicas de pacientes				
Episodios				
Exámenes				
Interconsultas				
Publicaciones				
Noticias				
Chat				
Foros				
Videos				

Los valores posibles para cada uno son: **G**eneral, **E**specialidad o **N**inguno.

A continuación se detalla los perfiles definidos por defecto en el Sistema, junto a su tabla de privilegios:

Administrador del Sistema: Perfil con privilegios de administración sobre todos los módulos, que se utiliza principalmente por los desarrolladores y área informática encargada de mantener el Sistema, como se aprecia en la tabla 3.2.

Tabla 3.2:Privilegios del administrador del sistema.

Módulo \ Privilegio	Crear	Visualizar	Editar	Eliminar
Cuentas de usuario	G	G	G	G
Módulos, privilegios y perfiles	G	G	G	G
Componentes, secciones y plantillas	G	G	G	G
Fichas Clínicas de pacientes	G	G	G	G
Episodios	G	G	G	G
Exámenes	G	G	G	G
Interconsultas	G	G	G	G
Publicaciones	G	G	G	G
Noticias	G	G	G	G
Chat	G	G	G	G
Foros	G	G	G	G
Videos	G	G	G	G

Administrador de especialidad: Perfil con privilegios para moderar comentarios de foros, publicar noticias, editar o crear nuevos episodios y/o exámenes, pero solamente sobre la especialidad a la que pertenece (tabla 3.3). Una de sus tareas principales es la de autorizar o denegar las solicitudes de creación de cuentas de usuario realizadas desde la página de registro.

Tabla 3.3: Privilegios administrador de especialidad.

Módulo \ Privilegio	Crear	Visualizar	Editar	Eliminar
Cuentas de usuario	E	E	E	N
Módulos, privilegios y perfiles	N	N	N	N
Componentes, secciones y plantillas	N	E	N	N
Fichas Clínicas de pacientes	E	E	E	N
Episodios	E	E	E	N
Exámenes	E	E	E	N
Interconsultas	E	E	E	E
Publicaciones	E	E	E	E
Noticias	E	E	E	E
Chat	E	E	E	E
Foros	E	E	E	E
Videos	E	E	E	E

Médico especialista: La tabla 3.4. Describe el perfil con privilegios para crear, visualizar y editar episodios y exámenes en la especialidad a la que pertenece. Además, puede solicitar interconsultas, realizar publicaciones, crear y visualizar foros, crear, visualizar y editar videos.

Tabla 3.4: Privilegios médico especialista.

Módulo \ Privilegio	Crear	Visualizar	Editar	Eliminar
Cuentas de usuario	N	N	N	N
Módulos, privilegios y perfiles	N	N	N	N
Componentes, secciones y plantillas	N	N	N	N
Fichas Clínicas de pacientes	N	E	N	N
Episodios	E	E	E	N
Exámenes	E	E	E	N
Interconsultas	E	E	E	E
Publicaciones	E	E	E	N
Noticias	N	E	N	N
Chat	N	E	N	N
Foros	E	E	N	N
Videos	E	E	E	N

Médico general: Perfil con privilegios para visualizar fichas clínicas y crear episodios y exámenes generales (tabla 3.5).

Tabla 3.5: Privilegios médico general.

Módulo \ Privilegio	Crear	Visualizar	Editar	Eliminar
Cuentas de usuario	N	N	N	N
Módulos, privilegios y perfiles	N	N	N	N
Componentes, secciones y plantillas	N	N	N	N
Fichas Clínicas de pacientes	N	E	N	N
Episodios	E	E	N	N
Exámenes	E	E	N	N
Interconsultas	E	E	E	E
Publicaciones	E	E	E	N
Noticias	N	E	N	N
Chat	N	E	N	N
Foros	N	E	N	N
Videos	N	E	N	N

Secretaria: Perfil con privilegios para visualizar fichas clínicas, episodios, exámenes, publicaciones y noticias (tabla 3.6).

Tabla 3.6: Privilegios de Secretario (a).

Módulo \ Privilegio	Crear	Visualizar	Editar	Eliminar
Cuentas de usuario	N	N	N	N
Módulos, privilegios y perfiles	N	N	N	N
Componentes, secciones y plantillas	N	N	N	N
Fichas Clínicas de pacientes	N	E	N	N
Episodios	N	E	N	N
Exámenes	N	E	N	N
Interconsultas	N	N	N	N
Publicaciones	N	E	N	N
Noticias	N	E	N	N
Chat	N	N	N	N
Foros	N	N	N	N
Videos	N	N	N	N

3.4.3 Diseño de Interfaces

Antes de implementar los formularios e informes, fue necesario diseñar su aspecto, actividad para la que se consideraron los siguientes parámetros:

- a) Utilizar títulos que sean significativos, que identifiquen sin ambigüedad el propósito del informe o formulario: Hojas de estilo, títulos propios del bloque o módulo, identificador de la especialidad (para así evitar confusiones, por ejemplo: “Foro área de Obstetricia”)
- b) Dar instrucciones breves y fáciles de comprender: frases cortas, concisas y precisas; instrucciones coherentes por ejemplo: “Ingrese su RUT”.
- c) Agrupar y secuenciar los campos de forma lógica: empleando una estructura de secciones y subsecciones, además se mantiene el orden original de despliegue de las fichas en papel (modificable mediante un código de orden de despliegue).
- d) Hacer que el aspecto del informe o formulario sea atractivo a la vista: basados en un diseño limpio sin sobrecarga de información; división de la pantalla en

cabecera, cuerpo y pie de página; tipografía clara y distinguible (familia arial y verdana); las ventanas no traspasen el límite de la pantalla (función que determina y ajusta el sistema a la resolución de la pantalla del usuario).

- e) Utilizar nombres familiares para etiquetar los campos: basados en la información que contienen e identificando el grupo, por ejemplo: “rut”, “mail”, “Pared abdominal”.
- f) Utilizar terminología y abreviaturas consistentes: se recabaron de los documentos originales (papel) y de las conversaciones con el personal del hospital, por lo que en su mayoría es terminología médica a la que los usuarios están familiarizados, por ejemplo: LCN, “Longitud Céfaló Nalga”.
- g) Utilizar en forma razonable y consistente los colores: se privilegiaron colores suaves, agradables a la vista y relajantes, combinando unos con otros; en cuanto a las letras, se optó por colores que resalten respecto de los fondos, para que no signifique un esfuerzo leer en la pantalla.
- h) Dejar un espacio visible para los datos de entrada y delimitarlos: cada formulario está debidamente destacado, indicándose además como debe llenarse en base a un ejemplo.
- i) Permitir un uso sencillo y adecuado del cursor: mediante enlaces fácilmente identificables e intuitivos en su función.
- j) Permitir la corrección carácter a carácter y de campos completos: al tratarse mayoritariamente de cajas de texto, esto se cumple siempre.
- k) Dar mensajes de error para los valores “ilegales”: para cada entrada crítica fue programado el manejo de errores, evitando el ingreso de valores o tipos no permitidos y de acuerdo a los formatos establecidos; se envían mensajes al usuario para que corrija la entrada y reingrese los datos.
- l) Marcando los campos que son opcionales. Dando mensajes a nivel de campo para explicar su significado.

3.5. Aseguramiento de la calidad del software

“Los requisitos del software son la base de las medidas de calidad, por consiguiente, una falencia en este aspecto se traduce en una falta de calidad” [CUE99].

Bajo esta premisa se enfocó el aseguramiento y la gestión de la calidad del software implementado, al no emplear una metodología siempre habrá falta de calidad en el producto final. Lo anterior se aplica tanto a los requisitos explícitos (por ejemplo, registro de usuario, los datos del paciente, etc que han sido solicitados), como a los requisitos implícitos o expectativas, que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo que el programa sea fácil de actualizar e integrable con otras aplicaciones).

Para lo anterior se cumplió con los siguientes pasos:

- a. Inspecciones en todos los pasos del proceso de desarrollo del software, resumidas en pruebas de funcionamiento basadas en diferentes entradas con resultado conocido, verificación de la consistencia de los datos desplegados y almacenados en la base de datos, finalmente, pruebas de desempeño y facilidad de uso para usuarios de un nivel normal-básico, empleando terceras personas sin experiencia en sistemas de información.
- b. Control del software y de los cambios realizados, por medio de comentarios en el código, explicando detalladamente que función cumple cada bloque, cada archivo y cada directorio. Por otro lado, se realizaron versiones de los distintos prototipos de acuerdo a metas auto impuestas, conservando las versiones anteriores, es decir, al incluir una nueva funcionalidad en el prototipo, se realizaba un respaldo del trabajo, se llevaban a cabo las inspecciones de rigor (véase punto anterior) y se continuaba con la versión siguiente.
- c. Procedimientos para ajustarse a los estándares definidos por el equipo y por el trabajo anterior existente. Esto se refiere a todo el software implementado en el Hospital Regional, específicamente a la base de datos, de la que se mantuvo la estructura y forma de nombrar las tablas. Como estándar propio se puede mencionar la estructura de

directorios, los nombres de funciones procedimientos, variables y archivos (basados en el estándar del lenguaje C++).

Una vez terminado el producto-software, se vio la necesidad de comprobar la calidad del software lograda en el proceso, para esto se realizó un análisis basado en los tres grandes factores determinantes de la calidad:

a. Operación, características operativas

a.1. Corrección: el software logrado ¿Hace lo que se pidió que haga?. Esta pregunta permite un análisis que desemboca en 2 vertientes: requisitos explícitos e implícitos.

a.1.1. Requisitos funcionales desprendidos del análisis de requerimientos:

Como se menciona en el apartado de análisis, los requerimientos fueron obtenidos por medio de procedimientos ordenados, por etapas (de acuerdo al nivel de generalización), para finalmente ser resumidos en el punto 3.2.3. En las conclusiones de este trabajo de tesis se indica el grado de completitud que se logró para cada requisito y su estado al finalizar el software.

a.1.2. Requisitos no funcionales, desprendidos del análisis inicial, listados en el punto 3.2.4, satisfaciendo las especificaciones y consiguiendo gran parte de los objetivos encomendados por el cliente.

a.2. Fiabilidad, es decir, ¿El programa trabaja de forma fiable todo el tiempo?. Para determinar esto se realizaron pruebas de uso, en las cuales se introdujeron diversas entradas (alfanuméricas, con y sin formato, de largo variable, correctas y erróneas de acuerdo al contexto, etc.), obteniéndose en la versión final una excelente tolerancia a fallos de ingreso de valores en los formularios. De los fallos producidos, el 100% de debió a problemas en la conexión con el servidor, lo que se considera externo al diseño y programación de las aplicaciones. Para esto se aplicó la metodología de revisión por pares o arbitraje.

a.3. Eficiencia, medida bajo la pregunta: ¿Qué recursos hardware y software se necesitaron para la implementación del sistema?.

Una de las principales preocupaciones al momento de programar cada módulo, fue la utilización eficiente de los recursos software y hardware, debido a que se consideraron siempre como factores críticos dentro del proyecto. Se optó por emplear un diseño estructurado, siendo el orden, facilidad de mantenimiento, revisión y fundamentalmente la poca carga de trabajo al servidor, sus características más atractivas.

La carga de trabajo a mediano plazo corresponderá a los datos asociados al número de habitantes de la provincia. Debido a que el Hospital base de Valdivia es considerado como hospital regional. En principio se considera como carga solo a los habitantes de la comuna (aproximadamente 140.000). Los requerimientos de hardware están asociados directamente al crecimiento de la base de datos e incremento en el número de pacientes, por lo que el sistema podría requerir un aumento en cuanto a capacidad de almacenamiento y procesamiento, esto de acuerdo a la documentación del motor de base de datos Oracle [REF, 07].

a.4. Integridad, es decir, ¿Puedo controlar su uso?; obteniéndose una respuesta afirmativa a esa pregunta. El sistema fue diseñado de tal manera que tanto a nivel de usuarios, como programación, el software es íntegro.

Para los usuarios el sistema discrimina por medio de privilegios y especialidades, los que determinan los permisos de acceso, por ejemplo, un usuario del área de obstetricia de nivel normal, podrá acceder solo a la información del área de obstetricia, en cuanto a pacientes, foro, publicaciones, noticias, etc. y este usuario podrá crear fichas clínicas de pacientes solo para dicha área, no teniendo la posibilidad de modificar nada más excepto aquello sobre lo cual tenga privilegios. Por otro lado, un administrador de especialidad, podrá modificar cualquier cosa que pertenezca a su área, incluyendo los permisos de acceso de otros usuarios. Todo lo anterior se resguarda con un registro de modificaciones, que indica la fecha, hora y responsable de cada cambio efectuado a la información almacenada en el sistema.

A nivel de programación, el sistema posee un archivo de configuración, donde se pueden editar varias opciones, incluyendo conexión al servidor y a la base de datos, haciéndolas fácilmente modificables de acuerdo a los requerimientos que se impongan. Los archivos fuentes y en general el sistema, estará ubicado en el servidor del hospital, por lo que solo tendrán acceso los usuarios autorizados, y cuya seguridad escapa al alcance de esta tesis.

Las pruebas realizadas en este apartado se dividen en 2 áreas: la primera, intentando realizar acciones que corresponden a usuarios de nivel avanzado, pasado de manera exitosa, puesto que el sistema no permitió efectuar acciones que no concordaban con los privilegios del usuario activo; la segunda referida a afectar el sistema por medio de “inyecciones de código”, las que corresponden a la inserción de trozos de código malicioso en formularios o direcciones, determinándose que un buen “firewall” (o cortafuegos), soluciona los accesos indebidos y anula dichas acciones.

a.5. Facilidad de uso, donde cabe preguntar si el sistema ¿Es fácil y cómodo de manejar?. Esto significó un esfuerzo importante porque el equipo desarrollador del sistema no posee formación en cuanto a diseño gráfico. El hecho de ser usuarios avanzados influyó en que muchas cosas triviales (como combinaciones de teclas o formas de utilizar un navegador de Internet), no necesariamente lo son para las demás personas. Para solucionar este inconveniente, se probó el sistema con usuarios normales (con conocimientos básicos) y así se determinó las modificaciones necesarias para hacerlo mas intuitivo. El resultado puede observarse al utilizar el software final, en el que están implementadas muchas de las características desprendidas de esta fase de prueba, por ejemplo, las combinaciones de colores, la distribución de los componentes y formularios, la ubicación de los menús, las descripciones de los módulos, las indicaciones generales y la ayuda.

b. Revisión del producto, soporte de cambios.

b.1. Facilidad de mantenimiento, ¿Puedo localizar los fallos?. Debido a que el programa ha sido desarrollado en PHP, es muy fácil identificar los fallos a la hora de programar, también por utilizarse una metodología estructurada de programación, el seguimiento y corrección de los errores es una tarea relativamente sencilla, al estar debidamente comentado, no se requiere un gran esfuerzo para localizar bloques de código y repararlos. Del mismo modo, la estructura de directorios está ordenada, es decir, no hay desgaste en ubicar un archivo o carpeta dentro del sistema.

b.2. Flexibilidad, es decir, ¿Puedo añadir nuevas opciones?. El sistema ha sido diseñado con éste propósito, no hay que olvidar que el módulo de obstetricia se considera como un “prototipo funcional”. Dentro de los requisitos se nombraba la capacidad de ampliación, de modo que una nueva especialidad u otros episodios con sus exámenes puedan ser configurados dentro del sistema, para esto fueron creadas las herramientas administrativas y diseñada una estructura flexible para la incorporación de nuevas funcionalidades.

b.3. Facilidad de prueba, ¿Puedo probar todas las opciones?. El único impedimento para lograrlo es el nivel de acceso, en este caso, solo el usuario “administrador” podrá probar cada una de las funcionalidades.

c. Transición, adaptabilidad.

c.1 Portabilidad, ¿Podré usarlo en otra máquina?. Si, el sistema se ha programado para plataformas Web, por lo que cualquier computador con acceso a la red del hospital, que posea un navegador de Internet podrá utilizarlo. Por otra parte, el sistema ha sido programado en PHP, lo que significa que, teniendo un servidor Web que soporte dicho lenguaje se podrá habilitar para su uso. El mayor problema en este caso lo presenta la base de datos Oracle, tanto por el tema de las licencias, como por los requisitos mínimos de instalación, sin embargo, el sistema se conecta a la base de datos a través de

procedimientos y funciones (configurados en el archivo de configuración del sistema), utilizar una base de datos distinta significa intervenir dichas funciones y procedimientos, lo que no es trivial, pero tampoco necesitará de un gran esfuerzo.

c.2. Reusabilidad, ¿Podré utilizar alguna parte del software en otra aplicación?. Como se planteó en el anteproyecto de la tesis, el trabajo realizado sería por medio de reutilización, recopilación y programación de funcionalidades, es decir, cada procedimiento, función y módulo tienen la posibilidad de ser reutilizados, tanto en áreas afines como en no afines, por ejemplo la función “obtieneDato()” (anexo 7.3), que busca datos de una base de datos utilizando como criterio los parámetros que se le incorporen; los módulos de foro, chat, publicaciones, noticias, etc.

c.3. Interoperabilidad, ¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?. Si, debido a que toda la información se guarda en la base de datos, basta con acceder a dicha base de datos para interactuar con el sistema.

3.6. Implementación, validación y prueba.

3.6.1 Implementación

El proceso de implementación será labor del equipo encargado de administrar el sistema de información del hospital base, debido a las restricciones de acceso impuestas por el medio, es imposible manipular la base de datos, o reconfigurar el servidor.

El actual motor de base de datos que utiliza el hospital es Oracle 8i, en proceso de migración a Oracle 10G. De acuerdo a las especificaciones técnicas de dicho motor, no existen problemas de compatibilidad con el sistema telemedicina, gracias a la retrocompatibilidad que tiene con sus versiones anteriores, las nuevas librerías, y a la sencillez de las consultas realizadas por parte del programa desarrollado.

Como se menciona en los requerimientos de hardware, es necesaria la instalación del software apache (servidor Web, soporte html y php) y habilitar las librerías de acceso

a motor de base de datos. Este proceso ha sido probado por parte del equipo de desarrollo, en colaboración a un operario del sistema de información del Hospital Regional de Valdivia, en aplicaciones Web que empleaban registros reales de la base de datos hospitalaria.

Se asume que un equipo calificado no debe tener problemas a la hora de configurar un servidor Web, con soporte del lenguaje php y conectado a una base de datos Oracle.

3.6.2 Validación y prueba

El proceso de validación y prueba se orientó hacia dos ámbitos: local y médico. En forma local, se efectuaron revisiones por pares y con colaboración de terceras personas. Se trabajó con usuarios inexpertos, permitiendo mejorar la accesibilidad, como también el menú de ayuda del sistema. Esto sirvió para detectar inconsistencias y errores en el programa. El prototipo final, responde a las modificaciones realizadas gracias al trabajo en éste apartado.

A nivel médico, el trabajo se realizó en prototipos iniciales, donde el primero de ellos se evaluó al iniciar la programación, indicando la información que debía almacenar el programa, los apartados y secciones. El segundo prototipo se llevó a cabo con colaboración del personal médico, quienes probaron una versión instalada en dependencias de la escuela de Ingeniería Civil en Informática de la Universidad Austral de Chile, en un servidor facilitado por el Servicio de Salud. Se indujeron modificaciones en la forma de despliegue de la información, en las pantallas del sistema. De cierto modo, se instó a mejorar la “intuitividad” o facilidad de uso sin conocimientos previos de la aplicación.

Aún falta una etapa formal de prueba, la que debería producirse al momento de instalar el software en el sistema del Hospital Base. Pese a lo anterior, el Ministerio de Salud ha demostrado interés en la implementación de sistemas similares, la Ministra de salud (a la fecha en que se realizó la aplicación), ha indicado que el Gobierno está

realizando una modernización en el sistema de salud, con la meta “utópica” de unificar el sistema de información hospitalario a nivel nacional, iniciándose con la implementación de sistemas locales para su evaluación [dichos efectuados en el mes de diciembre del año 2006], de tal suerte, que el trabajo puede ser derivado al Servicio de Salud de la provincia de Valdivia. Este paso sería crucial, debido a que una evaluación favorable permitiría instalar la aplicación en los recintos hospitalarios de la provincia, no existiendo una mejor manera de poner a prueba el proyecto telemedicina.

4. Conclusiones

Al alero del sistema de telemedicina, se ha logrado:

- Optimizar el modelo tradicional de fichas en papel; reducción de costos, tiempo y recursos, esto es, de períodos que podían demorar días (debido al traslado de las fichas), a solo un par de minutos en el navegador Web. Reducción en costo de artículos de oficina y personal dedicado a la labor de traspaso de información.
- Mejorar significativamente la forma tradicional de comunicarse entre recintos hospitalarios; envío de noticias y comunicados al instante.
- Reducir los tiempos de envío y recepción de la información de los pacientes dentro y entre establecimientos, ya que al tratarse de una base de datos en línea los cambios se reflejan instantáneamente.
- Canal de comunicación expedito entre funcionarios, discriminado por especialidad y de forma directa, individual o por grupos.
- Eliminar la inconsistencia y replicación de datos, propio de los sistemas antiguos, por el hecho de trabajar en una base de datos común, se unifican los formatos y los tipos de entradas al sistema se homologan, eliminando errores tipográficos e inconsistencias en algunos campos críticos.
- Fácil acceso al historial clínico de los pacientes, guardado en la base de datos central.

Revisando los objetivos propuestos en el anteproyecto original de la tesis, se concluye el logro de los siguientes:

- Investigación y recopilación de información referida a la atención hospitalaria, reflejada en la realización de ésta tesis.
- Investigación de herramientas informáticas adecuadas para la realización del proyecto, de acuerdo al capítulo 3, punto 3.2.5.1.

- Diseño e implementación de una solución informática que permita automatizar –en parte- el sistema de atención hospitalaria al nivel del profesional de la salud con herramientas tales como: fichas electrónicas en línea y acceso a información centralizada; permitiéndole mejorar y agilizar dicho proceso; comprobándose en el prototipo resultante.
- Diseño e implementación de un sistema central de noticias y comunicados que permita mantener un registro de las políticas y normas a adoptarse en los distintos Centros de Salud, con la finalidad de agilizar su difusión, de acuerdo al punto 3.4.
- Diseño e implementación de un entorno Web que permita almacenar información y documentación, en un medio centralizado y accesible desde cualquier computador perteneciente a la red de salud.

En deuda han quedado los siguientes puntos:

- Diseño e implementación de un sistema de apoyo a la atención hospitalaria en forma remota para realizar diagnósticos y / o consultas asistidas.
- Validar prototipo funcional del sistema.

Por efectos de falta de tiempo y recursos, el sistema no se probó el tiempo necesario, es ideal la posibilidad que se presenta de integrarlo por medio del Servicio de Salud Valdivia, completaría el proceso de marcha blanca que quedó inconclusa, además serviría para ampliarlo a las demás especialidades, integrando nuevas posibilidades, pero principalmente, para contribuir a la modernización del Estado, con quién hay un enorme compromiso.

Para Finalizar, se ha logrado diseñar una plataforma de software que permite a los profesionales de la salud apoyar la atención de pacientes en forma local y remota, disponiendo de un medio formal para el almacenamiento de información e intercambio de opiniones y experiencias sobre temas de interés.

5. Glosario

BCPL: Es un acrónimo inglés de Basic Combined Programming Language (Lenguaje de Programación Básico Combinado). Fue diseñado por Martin Richards de la Universidad de Cambridge en 1966 debido a las dificultades experimentadas con los lenguajes de programación durante los años 60. El primer compilador implementado fue escrito en 1967 mientras Richards visitaba el MIT. El lenguaje fue descrito por primera vez en un proyecto presentado en una conferencia informática en 1969.

BSD: Son las iniciales de Berkeley Software Distribution (en español, Versión de Software Berkeley) y se utilizan para identificar un sistema operativo derivado del sistema Unix nacido a partir de las aportaciones realizadas a ese sistema por la Universidad de California en Berkeley.

C: Es un lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell basándose en el lenguajes BCPL. Al igual que sus dos predecesores, es un lenguaje orientado a la implementación de Sistemas Operativos (los sistemas operativos Linux y UNIX están escritos mayormente en C), pero se ha convertido en un lenguaje de propósito general de los más usados.

C++: Lenguajes de programación (orientado a objetos en el caso de C++) utilizados en el World Wide Web a través de un CGI, principalmente para realizar consultas a bases de datos como Oracle, SQL-Server, SyBase, etc. Generalmente el servidor donde se encuentra el programa funciona en ambiente UNIX.

Ecografías: La ecografía o ultrasonografía es un procedimiento de radiología que emplea los ecos de una emisión de ultrasonidos dirigida sobre un cuerpo u objeto como fuente de datos para formar una imagen de los órganos o masas internas con fines de diagnóstico.

E-commerce: Es el tipo de transacción económica -compra y venta- que se realiza a través de sistemas electrónicos. Una empresa, comúnmente presente en la red, vende productos o servicios a través de Internet. Allí llega el comprador que suele pagar con tarjeta de crédito un producto que en un plazo acordado recibirá en la puerta de su casa.

Episodio: Circunstancia que forma parte de una serie de acontecimientos que constituyen un todo. En medicina se refiere a un acontecimiento ocurrido en un paciente, que genera exámenes y diagnósticos sobre el estado de éste.

Open Suorce: Código abierto (open source en inglés) es el término por el que se conoce al software distribuido y desarrollado en una determinada forma. Este término empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original, en inglés, del software libre (free software).

PHP: Personal Home Page. Lenguaje script avanzado para diseño de sitios.

GCI: Common Gateway Interface (en inglés «Pasarela de Interfaz Común», abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa.

GPL (General Public License): Es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre.

Host virtual: Se trata de servir diferentes sitios web con una sola entidad de Apache. El hosting virtual de IPs diferencia los sitios web basandose en sus direcciones IP, mientras que el hosting virtual basado en nombres usa solo el nombre del host y de esta manera puede alojar muchos sitios web con la misma dirección IP.

HTML: Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y

presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTTP (HyperText Transfert Protocol): Protocolo de Transferencia de HiperTexto.

Javascript: Es un lenguaje de computadora de plataforma independiente desarrollado por Netscape Communications en 1994 para que los autores Web diseñen los sitios Web interactivos. JavaScript no requiere acceso a las partes potencialmente 'sensibles' o importantes de su computadora, sistema o red. Son ejemplos de JavaScript en los sitios Web los receptores de cotizaciones de bolsa y los exámenes interactivos. A diferencia de Java, JavaScript es texto que está dentro de la página Web.

Linux: Sistema operativo derivado de UNIX que, manteniendo casi todas las ventajas que este último ofrece, puede ser ejecutado en computadoras personales. Fue desarrollado originalmente por el estudiante finlandés de informática Linus Torvalds, que publicó su código fuente en 1990, en la forma de código abierto.

Md5: En criptografía, MD5 (acrónimo de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. El código MD5 fue diseñado por Ronald Rivest en 1991.

MIT: Es el acrónimo de Massachusetts Institute of Technology (Instituto Tecnológico de Massachusetts), una de las principales instituciones dedicadas a la ciencia, la ingeniería y la investigación. Está situado en los Estados Unidos y se dedica a la docencia y a la investigación. Cuenta con numerosos premios Nobel.

Mysql: Es un gestor de Bases de Datos multiusuario que gestiona bases de datos relacionales poniendo las tablas en ficheros diferenciados. Es muy criticado porque carece de muchos elementos vitales en bases de datos relacionales y no es posible lograr una integridad referencial verdadera. Es mas utilizado en plataformas Linux aunque puede usarse en otras plataformas. Su uso en un

servidor web es gratuito salvo en los casos que se necesite el uso de aplicaciones especiales.

Oracle: Es un sistema de administración de base de datos (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Perl: Perl (Practical Extraction and Report Language) es un lenguaje de programación desarrollado por Larry Wall (lwall at netlabs.com) inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell, para la administración de tareas propias de sistemas UNIX.

Phpmyadmin: En las palabras del desarrollador: phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas webs, utilizando la Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas.

PostgreSQL: Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto, así como sistemas propietarios.

Proxy: Servidor especial encargado, entre otras cosas, de centralizar el tráfico entre Internet y una red privada, de forma que evita que cada una de las máquinas de la red interior tenga que disponer necesariamente de una conexión directa a la red. Al mismo tiempo contiene mecanismos de seguridad (firewall o cortafuegos) los cuales impiden accesos no autorizados desde el exterior hacia la red privada. También se le conoce como servidor cache.

Radiografías: Imagen extraída de una placa bombardeada con Rayos X que muestra los elementos duros de un objeto (o ser vivo) para descubrir fisuras o daños internos en partes que no son visibles a simple vista (por ejemplo: huesos rotos).

RDBMS: Es un Sistema Administrador de Bases de Datos Relacionales. RDBMS viene del acrónimo en inglés Relational Data Base Manager System.

Semiología: En medicina, se refiere al estudio de los síntomas físicos y mentales.

Unix: UNIX® (o Unix) es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

URL (Uniform Resource Locator: Localizador Uniforme de Recursos.

Weblogs: Un weblog, también llamado blog o bitácora, es un sitio web donde se recopilan cronológicamente mensajes de uno o varios autores, sobre una temática en particular siempre conservando el autor la libertad de dejar publicado lo que crea pertinente.

World Wide Web Consortium (W3C): Es una organización que produce estándares para la World Wide Web (o Telaraña Mundial). Está dirigida por Tim Berners-Lee, el creador original de URL, HTTP y HTML sobre el que se basa la Web.

Xml: Es el acrónimo del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C).

6. Referencias

- **[CUE99]:** Cueva, Juan Manuel, “*Calidad del Software*”, Universidad de Oviedo 1999.
- **[ING01]:** Sommerville, Ian, “*Ingeniería de Software*” Prentice Hall 2001, sexta edición.
- **[KEN97]:** Kendall, Kenneth E., “*Análisis y diseño de sistemas*”, Prentice Hall 1997, tercera edición.
- **[UML99]:** Larman, Craig, “*UML y Patrones*”, Prentice Hall 1999.
- **[REF,01]:** Phpmyadmin.
<http://www.phpmyadmin.net/>
- **[REF,02]:** Postgre SQL.
<http://www.postgresql.org/>
- **[REF,03]:** Easy php.
<http://www.easyphp.org/>
- **[REF,04]:** Wikipedia en español
<http://es.wikipedia.org/>
- **[REF,05]:** Modelo entidad-relación
<http://www3.uji.es/~mmarques/f47/apun/node83.html>
- **[REF,06]:** GTK.
<http://www.gtk.org/>
- **[REF,07]:** Oracle.
<http://www.oracle.com/>
- **[REF,08]:** Base de datos en castellano
<http://www.programacion.net/bbdd/tutorial/modrel/2/>
- **[REF,09]:** International Engineering Consortium
<http://www.iec.org>

7. ANEXOS

7.1. Estimación de costo esfuerzo según COCOMO II

El sistema cuenta con 14 módulos, cuyos tamaños han sido estimados mediante puntos de función. El modelo usado es el diseño anticipado COCOMO II 1999, Bohem B, y a continuación se detalla cada una de las fases para la elaboración del plan de trabajo.

Selección de factores de escala:

Precedencia del desarrollo (PREC): Se considera nominal, debido a la escasa experiencia con sistemas de éste tipo.

Flexibilidad del desarrollo (FLEX): Se considera nominal, ya que no se trata de un proyecto en el cual se perciba remuneraciones, lo que no es un factor crítico para terminarlo en el plazo fijado.

Arquitectura / Resolución de riesgos (RESL): Se considera alta, ya que es un sistema nuevo sin experiencias previas.

Cohesión del equipo (TEAM): Se considera alto, ya que en el área de desarrollo hay tendencia a trabajar en forma individual.

Madurez del proceso (PMAT): Según el modelo de Madurez de Capacidad del Instituto de Ingeniería del Software (CMM) utilizando las líneas de código estimadas del proyecto, se concluye que se trata de un factor de mucha importancia (VHI).

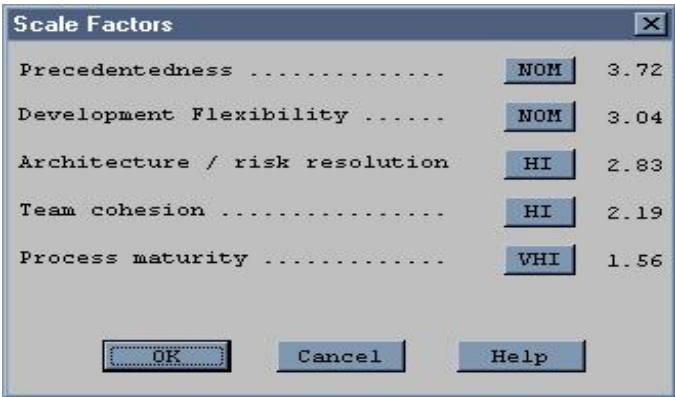


Figura 7.1: resumen factores de escala COCOMO II:

Drivers de costes para el modelo anticipado

- I.- Fiabilidad del producto y complejidad (RCPX): Alta. Combina cuatro drivers de coste:
- Fiabilidad del software (RELY): Alta, puesto que una falla del sistema resultará muy difícil de resolver por parte del usuario final.
 - Documentos del ciclo de vida (DOCU): Alta, ya que ante cualquier error o modificación del sistema, será de mucha utilidad para que el programador pueda realizarlo rápidamente.
 - Complejidad del producto (CPLX): Alta. Por tratarse de un sistema nuevo, sin experiencias anteriores, y en un rubro distinto al de los desarrolladores.
 - Medida de la base de datos (DATA): Alta, dado el gran volumen de información que albergará el sistema.
- II.- Reutilización requerida (RUSE): Alta, debido a que se espera poder utilizar los mismos módulos de Obstetricia para desarrollar los restantes de las otras especialidades.
- III.- Dificultad de la plataforma (PDIF): Alto. Combina tres drivers de coste:
- Restricciones de tiempo (TIME): Extra alto. El sistema ocupará 100% de recurso tiempo, debido a que el servidor Web y de base de datos deben estar siempre disponibles.

- Restricciones de almacenamiento (STOR): Alto. La información almacenada, ya sea, documento, foto, ficha clínica, etc. aumenta progresivamente el tamaño de la BD.
- Volatilidad de la plataforma (PVOL): Alto. Se espera que no haya mayores cambios en el sistema una vez realizada la toma de requisitos, pero debido a que se trata de una versión prototipo está abierta a posibles mejoras a futuro.

IV.- Factores del personal (PERS): Nominal. Combina tres drivers de coste:

- Habilidad del analista: Nominal.
- Habilidad del programador: Nominal.
- Continuidad del personal: Nominal.

V.- Experiencia personal (PREX): Nominal. Combina tres drivers de coste:

- Experiencia en las aplicaciones: Bajo. El equipo no tiene experiencia en un sistema de este tipo.
- Experiencia en la plataforma: Nominal. El equipo maneja ambientes de interfaces gráficas y comunicaciones a través de Internet.
- Experiencia en el lenguaje: Nominal. El equipo tiene una experiencia de aproximadamente 5 años en lenguajes de programación de diversos tipos.

VI.- Facilidades (FCIL): Nominal. Combina dos drivers de coste:

- Uso de herramientas de software (TOOL): Muy bajo. Se usa solo para edición de código fuente y depuración de errores.
- Desarrollo Multilugar (SITE): Extra alto. El servidor Web y de base de datos estarán ubicados en un solo lugar, y las modificaciones y correcciones podrán hacerse desde cualquier lugar del mundo gracias al uso de Internet.

VII.- Planificación temporal (SCED): Alto. Se tratará en lo posible lograr los objetivos planteados en los plazos fijados.

Maintenance EAF - Project

base + incr % = rating

	RCPX	RUSE	PDIF	PERS	PREX	FCIL	USR1	USR2
base	HI	HI	HI	NOM	NOM	NOM	NOM	NOM
Incr%	0%	0%	0%	0%	0%	0%	0%	0%

EAF: 1.68

OK

Cancel

Help

Figura 7.2: resumen drivers de coste COCOMO II

Control de riesgos

Riesgos asociados con el tamaño del producto (RTP) según tabla 7.1, se considera marginal, ya que un aumento potencial del número de líneas de código (SLOC'S), puede retrasar la fecha de entrega del proyecto, pero por tratarse de un sistema no remunerado, no es crítico.

Riesgos asociados con el personal (RP). Se considera un factor crítico por el impacto de los siguientes puntos:

- Malas relaciones interpersonales en el equipo de trabajo puede significar pérdidas de tiempo en solucionar diferencias de opiniones, y retraso en la realización de las tareas planificadas.
- Poca motivación del equipo puede traducirse en un producto potencialmente de baja calidad.
- Renuncia de personal puede provocar un retraso y posible aumento en los errores de código, al retomar los nuevos integrantes tareas a medio realizar.

Riesgos asociados al ambiente de desarrollo (RAD). Se considera marginal, ya que en el caso de hacer una mala selección de las herramientas adecuadas de desarrollo y análisis, solo se producirían retrasos menores en la implementación de módulos que pueden ser nativos en otros lenguajes.

Riesgos tecnológicos (RTEC)

- Los problemas de integración Cliente – Servidor en las comunicaciones en tiempo real se consideran catastróficos, debido a que es indispensable para el apoyo a la atención hospitalaria en forma remota.
- El ancho de banda limitado del medio de comunicación (Red Internet) se considera crítico, ya que afecta en la calidad del sonido e imagen, pero no impide el funcionamiento del sistema.
- La Incompatibilidad del sistema con el hardware existente (ecógrafos, maquinas de rayos X) se considera catastrófico, ya que afecta directamente a uno de los módulos principales del sistema de fichas clínicas y asistencias médicas remotas.

Riesgos asociados con presupuesto y venta del producto (RPV).

- Se considera crítico, ya que debido a los factores anteriores se puede extender la fecha de término del proyecto, agotando así el presupuesto existente.
- Se considera catastrófico, ya que de no tener interfaces agradables a la vista, intuitivas y de fácil configuración, puede ocasionar rechazo por parte de los posibles compradores o usuarios finales del sistema.

Categoría:

RTP: Riesgos asociados con el tamaño del producto.

RP: Riesgos asociados con el personal.

RAD: Riesgos asociados al ambiente de desarrollo.

RTEC: Riegos tecnológicos.

RPV: Riegos asociado con presupuesto y venta del producto.

Impacto

- 1. Catastrófico.
- 2. Crítico.
- 3. Marginal.
- 4. Despreciable.

Riesgo	Categorí a	Probabilidad	Impacto
Aumento potencial del número de SLOC'S	RTP	25 %	3
Malas relaciones interpersonales	RP	15 %	2
Poca motivación del equipo	RP	10 %	2
Abandono por parte del personal	RP	10 %	2
Herramientas de desarrollo inadecuadas	RAD	5 %	3
Integración Cliente – Servidor	RTEC	60 %	1
Ancho de banda del medio de comunicación	RTEC	65 %	2
Incompatibilidad del sistema con hardware existente	RTEC	45 %	1
Se extiende la fecha de entrega y se acaba el presupuesto	RPV	30 %	2
Rechazo por parte de los posibles compradores	RPV	25 %	1

Tabla 7.1: Riesgos

Se asume el pago de US\$ 9,375 / hora de trabajo, con lo cuál se obtiene la siguiente tabla 7.2 de costos por módulo, junto con el tiempo estimado de desarrollo de cada uno (solo programación). El costo por hora se estima calculando US\$ 1500 por mes / 20 días / 8 horas al día.

Nº	Nombre del módulo	Tiempo de desarrollo (Semanas)	Costo (US\$)
1	Validación de usuarios	1	375
2	Ficha clínica	2	750
3	Chat	1	375
4	Foro	1	375
5	Vides	2	750
6	Voz	2	750
7	Calificación profesional	2	750
8	Video Conferencia	3	1125
9	Almacenamiento de imágenes	2	750
10	Verificación de dispositivos	2	750
11	Búsqueda de información	2	750
12	Fichas personales de médicos	1	375
13	Administración de usuarios	2	750
14	Almacenamiento de información	2	750

Tabla 7.2: Costo por módulo.

Principal COCOMO II con los 14 módulos del sistema, según figura 7.3:

Name: Telemedicina

Scale Factor

Sche

Development Model:

Early Desi

Module Name	Module Size	LABOR Rate (\$/month)	EEF	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff
Accion Usuario	S:1408	375.00	1.79	5.0	8.9	158.1	3340.62	2.4	
Atencion Clinica	F:32000	1125.00	1.79	112.8	202.5	158.1	227769.88	7.1	
	S:3300	375.00	1.79	11.6	20.9	158.1	7829.59	2.4	
	S:3872	375.00	1.79	13.7	24.5	158.1	9186.72	2.4	
	F:560	750.00	1.79	2.0	3.5	158.1	2657.32	4.7	
	F:740	750.00	1.79	2.6	4.7	158.1	3511.45	4.7	
At. profesional	S:3256	750.00	1.79	11.5	20.6	158.1	15450.39	4.7	
At. conferencia	F:900	1125.00	1.79	3.2	5.7	158.1	6406.03	7.1	
Gen. imagenes	S:1188	750.00	1.79	4.2	7.5	158.1	5637.30	4.7	
Gen. dispositivos	S:1540	1125.00	1.79	5.4	9.7	158.1	10961.43	7.1	
Reda informacion	F:1261	750.00	1.79	4.4	8.0	158.1	5983.70	4.7	
As Pers. Medicos	S:3696	375.00	1.79	13.0	23.4	158.1	8769.14	2.4	
At. usuarios	S:8800	750.00	1.79	31.0	55.7	158.1	41757.81	4.7	
At. Informacion	S:3344	1125.00	1.79	11.8	21.2	158.1	23801.95	7.1	

Detalle de los resultados:

		Estimated	Effort	Sched	PROD	COST	INST	Staff
Total Lines of Code:	65865	Optimistic	279.2	20.6	235.9	249952.43	3.8	13.5
		Most Likely	416.7	23.3	158.1	373063.33	5.7	17.9
		Pessimistic	625.1	26.4	105.4	559595.00	8.5	23.6

Figura 7.4: Detalle de los resultados

Describiendo los resultados obtenidos con COCOMOII, de acuerdo a la figura 7.4, para las 65865 SLOC´s tenemos que el esfuerzo ajustado es de 416.7 personas mes, el tiempo de desarrollo es de 23 meses aproximadamente, el costo total de desarrollo del proyecto es de 373063 dólares y un Staff de 18 personas. Considerando el valor del dólar en \$597.27 (13/12/2004), se tiene un costo en pesos Chilenos de \$ 222819535,1091 (doscientos veintidós millones ochocientos diecinueve mil quinientos treinta y cinco pesos).

Aclaración de resultados: la duración del proyecto se estimó en 6 meses de acuerdo a lo que establece COCOMOII para la etapa de diseño. Se incluyó en esta estimación temporal la implementación de el prototipo para obstetricia debido a los siguientes factores: primero, el proyecto no parte de cero debido a que en el módulo más importante (ficha clínica para obstetricia) ya se cuenta con información que solo debe implementarse y mejorarse en algunos aspectos; segundo, puesto que se trata de un proyecto de integración de tecnologías, existe una amplia gama de códigos a reutilizar; y finalmente, se debe recordar que el COCOMOII es una herramienta que estima el “esfuerzo relativo” y no la duración real de los proyectos.

Desglose de la pantalla de resultados

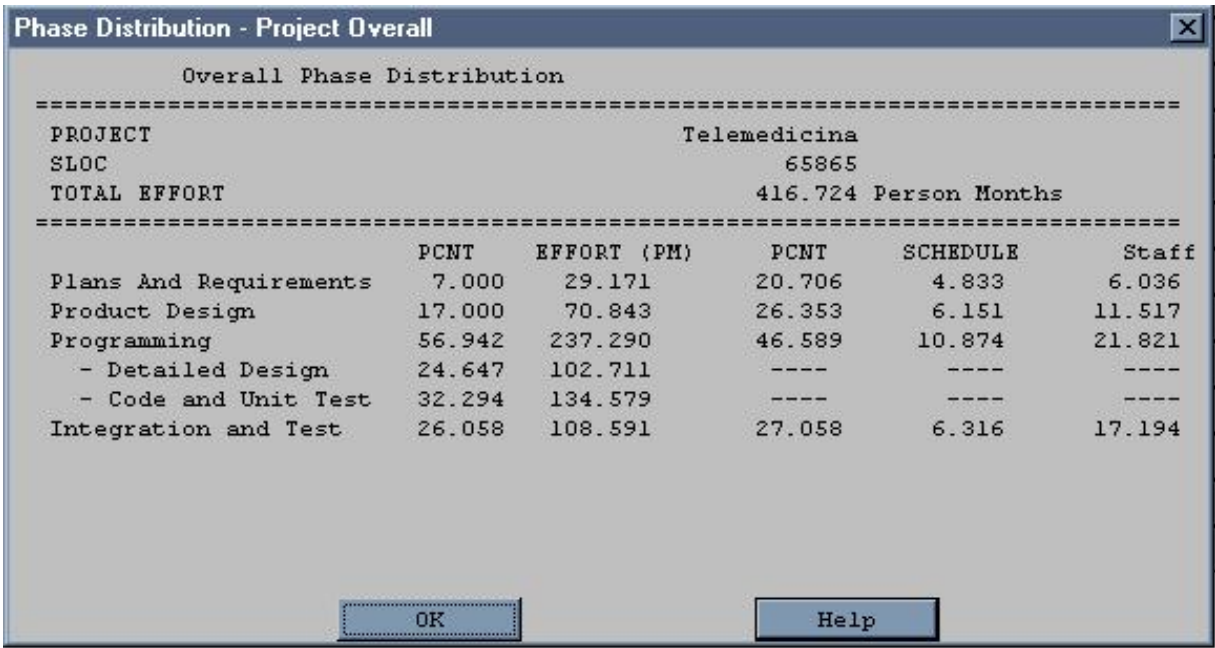


Figura 7.5: Distribución global del proyecto

A continuación se muestra la tabla 7.3 de resultados para cada fase del proyecto, teniendo mayor importancia las dos primeras, ya que serán éstas la que se desarrollarán en detalle como tema de tesis.

Fase de desarrollo	Tiempo (%)	Esfuerzo (p/m)	Tiempo requerido (m)	Personal
Planes y requerimientos	7	29,171	4,833	6,036
Diseño del producto	17	70,843	6,151	11,517
Programación	56,942	237,290	10,874	21,821
Integración y pruebas	26,058	108,591	6,316	17,194

Tabla 7.3: Resultados por fase

Se observa que el tiempo requerido para el diseño del sistema es de 6 meses, con lo cual se está en condiciones de elaborar el plan de trabajo de acuerdo a las 14 tareas a realizar.

El plan de trabajo detallado de la tabla 7.4 muestra los tiempos empleados para el diseño de cada tarea.

Actividades relacionadas con los objetivos específicos	Semanas empleadas en cada tarea						Responsable	Recursos asignados
	Mes 1	Mes 2	Mes 3	Mes 4	Mes5	Mes 6		
Análisis de requerimientos del medio	X						N. Hernández	Computador PowerEdge Pentium III 512MB RAM 2 discos duros Disquetera CD-ROM Teclado
Análisis de plataformas y herramientas	X						L. Faúndez	
Validación y administración de usuarios		X					N. Hernández	
Ficha clínica para obstetricia			X	X	X		N. Hernández	
Chat			X				L. Faúndez	
Foro				X			L. Faúndez	
Video Conferencia					X		L. Faúndez	
Fichas personales de médicos					X		N. Hernández	

7.2. Descripción de las tablas y sus campos:

AM_TSERVI Servicio

Grupo general, tabla de servicios (especialidades)

COD_SERVIC: Código servicio (Clave primaria)
NOM_SERVIC: Nombre de servicio

GG_TPROFE Profesión

Grupo Telemedicina, tabla de profesiones

COD_TPROFE: Código profesión (Clave primaria)
NOM_TPROFE: Nombre profesión

GG_TPROME Médico

Grupo General, tabla de profesionales médicos

COD_PROMED: Código del profesional (Clave primaria)
NOM_APEPAT: Apellido paterno
NOM_APEMAT: Apellido materno
NOM_NOMBRE: Nombres
COD_RUTPRO: Rut del médico, sin dígito verificador
COD_TPROFE: Código de profesión

GG_TCOMUN Comuna

Grupo General, tabla de comunas

COD_COMUNA: Código de comuna (Clave primaria)
NOM_COMUNA: Nombre de la comuna
COD_PROVIN: Código de la provincia

GG_TPROVI Provincia

Grupo General, tabla de comunas

COD_PROVIN: Código de provincia (Clave primaria)
NOM_PROVIN: Nombre de la provincia
COD_REGION: Código de la región

GG_TREGIO
Región

Grupo General, tabla de regiones

COD_REGION: Código de la región (Clave primaria)
NOM_REGION: Nombre de la región

GG_TESTABL
Establecimiento

Grupo General, tabla de establecimientos

COD_ESTABL: Código del establecimiento (Clave primaria)
NOM_ESTABL: Nombre del establecimiento
COD_RUTEST: Rut del establecimiento
NOM_DIRECC: Dirección del establecimiento

BS_TGRUSA
Grupo sanguíneo

Tabla de grupos sanguíneos

COD_GRUSAN: Código de grupo sanguíneo (Clave primaria)
NOM_GRUSAN: Nombre del grupo sanguíneo

RP_TPACTE
Paciente

Tabla de pacientes

NUM_CORPAC: Número de paciente (Clave primaria)
COD_NFICHA: Código de ficha
COD_RUTPAC: Rut del paciente, sin dígito verificador
NOM_NOMBRE: Nombre del paciente
NOM_APEPAT: Apellido paterno
NOM_APEMAT: Apellido materno
NOM_NPADRE: Nombre de la madre
NOM_NMADRE: Nombre del padre
NOM_PAREJA: Nombre de la pareja
IND_ESTCIV: Estado civil
COD_RUTTIT: Rut del titular
FEC_NACIM: Fecha nacimiento
IND_SEXO: Sexo
NOM_DIRECC: Dirección
NUM_TELEFO: Teléfono
COD_COMUNA: Comuna
COD_GRUSAN: Código grupo sanguíneo

GT_TSVSE
Servicio sección

Grupo Telemedicina, tabla secciones de servicio

COD_SVSE: Código servicio sección (Clave primaria)
COD_SERV: Código de servicio
COD_SECC: Código de sección
NUM_ORDE: Número de orden

GT_TSECC
Sección

Grupo Telemedicina, tabla secciones

COD_SECC: Código sección (Clave primaria)
NOM_SECC: Nombre de sección
COD_MODI: Código modificaciones
COD_PLAN: Código plantilla
NUM_REPL: Número de réplicas de la sección (1-50).
VAL_NRED: Valor número de réplicas por defecto (1-NUM_REPL)
COD_TREP: Código tipo de replica (selección manual, replicas fijas)
COD_ESTA: Código estado (activa, inactiva)

GT_TSESE
Sección sección

Grupo Telemedicina, tabla secciones de sección.

COD_SESE: Código sección sección (Clave primaria)
COD_SPAD: Código sección padre.
COD_SHIJ: Código sección hija.
NUM_ORDE: Número de orden (posición)

GT_TCOMP
Componente

Grupo Telemedicina, tabla componentes

COD_COMP: Código de componente
NOM_COMP: Nombre de componente
VAL_COMP: Valor actual del componente
COD_TIPO: Tipo de componente
COD_MODI: Código de modificaciones
COD_ESTA: Código de estado (activo, inactivo)

GT_TPLAN
Plantilla

Grupo Telemedicina, tabla plantillas

COD_PLAN: Código plantilla (Clave primaria)
NOM_PLAN: Nombre de Plantilla
COD_NVIS: Código nombre visible (visible, oculto)
COD_OVIS: Código observación visible (visible, oculta)
COD_TIPO: Código tipo (S. comp, S padre, S. secc)
COD_DIST: Código distribución (Vert, Horiz)

GT_TSECO
Componente sección

Grupo Telemedicina, tabla componentes de sección.

COD_SECO: Código sección componente (Clave primaria)
COD_SECC: Código de sección
COD_COMP: Código de componente
NUM_ORDE: Número de orden (posición)

GT_TPACO
Componente paciente

Grupo Telemedicina, tabla componentes del paciente

COD_PACO: Código paciente componente (Clave primaria)
COD_PACI: Código de paciente
COD_COMP: Código de componente
VAL_COMP: Valor actual del componente.
COD_MODI: Código de modificaciones

GT_TMODI
Modificaciones

Grupo Telemedicina, tabla modificaciones de componentes

COD_MODI: Código modificación (Clave primaria)
COD_COSE: Código de componente o sección
COD_USUA: Código de usuario
FEC_MODI: Fecha modificación
HOR_MODI: Hora modificación
VAL_ORIG: Valor original
OBS_MODI: Observaciones

GT_TPASE
Sección paciente

Grupo Telemedicina, tabla secciones del paciente

COD_PASE: Código paciente sección (Clave primaria)
COD_PACI: Código de paciente
COD_SECC: Código de sección
COD_MODI: Código modificación.

GT_TUSES
Establecimiento usuario

Grupo Telemedicina, tabla establecimientos usuario.

COD_USES: Código usuario establecimiento (Clave primaria)
COD_USUA: Código usuario
COD_ESTA: Código establecimiento

GT_TUSUA
Usuario

Grupo Telemedicina, tabla usuario

COD_USUA: Código usuario (Clave primaria, Email)
PASSWORD: Password de acceso
COD_MEDI: Código de médico
COD_PERF: Código de perfil
COD_PROF: Código de profesión
COD_SERV: Código de servicio (Ej: Obstetricia)
COD_ESTA: Código estado de cuenta (activa, inactiva, pendiente)
COD_COMU: Código de comuna
FONO: Fono

GT_TPERF
Perfil

Grupo Telemedicina, tabla perfiles

COD_PERF: Código perfil (Clave primaria)
NOM_PERF: Nombre del perfil
COD_PEPR: Código perfil privilegio

GT_TPEPR
Perfil privilegio

Grupo Telemedicina, tabla perfil privilegio

COD_PEPR: Código perfil privilegio (Clave primaria)
COD_PRIV: Código de privilegio

GT_TPRIV
Privilegios

Grupo Telemedicina, tabla modificaciones de componentes

7.3. Detalles de Codificación y Programación

En el siguiente capítulo, se mencionarán algunos aspectos importantes del proceso de programación, intentando aclarar dudas y estableciendo una guía para la manipulación del código del programa, útil futuras modificaciones o actualizaciones.

7.3.1 Funciones y procedimientos

A continuación se listarán las funciones y procedimientos más relevantes dentro del sistema, separándolas en 2 grupos: las que realizan llamadas a la base de datos y las que trabajan con texto.

Funciones de base de datos

1. agregarUsuario(\$rut, \$nombres, \$apepat, \$apemat, \$cod_esp, \$email, \$fono, \$direcci, \$comuna, \$passwd, \$tipo, \$estado)

Función que almacena en la base de datos la información del usuario que se registra en el Sistema. Retorna true si el usuario se almacena correctamente, o false en caso contrario.

2. functionEstaUsuario(\$rut)

Función que realiza consulta en la base de datos para verificar si el rut ya ha sido registrado. Retorna true si el usuario ya se encuentra registrado en el Sistema, o false en caso contrario.

3. obtieneDato(\$col_busc, \$nom_tabl, \$col_actu, \$valor)

Función genérica de base de datos, que realiza consulta de selección sobre una tabla ingresada como parámetro en base a un criterio de búsqueda. Retorna dato buscado, o false en caso de error.

4. `obtieneDato2($col_busc, $nom_tabl, $columna1, $valor1, $columna2, $valor2)`

Función genérica de base de datos, que realiza consulta de selección sobre una tabla ingresada como parámetro en base a dos criterios de búsqueda. Retorna dato buscado, o false en caso de error.

5. `obtieneDato3($col_busc, $nom_tabl, $columna1, $valor1, $columna2, $valor2, $columna3, $valor3)`. Función genérica de base de datos, que realiza consulta de selección sobre una tabla ingresada como parámetro en base a tres criterios de búsqueda. Retorna dato buscado, o false en caso de error.

6. `obtieneMayor($col_busc, $nom_tabl)`

Función que busca valor máximo de un registro ingresado como parámetro. Retorna valor mayor, o false en caso de error.

7. `tipoUsuario($cod_usua)`

Función que determina el tipo de cuenta de usuario (perfil), según su identificador único. Retorna el tipo de usuario en base a sus privilegios.

8. `estadoCuenta($cod_usua)`

Función que determina el estado de una cuenta de usuario en base a su identificador único. Retorna el estado de la cuenta de usuario: (-1) inactiva, (1) activa, (0) pendiente

9. `numDatos($col_busc, $nom_tabl, $col_actu, $valor)`

Función que obtiene el número total de registros de una tabla pasada como parámetro, en base a un criterio de búsqueda. Retorna número de registros, o cero en caso de error.

10. `numDatos2($columna_buscada, $nombre_tabla, $columna1, $valor1, $columna2, $valor2)`

Función que obtiene el número total de registros de una tabla pasada como parámetro, en base a dos criterios de búsqueda. Retorna número de registros, o cero en caso de error.

11. `obtieneColumna($col_busc, $nom_tabl, $col_actual, $valor, $col_ord, $orden)`

Función que retorna vector de registros en base a un criterio de búsqueda. Retorna vector, o false en caso de error.

12. `obtieneColumna2($col_busc, $nom_tabl, $columna1, $valor1, $columna2, $valor2, $columna_orden, $orden)`

Función que retorna vector de registros en base a dos criterios de búsqueda. Retorna vector, o false en caso de error.

13. `obtieneColumna3($col_busc, $nom_tabl, $columna1, $valor1, $columna2, $valor2, $columna3, $valor3, $col_orde, $orden)`

Función que retorna vector de registros en base a tres criterios de búsqueda. Retorna vector, o false en caso de error.

14. `bdQueryUDIS($query)`

Función de base de datos para realizar query update, delete, insert o select

Retorna arreglo bidimensional con resultado de la consulta SQL de selección entregada como parámetro, true en caso de que se trate de una consulta de inserción, modificación o eliminación, y false en cualquier otro caso.

15. `verContenidoSeccion($cod_spad)`

Función recursiva que genera tablas HTML con contenido ordenado de la sección padre, cuyo código único es pasado como parámetro. La estructura gráfica es determinada por su plantilla, la cual define orientación, forma, visibilidad de propiedades, etc. Así mismo, permite generar el listado de componentes de las secciones que corresponda, y de esta forma lograr un formulario listo para pasar a producción, tarea que se realiza asociando la sección a un episodio.

Funciones de texto

1. `caracterAceptado($c) {`

Retorna true si el carácter ingresado como parámetro es alfanumérico, y false en caso contrario. Se usa para la validación y generación de contraseñas de acceso o claves para registros de las tablas de base de datos.

2. `generaCodigo($entrada)`

Retorna contraseña o clave a partir de una frase. Quita caracteres inválidos y trunca a un largo de diez.

3. dv(\$r)

Retorna dígito verificador a partir del RUT

4. rutValido(\$rut)

Retorna true en caso de que el RUT ingresado sea válido, o false en caso contrario.

5. isMail(\$mail)

Retorna true si la dirección de correo electrónico es válida, o false en caso contrario.

7.3.2 Estructura del Sistema

El Sistema se programó en módulos independientes, lo cuales se encuentran organizados en un árbol de directorios estructurado de tal forma que cada uno de los componentes pueda ser fácilmente identificado y modificado por cualquier programador con conocimientos de lenguaje PHP y HTML.

En la raíz se encuentran los siguientes archivos y directorios:

index.php: Página principal del Sistema. Despliega página de Login o Noticias de la especialidad, según corresponda.

config.inc.php: Archivo de configuración del Sistema. Datos de conexión a la base de datos, número de elementos a desplegar por página, rutas hacia repositorios de documentos e imágenes, nombre de etiquetas principales de las fichas clínicas (episodios, exámenes), etc.

header.inc.php: Panel superior con menú principal, común a todos los módulos del Sistema en el nivel cero (raíz)

header_sec.inc.php: Panel superior con menú principal, común a todos los módulos del Sistema en el nivel uno (directorios secundarios)

footer.inc.php: Panel inferior con menú secundario, ayuda y buscador general del Sistema en el nivel cero.

footer_sec.inc.php: Panel inferior con menú secundario, ayuda y buscador general del Sistema en el nivel uno.

imágenes: Directorio en donde se almacenan las imágenes usadas en las diferentes interfaces de los módulos del Sistema.

admin: Directorio que contiene módulo de administración de usuarios, fichas clínicas, secciones, componentes, plantillas, etc. Está restringido a los usuarios con el nivel de privilegios suficientes según su perfil.

público: Directorio que contiene todos los módulos a ser accedidos por los usuarios del Sistema, fichas clínicas, noticias, interconsultas, foros, chats, etc.

include: Directorio que contiene todos los scripts comunes (librerías) a los módulos del Sistema.

css: Directorio que contiene archivos con hojas de estilo del Sistema.

seguridad: Directorio que contiene scripts para validación de cuentas de usuario y generación de variables de sesión, con acceso a los distintos módulos del Sistema sobre los cuales tiene privilegios en base a su perfil.

funciones: Directorio que contiene funciones comunes a todos los módulos del Sistema, incluidas aquellas para conexión/desconexión de base de datos y consultas.

comunes: Directorio que contiene archivos comunes a los módulos del Sistema.

7.3.3 Reglas generales de programación y diseño

- Todos los nombres de funciones con más de una palabra se escriben juntos, en minúscula y desde la segunda empiezan con la primera letra en mayúscula.
- Todos los pasos de variable entre páginas PHP por métodos POST o GET se realizan utilizando nombres de variables con largo de ocho caracteres, abreviadas en el caso de ser de más de una palabra, y concatenadas por “_”, guión bajo.

- Todas las operaciones de acceso a base de datos, ya sea consultas de selección, inserción, modificación o eliminación, fueron pensadas para una eventual migración de base de datos. Esto por tratarse de un Sistema genérico que pueda adaptarse fácilmente a las necesidades de cualquier entidad hospitalaria. Para ello, cada consulta se ejecuta en lenguaje SQL nativo haciendo uso de las funciones creadas para tal efecto.
- Las interfaces de usuario se realizaron utilizando normas de diseño para la adecuada ubicación de los componentes y facilidad de uso.
- Los colores de las interfaces fueron pensados para ser atractivos a la vista y facilitar la accesibilidad.

7.3.4 Concepto de Secciones, Componentes y plantilla de exámenes

Una de las principales ventajas del Sistema es la flexibilidad ofrecida para el diseño de fichas clínicas personalizadas haciendo uso de las herramientas de administración implementadas, sin ser necesaria la intervención en el código fuente. Ésta característica permite que cualquier profesional de la salud o administrador con los privilegios suficientes pueda crear plantillas de exámenes en pocos minutos si cuenta con un diseño bien definido.

Las plantillas de exámenes corresponden a un conjunto estructurado de secciones y componentes HTML, y las consideraciones en el momento de crearlas tienen una estrecha relación con la estructura de la base de datos y las tablas involucradas. Tal es el caso de la tabla GT_TPLAN, que define las características principales de las secciones:

COD_NVIS: Código que indica si el nombre de sección debe o no ser mostrado en las plantillas de exámenes.

COD_OVIS: Código que indica si la observación debe o no ser mostrada en las plantillas de exámenes.

COD_TIPO: Código que indica el tipo de sección.

S. padre: Sección padre. En el Sistema representa a los episodios.

S. secc: Sección que contiene solamente secciones. Un ejemplo de este tipo son los exámenes.

S. comp: Sección que contiene solamente componentes HTML (text, textarea, image, select, etc)

COD_DIST: Código que si los componentes o secciones pertenecientes se ordenarán de manera vertical u horizontal.

Además de las propiedades definidas en la plantilla de la sección, ésta cuenta con otras opciones propias, que permiten una mayor versatilidad y reutilización de los elementos creados. Tal es el caso de las opciones:

NUM_REPL: Número de réplicas de la sección (1-50). Permite crear una sola sección que se replica de manera automática en los formularios. Un ejemplo de ello en el caso de Obstetricia es la sección Feto, la cual podría eventualmente tener más de una ocurrencia (sección), pero con las mismas características.

VAL_NRED: Valor por defecto del número de réplicas de la sección. Corresponde al número de repeticiones que debiera generarse en el formulario que incluya a dicha sección (sin intervención del usuario).

COD_TREP: Código tipo de replica (selección manual, replicas fijas). Si la opción es “replicas fijas”, la sección desplegará todas las replicas que se haya definido en NUM_REPL, sin embargo, cuando tiene la opción “selección manual”, no se mostrará ninguna sección en el formulario que la incluya. En lugar de ello aparecerá un componente HTML de tipo SELECT, que permitirá seleccionar desde uno a NUM_REPL, y de acuerdo a ese valor se genera las replicas de las secciones en el formulario.

COD_ESTA: Código estado (activa, inactiva). Se utiliza para separar los ambientes de desarrollo y producción, al permitir incorporar nuevas secciones a un examen sin afectar el funcionamiento normal del Sistema. Por defecto, cualquier sección se crea en modo inactiva, y es el médico responsable de la especialidad el encargado

de autorizar el paso a producción de una sección determinada en una plantilla de examen.

La estructura en bloques de una plantilla de examen se puede ver en el siguiente esquema.

Plantilla de examen

a) Sección nivel 0 (tipo s.secc)			
b) Sección nivel 1 (tipo s.secc)			
		c) Sección nivel 2 (tipo s.comp)	
		Componente 1	
		d) Sección nivel 2 (tipo s.comp)	
		Componente 1	
		e) Sección nivel 2 (tipo s.secc)	
			f) Sección nivel 3 (tipo s.secc)
			g) Sección nivel 4 (tipo s.comp)
		Componente 1.	
		h) Sección nivel 3 (tipo s.comp)	
		Componente 1.	

7.4. Módulo interconsultas

Una interconsulta, se define como la solicitud que un médico perteneciente a una especialidad en particular, realiza a otra área de la medicina. Esto se aplica cuando la afección que sufre su paciente está por fuera de sus competencias profesionales.

En el Sistema implementado se consideró la programación de un módulo capaz de gestionar, tanto la solicitud, como la atención a las interconsultas. Para ello se definió ocho estados posibles aplicables a ambos casos, y que pueden ser administrados desde el menú principal.

- Creada
- Pendiente
- Aceptada
- Delegada
- Atendida

- Rechazada
- Terminada
- Reasignada

Junto con definir los estados de las interconsultas, se hace necesario disponer de un modelo de datos que soporte la información relevante, para entenderlo mejor, se cuenta con el diagrama de entidad – relación de la figura 7.6.

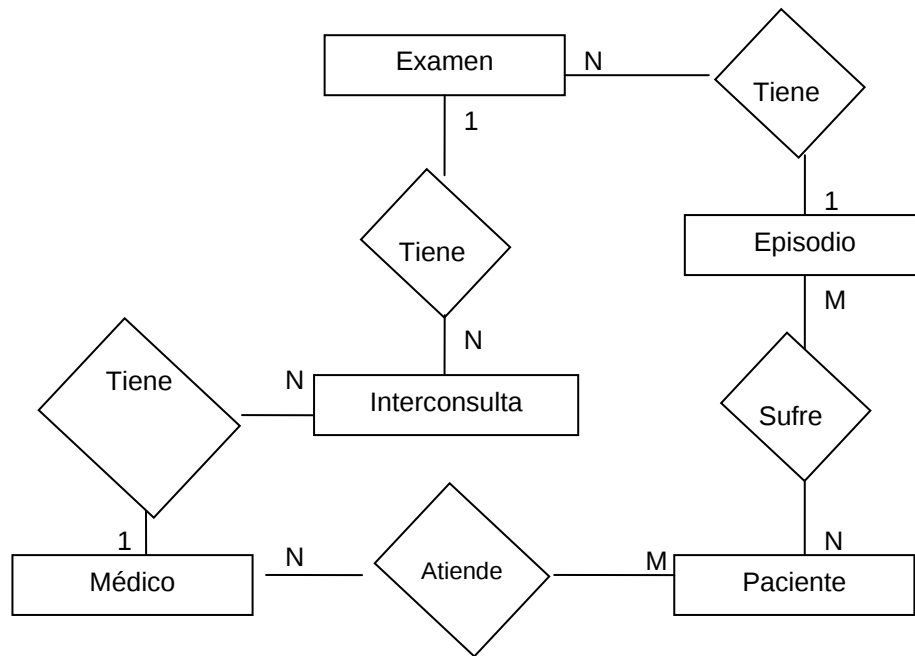


Figura 7.6: Diagrama entidad-relación del módulo interconsultas

Explicación:

- Uno o más exámenes, pertenecientes a un episodio sufrido por un paciente, tiene asociada una o más interconsultas.
- Un médico atiende a varios pacientes, por los cuales es necesario solicitar una o más interconsultas cuando corresponda.

7.4.1 Modelo de datos

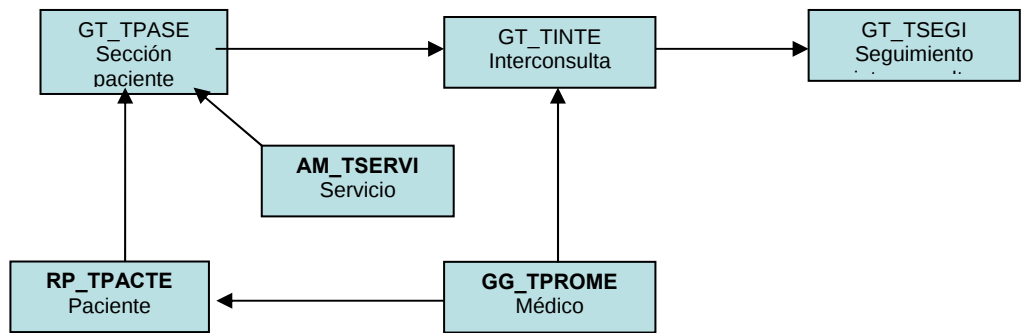


Figura 7.7: Modelo de datos del módulo interconsultas

7.4.1.1 Descripción de las tablas y sus campos

GT_TINTE Interconsulta	Grupo telemedicina, tabla de interconsultas. COD_INTE: Código interconsulta (Clave primaria) NUM_CORPAC: Número de paciente COD_SEGI: Código seguimiento interconsulta
GT_TSEGI Seguimiento interconsulta	Grupo telemedicina, tabla de seguimiento interconsultas. COD_SEGI: Código seguimiento interconsulta (Clave primaria) COD_INTE: Código interconsulta FEC_SEGI: Fecha seguimiento COD_RUTPRO: Rut del médico, sin dígito verificador COD_SERVIC: Código de servicio (especialidad) COD_ESTA: Código estado NOM_ASUN: Asunto de la interconsulta OBS_SEGI: Observaciones seguimiento

7.4.1.2. Estructura y relaciones entre tablas

Una interconsulta en el momento que es solicitada por un médico, se almacena en la tabla GT_TINTE solamente con el número de paciente (NUM_CORPAC) y los códigos de interconsulta y seguimiento generados automáticamente.

En una primera etapa, luego de almacenarse los datos anteriores, se crea una entrada en la tabla GT_TSEGI, en donde se especifica: los códigos de seguimiento e interconsulta (los mismos anteriores), el rut del médico solicitante (COD_RUTPRO) la fecha y hora de la solicitud (FEC_SEGI), la especialidad a la cual será enviada (COD_SERVIC), el estado (COD_ESTA), el asunto (NOM_ASUN), y las observaciones (OBS_SEGI).

7.4.2 Diagramas de Flujo de Datos

7.4.2.1. Diagrama nivel 0:

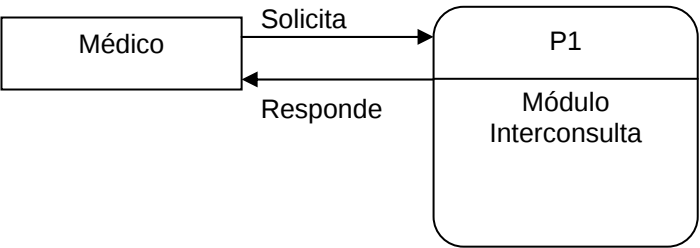


Figura 7.8: Diagrama nivel 0.

En la figura 7.8, el médico solicita una interconsulta, y recibe una respuesta del Sistema.

7.4.2.2. Diagramas de nivel 1

Módulo interconsulta

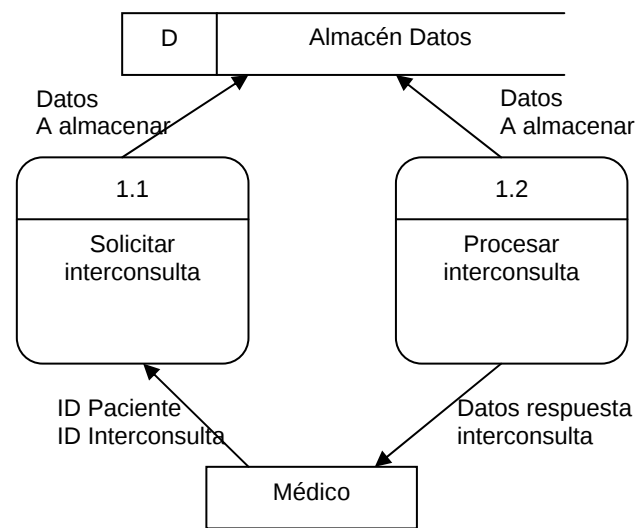


Figura 7.9: Diagrama nivel 1, módulo interconsulta

El médico envía los datos del paciente, junto con la especialidad a la cual solicita la interconsulta, como se aprecia en la figura 7.9.

El responsable de la especialidad procesa los datos enviados, y responde a la interconsulta (acepta o rechaza).

7.4.2.3. Diagramas de nivel 2

Módulo Solicitar interconsulta

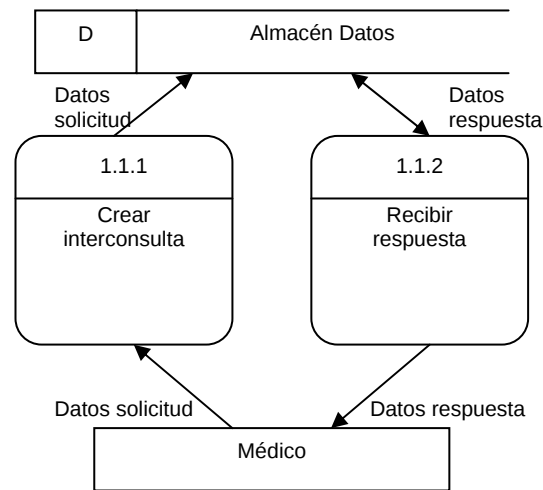


Figura 7.10: Diagrama nivel 2, solicitud de consulta

La figura 7.10 muestra que el médico crea una interconsulta, que es almacenada en la base de datos, y la asigna a alguna especialidad.

El médico responsable de la especialidad, lee los datos de la interconsulta y la acepta o rechaza según corresponda.

Módulo Procesar interconsulta

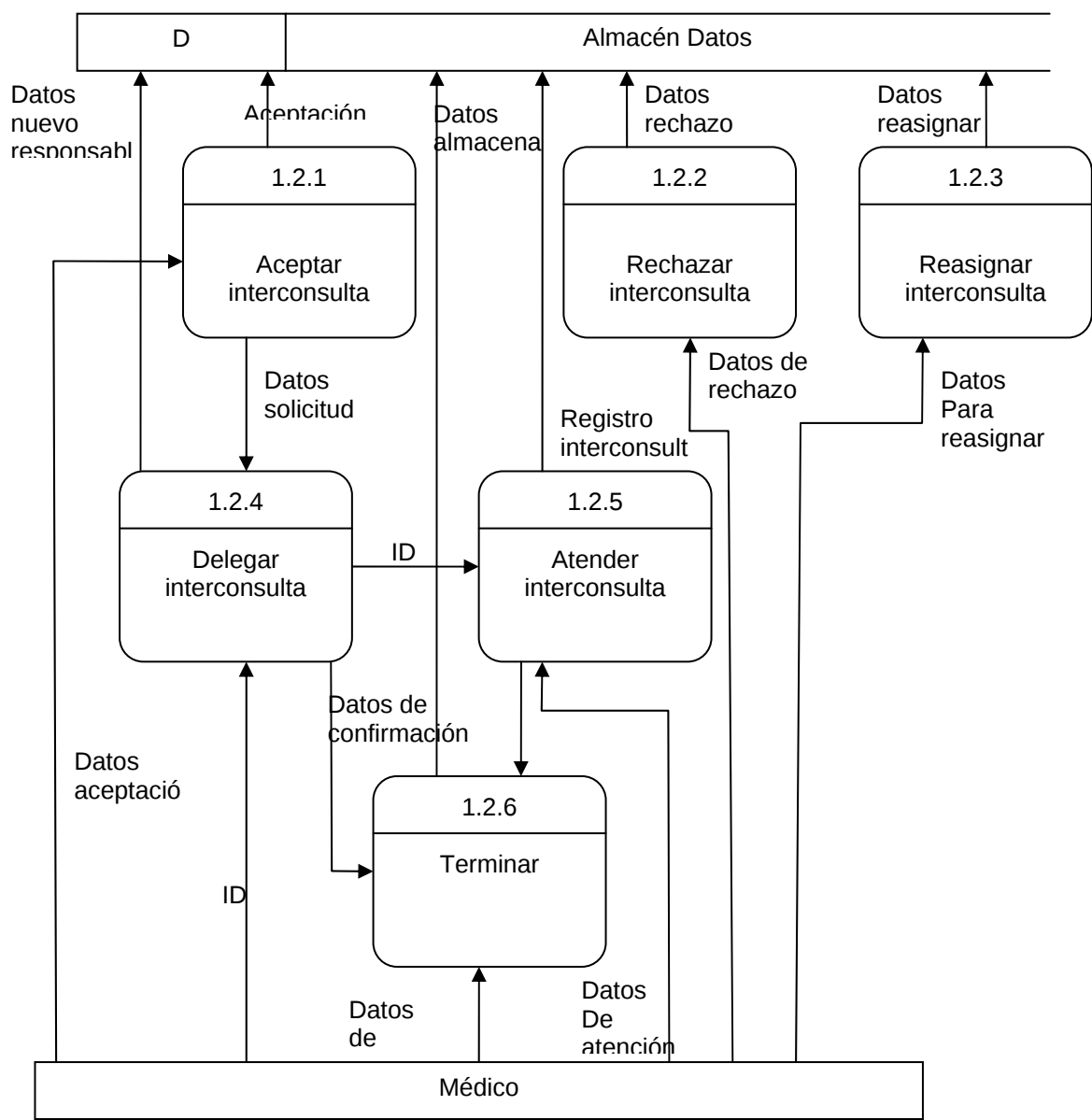


Figura 7.11: Diagrama nivel 2, procesar interconsulta

De acuerdo a la figura 7.11, el médico responsable de la especialidad, a la que se ha solicitado una interconsulta, lee los datos y decide si la acepta o rechaza.

En caso de que acepte, puede delegarla a un colega de la misma especialidad, o atenderla él mismo.

En caso de delegarla, el médico asignado puede atenderla, terminarla o rechazarla. Nuevamente es el médico responsable de la especialidad quien decide a quien delegarla, si el médico que asignó la rechaza.

En el caso de que el médico responsable haya rechazado la interconsulta en primera instancia, entonces le llega una notificación al médico solicitante indicándole las causas del rechazo. De esa forma él puede tomar la decisión de reasignarla a otra especialidad o terminarla.

Cuando un médico decide terminar la interconsulta, es porque ya ha realizado los exámenes o análisis que corresponda para dar respuesta a la interconsulta.

7.5. Guía de operación

a. Acceso, registro y validación de usuarios.

Para ingresar un usuario al sistema, debe hacerlo por medio de un Browser o navegador de Internet, figura 7.12. En la barra de direcciones debe anotar la dirección donde esté alojada la página (1), posteriormente debe escribir su Rut, Contraseña y presionar el botón ingresar (2).

En caso de que el usuario no se encuentre registrado en el sistema, debe hacerlo por medio del botón registrar en la barra de opciones (3).

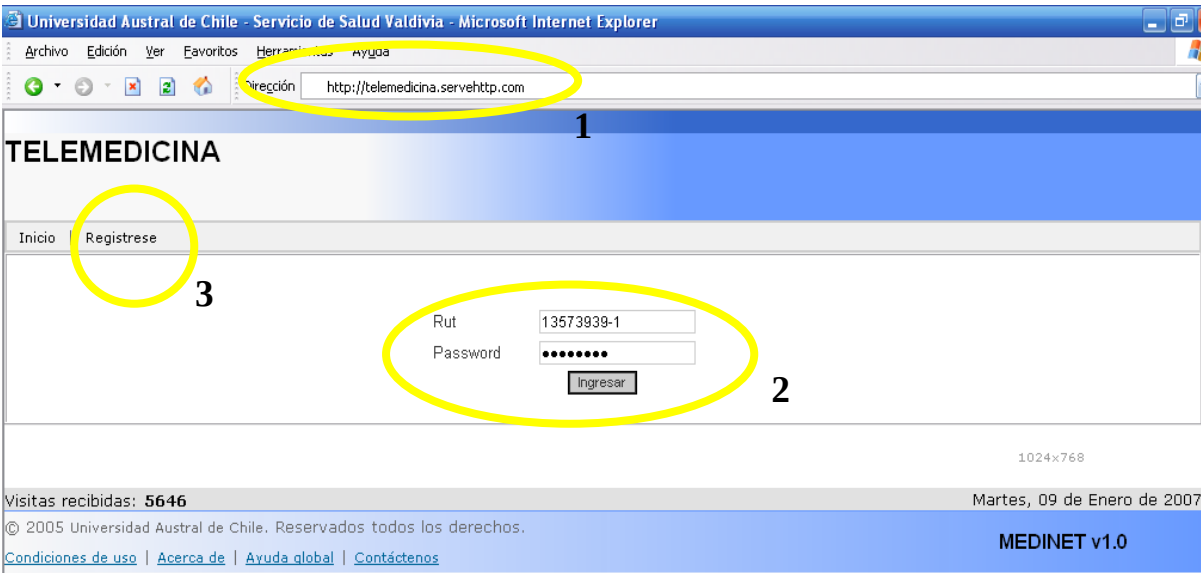


Figura 7.12: Pantalla principal de acceso al sistema telemédicina.

Para realizar la opción de registro, una vez presionada la opción de registrarse (3), se desplegará la pantalla correspondiente a la figura 7.13:

TELEMEDICINA

Inicio | Regístrese

Registrar nuevo usuario

Nombre especialidad: -- Seleccione especialidad --

Rut: 15549240-6

Nombres: Eduardo

Apellido paterno:

Apellido materno:

E-Mail:

Fono:

Dirección:

Comuna: ---- Seleccione comuna ----

Password:

Repetir Password:

Registrar usuario

1024x768

Visitas recibidas: **5646**

© 2005 Universidad Austral de Chile. Reservados todos los derechos.

Miércoles, 10 de Enero de 2007

[Condiciones de uso](#) | [Acerca de](#) | [Ayuda global](#) | [Contáctenos](#)

MEDINET v1.0

Figura 7.13: Pantalla registro de usuarios.

En la sección 4 deberán completarse todos los datos requeridos, y para finalizar presionar el botón registrar usuario.

Al completar ese paso, el usuario recibe una notificación por correo, confirmando que su cuenta se encuentra en proceso de validación, siendo el administrador de especialidad o administrador del sistema el encargado de realizar dicha labor. Cuando esto ocurre, el sistema notifica nuevamente vía correo electrónico, que su cuenta ha sido habilitada. Cuando ingresa al sistema, lo primero que se despliega es la ventana de la figura 7.14, que corresponde al módulo noticias de su especialidad (por ejemplo obstetricia):



Figura 7.14: Pantalla principal de acceso a la especialidad, módulo noticias.

Cada ventana está dividida en 3 secciones: sección a, que contiene la cabecera, en la cual se despliega la información del usuario activo (5) y los módulos activos según perfil (6: fichas clínicas, foro, chat, etc.); la sección b, cuerpo, que contiene la ventana del módulo activo (por ejemplo noticias); finalmente la sección c, pié de página, que despliega funcionalidades extra, tales como: buscador general, ayuda o preferencias. También se despliega información de contacto, propietario del sitio y desarrolladores (9).

El módulo noticias entrega 2 secciones: la derecha, que destaca una noticia relevante (8); y la izquierda, que presenta la lista con las últimas noticias agregadas a la especialidad del usuario activo (7). En cada una de ellas se puede ampliar la información por medio del enlace destacado.

Al ingresar al módulo Fichas Clínicas se despliega la ventana de la figura 7.15:

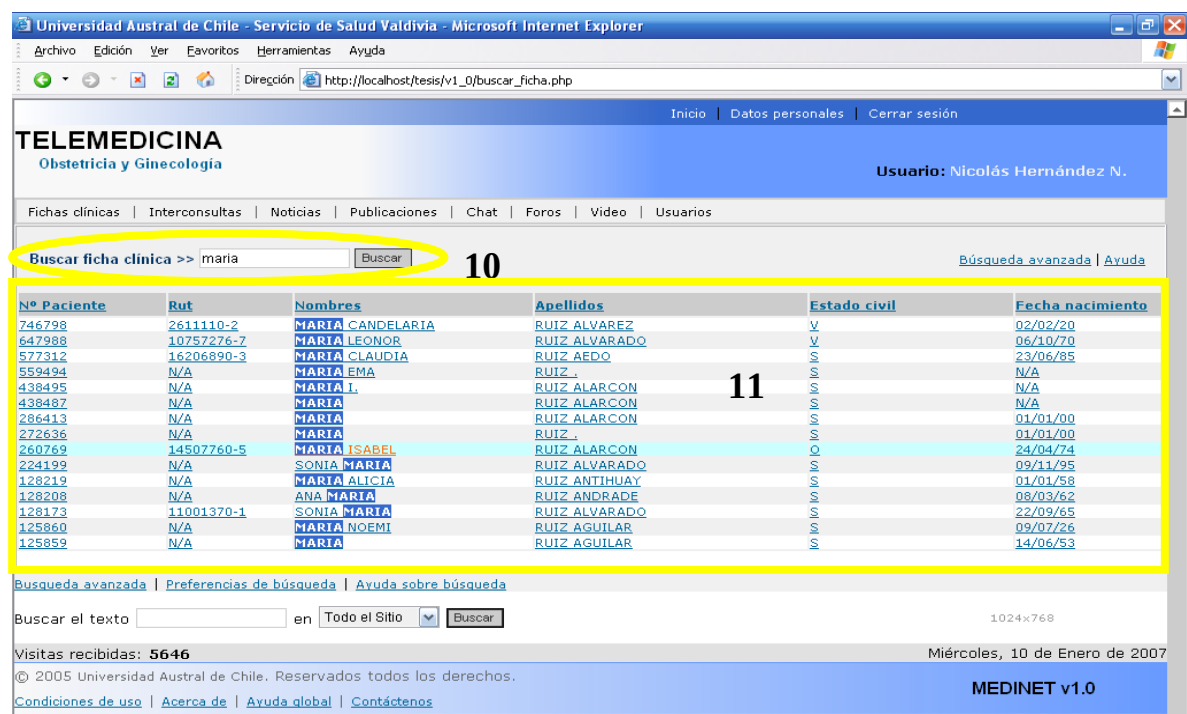


Figura 7.15: Pantalla módulo de fichas clínicas, buscador de pacientes.

El buscador (10) acepta como entrada cualquier texto o número, lo que posibilita encontrar pacientes bajo cualquier combinación de caracteres, por ejemplo, se puede ingresar el nombre, rut, apellido, nº de paciente o partes de ellos. El sistema despliega una lista con los pacientes y sus datos (11), el usuario escoge el paciente correcto y se despliega la pantalla con la información del paciente.

La ventana de la figura 7.16, con información del paciente contiene 3 secciones principales: identificación del paciente y datos de la ficha (12), que se mantiene en todo el módulo; enlaces a historial clínico y ficha (13); datos del paciente en revisión (14).

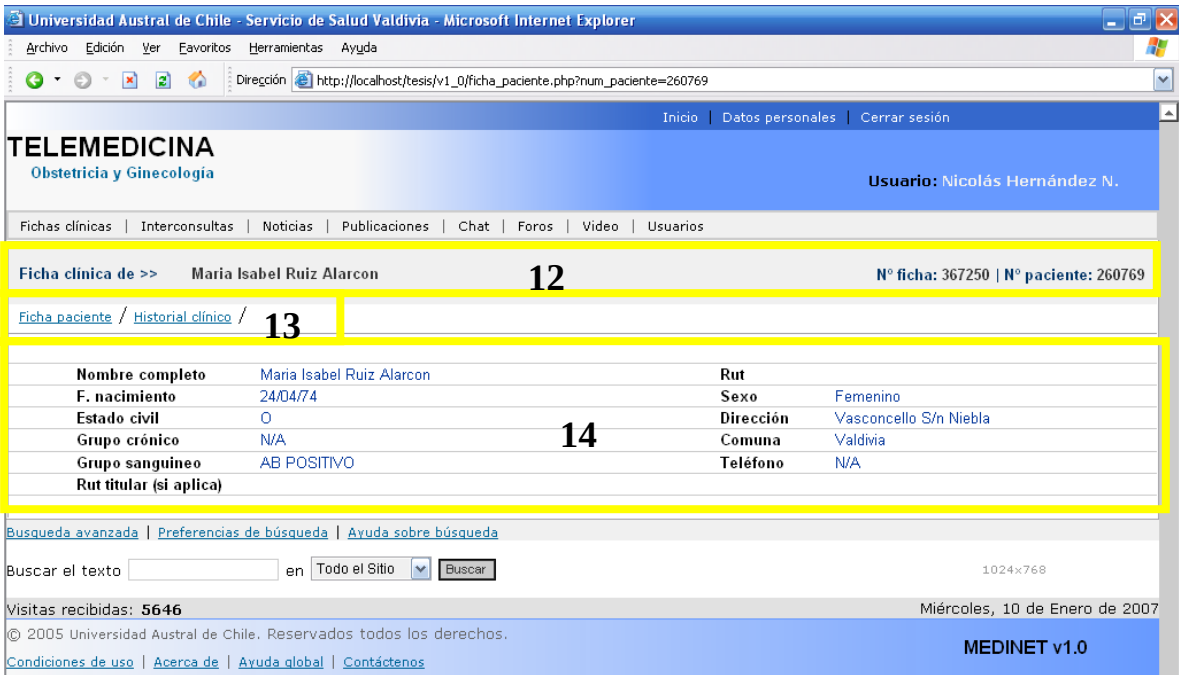


Figura 7.16: Pantalla módulo de fichas clínicas, datos del paciente.

Al enlazar el historial clínico, se despliega la siguiente ventana:

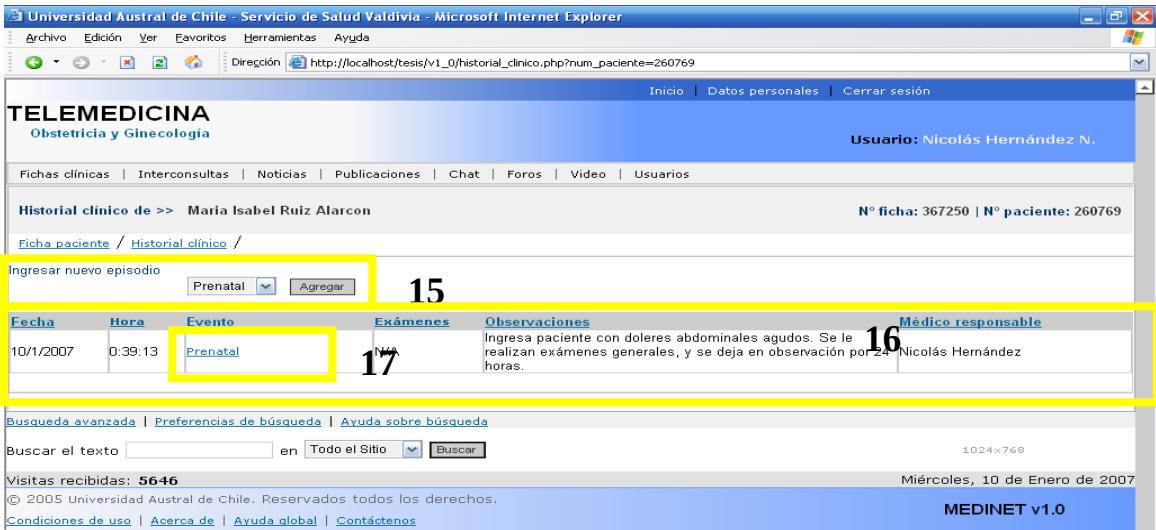


Figura 7.17: Pantalla módulo de fichas clínicas, historial del paciente.

La figura 7.17 muestra la lista de los episodios que ha tenido el paciente en la especialidad del médico (16), la opción de agregar un nuevo episodio (15) o modificar uno existente (17).

Si se desea agregar un nuevo episodio, se escoge el tipo (según los disponibles; 15), se desplegará lo siguiente:

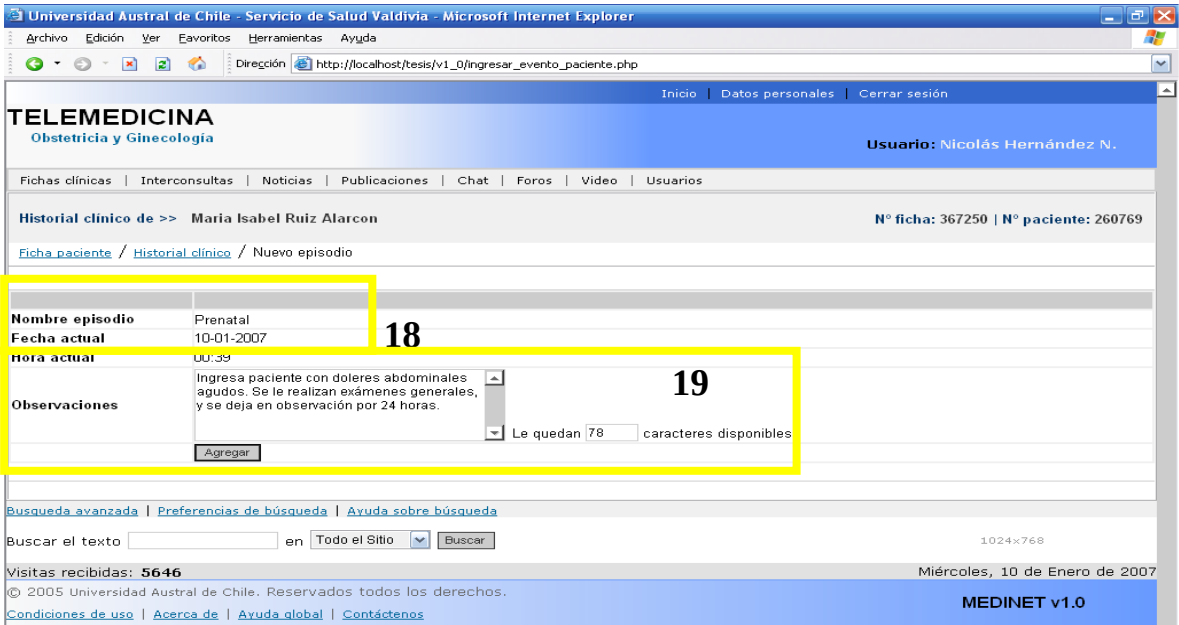


Figura 7.18: Pantalla módulo de fichas clínicas, agregar episodio.

La figura 7.18 muestra que algunos parámetros son establecidos por defecto (18), mientras que las observaciones deben ser ingresadas manualmente(19).

Para agregar un examen, debe seleccionarse un formato existente (20), como lo demuestra la figura 7.19.

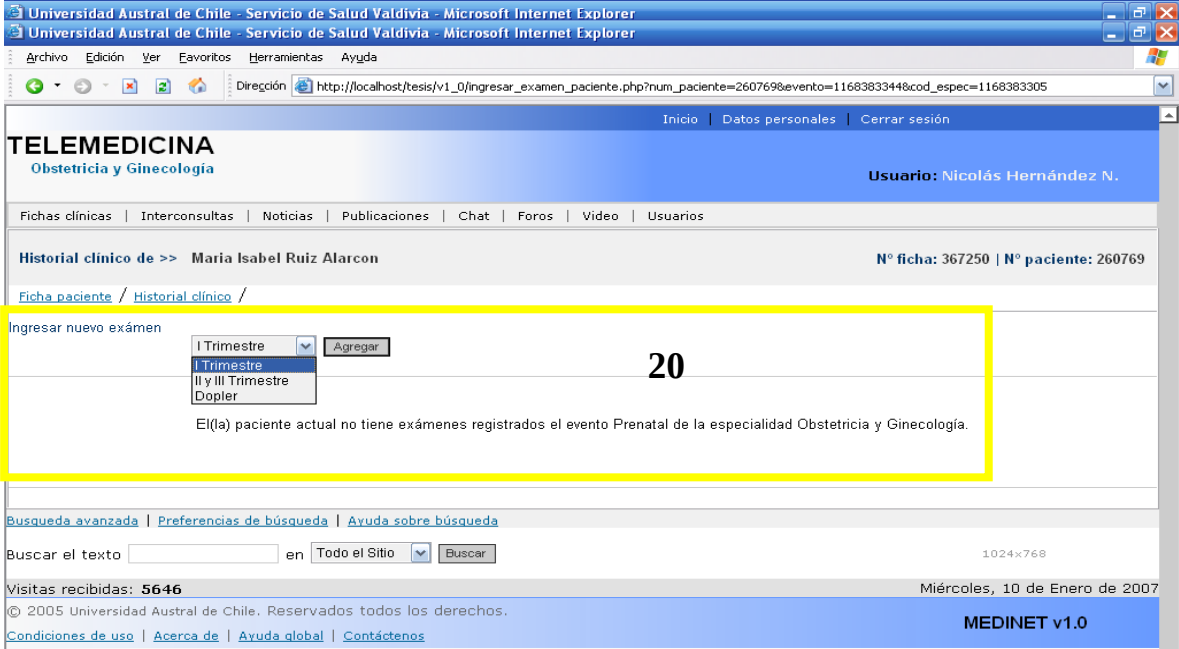


Figura 7.19: Pantalla módulo de fichas clínicas, agregar examen.

Como se aprecia en la figura 7.20, al agregar un examen, este se despliega para ser llenado por el médico y almacenado en la base de datos central (21).

Unidad de Salud - Servicio de Salud Valdivia - Microsoft Internet Explorer

Inicio | Datos personales | Cerrar sesión

TELEMEDICINA
Obstetricia y Ginecología

Usuario: Nicolás Hernández N.

Fichas clínicas | Interconsultas | Noticias | Publicaciones | Chat | Foros | Video | Usuarios

Historial clínico de >> Maria Isabel Ruiz Alarcon N° ficha: 367250 | N° paciente: 260769

Ficha paciente / Historial clínico / Nuevo examen

Nombre examen	I Trimestre
Fecha actual	10-01-2007
Hora actual	00:43

1.- Utero

Posición

Dimensiones Largo: mm Ant-post: mm Trans: mm

Miometrio

Saco Gestacional x x Promedio

Embriones ☐

21

Figura 7.20: Pantalla módulo de fichas clínicas, examen para completar.

En la figura 7.21, se aprecia un nuevo apartado correspondiente al módulo administración, requiere un rut y una contraseña para ingresar (22).

Unidad de Salud - Servicio de Salud Valdivia - Microsoft Internet Explorer

http://localhost/tesis/v3_0/admin/index.php

Administrador

Usuario

Password

Ingresar

22

Figura 7.21: Pantalla módulo de administración, validar usuario.

Una vez dentro, aparece el menú con las opciones de administración (23), según figura 7.22.



Figura 7.22: Pantalla módulo de administración, menú opciones.

Que nos permitirán modificar los diferentes módulos que componen el sistema.

Como se aprecia en la figura 7.23, el menú episodios permite editar o crear episodios (24).

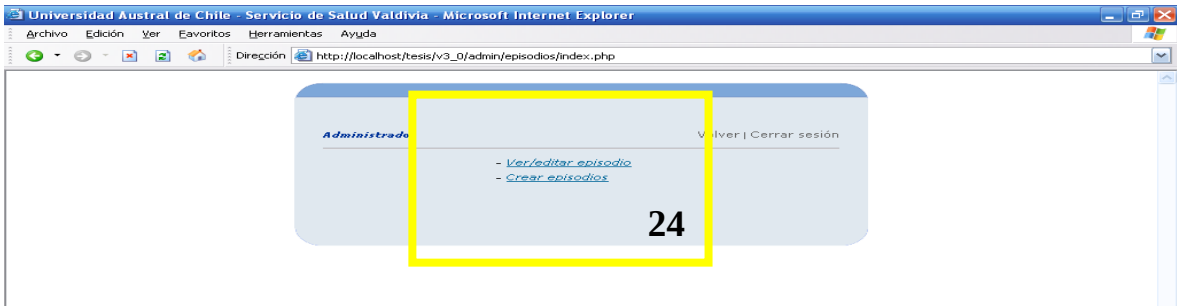


Figura 7.23: Pantalla módulo de administración, episodios.

Para editar, solo basta con modificar los datos que despliega el sistema (25), como indica la figura 7.24.

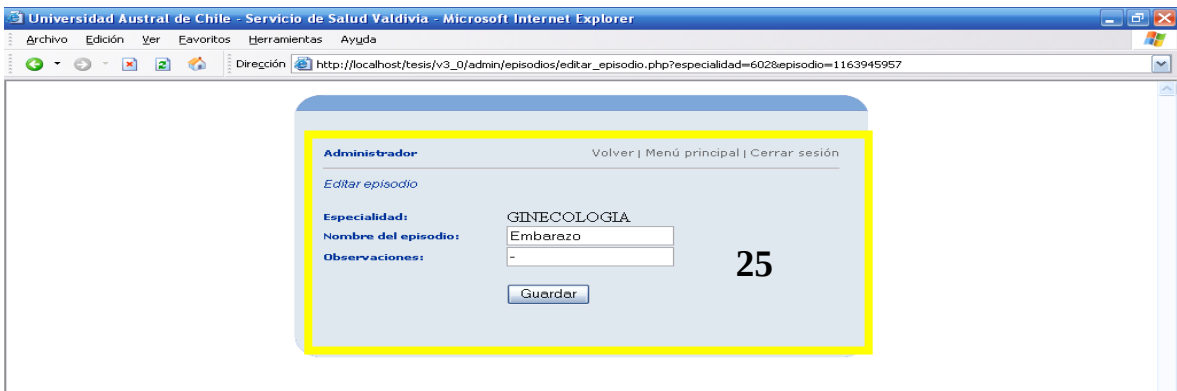


Figura 7.24: Pantalla módulo de administración, editar episodios.

Mientras que para crear hay que llenar cada campo solicitado (26), como en la figura 7.25.

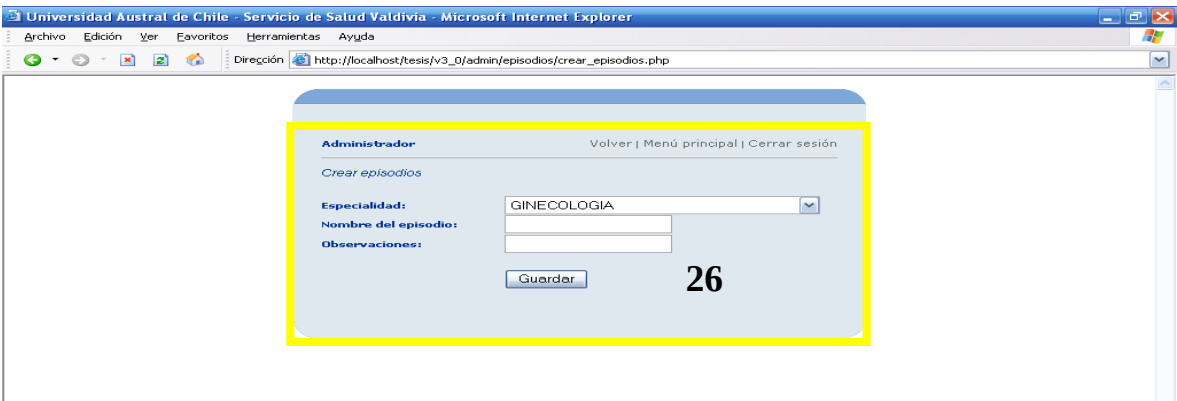


Figura 7.25: Pantalla módulo de administración, crear episodios.

De forma similar se opera con cada uno de lo demás módulos, basta con seguir las indicaciones y administrar la información necesaria.

Es recomendable que para modificar o crear episodios, exámenes, secciones y componentes, se lleve a cabo un acuerdo entre un equipo multidisciplinario, integrado por especialistas del área de la medicina y por los encargados del sistema de información del hospital, puesto que no es una tarea trivial el hecho de incorporar nuevos episodios o exámenes; se debe tener en cuenta el orden, tipos, y nombre de los componentes, definir las secciones y el episodio que las despliega, lo ideal sería que un encargado de área determine el formato de las nuevas fichas, y estas, una vez implementadas, pasen por la aprobación de los usuarios comunes o “marcha blanca”.

Lo anterior se debe a que el sistema fue desarrollado para crecer en el tiempo, así se está cumpliendo con uno de los requisitos más importantes: “Ser un sistema de propósito general en medicina”.

La figura 7.26 muestra que al crear un nuevo examen, debe seleccionarse: la especialidad, el episodio al que pertenece, el nombre del examen y alguna observación (27), por ejemplo: en el área de obstetricia, existe un episodio denominado embarazo, que

tiene un examen llamado III trimestre, el cual contiene la información recabada por el médico al examinar una embarazada con más de 6 meses de gestación.

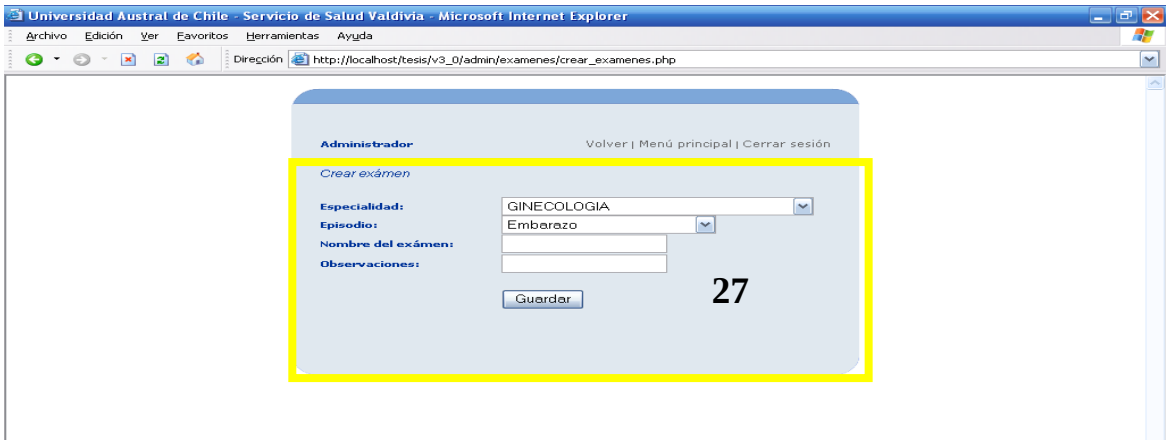


Figura 7.26: Pantalla módulo de administración, crear examen.

Para crear una sección, primero se asigna un nombre, luego se determina si posee alguna sección padre (en caso de ser una subsección), finalmente se indica a qué examen pertenece (28), figura 7.27.

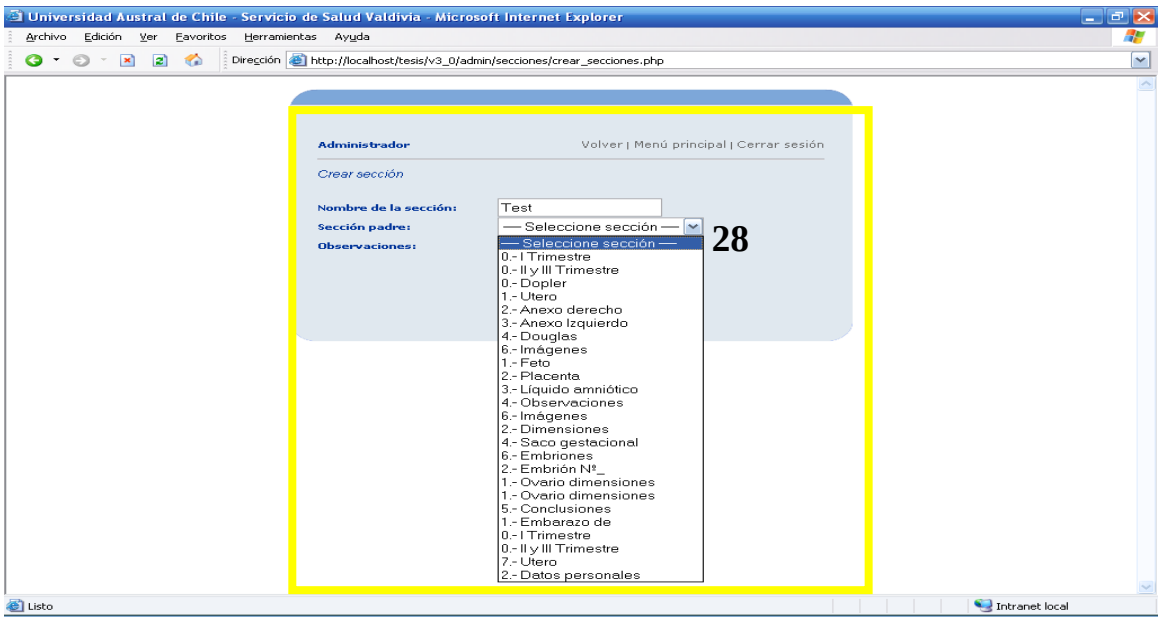


Figura 7.27: Pantalla módulo de administración, crear secciones o subsecciones.

En la figura 7.28, se ve que para crear un componente, hay que nombrarlo, definir el tipo (recordando que corresponde a un componente html) y la sección a la que pertenece (29).

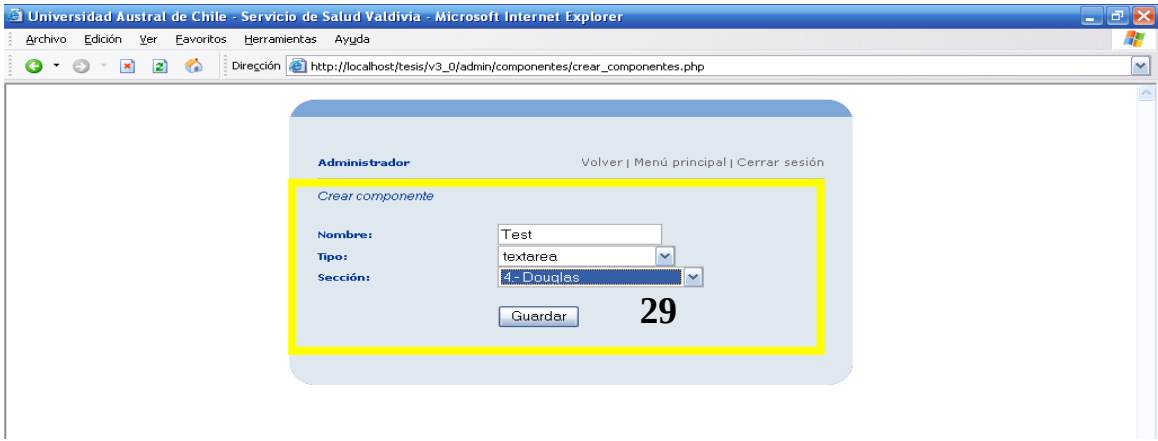


Figura 7.28: Pantalla módulo de administración, crear componentes.

De este modo se logra un esquema similar al desplegado en el anexo 7.4 “Plantilla Examen”, que en el sistema real se muestra en la figura 7.29:

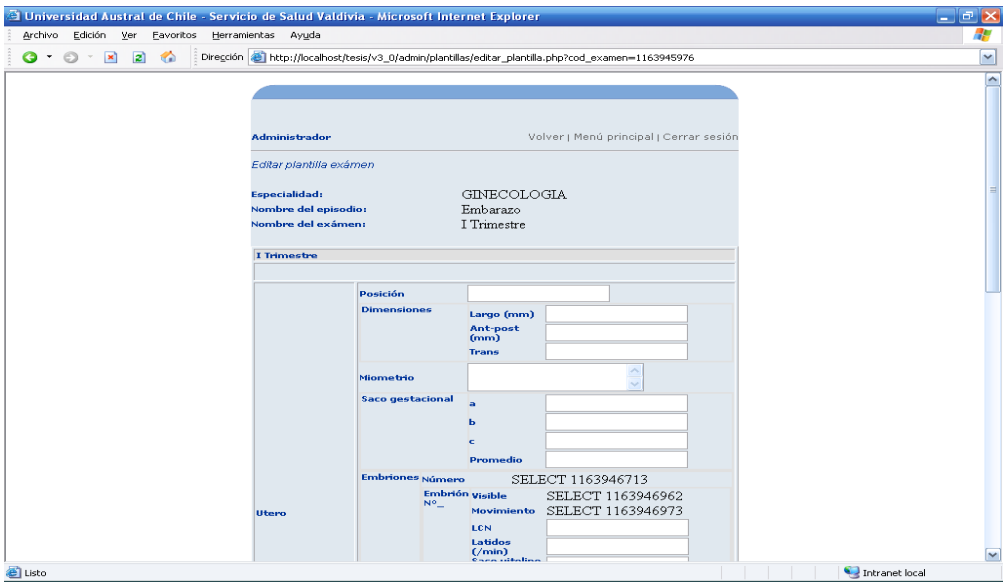


Figura 7.29: Pantalla módulo de administración, plantilla examen.