

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



PROYECTO DE GRADO

“SISTEMA INTEGRADO DE SALUD”

Caso: Gobierno Autónomo Municipal de La Paz

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: Eitner Montero Churata
TUTOR: Lic. Germán Huanca Ticona
REVISOR: Lic. Nancy Orihuela Sequeiros

LA PAZ – BOLIVIA

2011

Dedicatoria

Este trabajo está dedicado a mi querida madre Lic. Miriam Churata Pereira por su constante apoyo para seguir adelante, a quien debo mi respeto, gratitud y admiración y que es un claro ejemplo de superación y valentía que llevo todos los días de mi vida.

Eitner.

AGRADECIMIENTOS

Haber llegado a las instancias de presentar este Proyecto de Grado es el mérito de varias personas que han sido mis mentores, impulsores y sobre todo un gran apoyo para seguir adelante y llegar a la meta, a quienes les debo todo mi agradecimiento.

Ante todo, mis agradecimientos a Dios por brindarme la fuerza y fe para seguir adelante en momentos tan difíciles.

Expresar la más profunda gratitud a mi familia por el apoyo, la paciencia, la comprensión y tolerancia que tuvieron a pesar de estar lejos de casa durante el tiempo de mi formación profesional, y muy especialmente a mi hijo Jorge David, quien frecuentemente me pedía un regalo cada vez que le llamaba por teléfono y me decía “papi, ya me debes 50 autitos”, disculpas por haberme perdido tu rápido crecimiento.

Al Lic. German Huanca Ticona, por su orientación, consejos y por haber dedicado su tiempo en la tutoría de este proyecto.

Un agradecimiento especial a la Lic. Nancy Orihuela Sequeiros, quien fuera mi docente cuando iniciaba mi formación académica, y posteriormente como guía de este proyecto, por su valiosa colaboración y paciencia donde además ha sido una revisora ejemplar.

Al Lic. Yhonny Felipez Andrade, quien me brindo su amistad desinteresada y en condiciones académicas y humanas superiores, me permitió lograr este crecimiento.

Al personal de la Unidad de Alimentación Complementaria Escolar (G.A.M.L.P.), por darme la oportunidad de demostrar mis capacidades como Informático, desarrollando el presente Proyecto de Grado para su institución.

RESUMEN

La Unidad de Alimentación Complementaria Escolar (UNACE) dependiente del Gobierno Autónomo Municipal de La Paz, dedicada a la distribución del desayuno escolar, cuidado y prevención en salud integral del escolar, en la actualidad ha emprendido automatizar e integrar sus programas de salud escolar, buscando obtener facilidad en el manejo de la información del servicio que realizan los programas de salud escolar (Programa “Nutrición y salud escolar” y Programa “Por una sonrisa sana y feliz del escolar”) en todo el alumnado que tiene a cargo en los 6 macrodistritos del Municipio de La Paz, debido a que el trabajo y el proceso de administración de información es manual en más de 160.000 alumnos por gestión.

El presente trabajo denominado “Sistema Integrado de Salud”, es un sistema informático desarrollado en plataforma Web bajo un enfoque RIA, cuyo propósito es integrar el flujo de información de sus programas de salud en ejecución, anteriormente mencionados y producir información en tiempo real, con la implementación de un “Subsistema de Información Odontológica” que pertenecerá al programa: “Por una sonrisa sana y feliz del escolar”; el desarrollo de éste subsistema constituirá la base para un sistema integrado, orientado a reemplazar el flujo de papeles por el flujo de información computarizada, con una base de datos centralizada, a partir de una única carga de datos y producir múltiples combinaciones de información: nutricional, odontológica, afecciones de enfermedades, etc., facilitando la generación de información con claridad, confiabilidad y en tiempo real.

Para el desarrollo del proyecto se utilizó la metodología UWE (Ingeniería Web basada en UML), práctico para construir modelos de análisis y diseño para sistemas Web y un enfoque del desarrollo ágil del software basado en el modelo incremental con un proceso de software personal.

Para cuidar los detalles de la calidad del sistema se utilizaron métricas de evaluación Web (WEB-SITE QEM) basada en la ISO 9126 y se comprobó que este responde a las exigencias del usuario.

Para resguardar la seguridad de la información aplicamos la encriptación transparente (TDE) en la base de datos de SQL Server 2008, se proponen políticas de seguridad y para el análisis de costo del sistema se utiliza el modelo COCOMO II.

ÍNDICE GENERAL

CAPÍTULO I – MARCO REFERENCIAL	Pág.
1.1 INTRODUCCIÓN.....	2
1.2 ANTECEDENTES	4
1.2.1 Antecedentes de la Institución	4
1.2.2 Antecedentes de Trabajos Similares.....	5
1.3 ELECCIÓN DE TEMA.....	7
1.4 DEFINICIÓN DEL PROBLEMA.....	8
1.4.1 Formulación del problema.....	10
1.5 JUSTIFICACIÓN.....	10
1.5.1 Justificación técnica.....	10
1.5.2 Justificación operativa	11
1.5.3 Justificación económica.....	11
1.6 OBJETIVOS.....	12
1.6.1 Objetivo General.....	12
1.6.2 Objetivos Específicos	12
1.7 ALCANCES Y LÍMITES	13
1.7.1 Alcances.....	13
1.7.2 Límites	14
1.8 APORTES	15
1.8.1 Aporte Teórico.....	15
1.8.2 Aporte Práctico	15
1.9 METODOLOGÍA Y HERRAMIENTAS	16
CAPÍTULO II – MARCO TEÓRICO.....	
2.1 SISTEMAS DE INFORMACIÓN	19
2.1.1 Sistema Integrado	19
2.1.2 Sistemas de Información de Salud (SIS)	20
2.1.3 Historia Clínica Computarizada (HCC).....	21
2.2 INGENIERÍA DEL SOFTWARE.....	24

2.2.1	Ingeniería Web.....	24
2.2.2	Crisis del software	25
2.2.3	DESARROLLO ÁGIL	29
2.2.4	Principales Modelos de Gestión Ágil	30
2.2.5	Programación Extrema	31
2.3	REINGENIERÍA	34
2.3.1	Ingeniería Inversa	35
2.3.2	Ingeniería Directa	36
2.4	LENGUAJE UNIFICADO DE MODELADO	36
2.4.1	¿Qué es UML?.....	37
2.4.2	Diagramas UML.....	38
2.5	UWE (UML-Based Web Engineering)	41
2.5.1	UWE y su relación con UML.....	43
2.5.2	Modelos de UWE	43
2.5.2.1	Modelo de casos de uso.....	43
2.5.2.2	Modelo Conceptual	45
2.5.2.3	Modelo de navegación.....	46
2.5.2.4	Modelo de Presentación	47
2.6	RICH INTERNET APPLICATIONS (RIA)	48
2.7	CALIDAD DEL PRODUCTO SOFTWARE	51
2.7.1	ISO 9126.....	52
2.7.2	Web QEM.....	52
2.7.3	Usabilidad.....	53
2.7.4	Principios o reglas de usabilidad	54
2.8	HERRAMIENTAS INFORMÁTICAS.....	55
2.8.1	Tecnología de la plataforma .NET.....	55
2.8.2	Visual Studio 2010	55
2.8.3	SQL Server 2008	55
2.8.3.1	Características de SQL Server 2008 [URL-14].....	56
2.8.4	Arquitectura de tres capas.....	57
2.9	SEGURIDAD.....	58

2.9.1	Cifrado de datos transparente (TDE).....	58
-------	--	----

CAPÍTULO III – MARCO APLICATIVO

3.1	ANÁLISIS Y DISEÑO DEL SISTEMA DE INFORMACIÓN.....	61
3.2	OBTENCIÓN DE REQUISITOS	61
3.2.1	Definición de Actores	62
3.2.2	Lista de Requerimientos del Sistema.....	63
3.2.3	Definición de procesos	64
3.2.4	Herramientas de Desarrollo	66
3.3	MODELO DE CASOS DE USO	67
3.3.1	Caso de Uso Principal.....	67
3.3.2	Descripción de Casos de Uso	68
3.3.2.1	Caso de Uso Ingresar a Sistema	68
3.3.2.2	Caso de Uso Gestionar Usuarios.....	70
3.3.2.3	Caso de Uso Gestionar Pacientes	72
3.3.2.4	Caso de Uso Odontograma.....	74
3.3.2.5	Caso de Uso Historial Clínico.....	76
3.4	MODELO DE CONTENIDO	79
3.5	MODELO DE NAVEGACIÓN.....	80
3.5.1	Mapa de Navegación GUI Administrador.....	82
3.6	MODELO DE PRESENTACIÓN.....	83
3.7	DIAGRAMA ENTIDAD RELACIÓN	87
3.8	DIAGRAMA DE CLASES.....	88
3.9	MODELO FÍSICO	89
3.10	IMPLEMENTACIÓN	90
3.10.1	Captura de ventanas del sistema	90
3.11	SEGURIDAD.....	95
3.11.1	Seguridad Física.....	95
3.11.2	Seguridad Lógica	95
3.11.3	Cifrado de la base de datos	95
3.11.4	Políticas de seguridad	97

CAPÍTULO IV – CALIDAD DE SOFTWARE

4.1	METRICAS DE EVALUACIÓN DE CALIDAD WEB (WEB-SITE QEM).....	99
4.1.1	Criterios de Usabilidad	99
4.1.2	Eficiencia	100
4.1.3	Mantenimiento.....	101
4.1.4	Portabilidad.....	101
4.2	PRUEBAS	108
4.2.1	Revisión del Contenido	108
4.2.2	Pruebas de Navegación.....	109
4.2.3	Pruebas de enlaces	109
4.2.4	Pruebas de usabilidad	109
4.2.5	Pruebas de seguridad	109
4.2.6	Pruebas Unitarias (Caja Blanca).....	110
4.3	ANÁLISIS DE COSTO	111

CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES

5.1	CONCLUSIONES.....	119
5.2	RECOMENDACIONES	120

ÍNDICE DE FIGURAS

FIGURA 1.1	REPRESENTACIÓN DE RESULTADOS DEL MUNICIPIO DE LA CIUDAD DE LA PAZ.....	5
FIGURA 2.1	CICLO DE ENTREGA DE LA PE	32
FIGURA 2.2	EL PROCESO DE LA PROGRAMACIÓN EXTREMA	32
FIGURA 2.3	RESULTADOS DEL ESTUDIO CHAOS (2000-2006)	33
FIGURA 2.4	REINGENIERÍA.....	35
FIGURA 2.5	CASOS DE USO	39
FIGURA 2.6	DIAGRAMA DE CLASES	40
FIGURA 2.7	DIAGRAMA DE SECUENCIA	40
FIGURA 2.8	MODELOS UTILIZADOS EN EL MÉTODO DE CREACIÓN UWE.....	42
FIGURA 2.9	DIAGRAMA DE CASOS DE USO	44
FIGURA 2.10	DIAGRAMA DE CONTENIDO.....	46

FIGURA 2.11 DIAGRAMA DE NAVEGACIÓN	47
FIGURA 2.12 MODELADO ABSTRACTO DE LA INTERFAZ DE USUARIO	48
FIGURA 2.13 IDEA PRINCIPAL DE UN ENFOQUE RIA.....	49
FIGURA 2.14 ARQUITECTURA TÍPICA DE RIA.....	50
FIGURA 2.15 ARQUITECTURA DE TRES CAPAS.....	57
FIGURA 3.1 CASO DE USO PRINCIPAL	67
FIGURA 3.2 CASO DE USO INGRESAR A SISTEMA.....	68
FIGURA 3.3 CASO DE USO GESTIONAR USUARIOS	70
FIGURA 3.4 CASO DE USO GESTIONAR PACIENTES	72
FIGURA 3.5 CASO DE USO ODONTOGRAMA	74
FIGURA 3.6 CASO DE USO HISTORIAL CLÍNICO	76
FIGURA 3.7 CLASES DE ANÁLISIS	79
FIGURA 3.8 DIAGRAMA DE NAVEGACIÓN PARA VISITANTE.....	80
FIGURA 3.9 DIAGRAMA DE NAVEGACIÓN PARA ADMINISTRADOR.....	80
FIGURA 3.10 DIAGRAMA DE NAVEGACIÓN PARA ESTUDIANTE ODONTÓLOGO	81
FIGURA 3.11 DIAGRAMA DE NAVEGACIÓN DEL PROFESIONAL ODONTÓLOGO.....	81
FIGURA 3.12 MAPA DE NAVEGACIÓN GUI ADMINISTRADOR	82
FIGURA 3.13 DISEÑO DE INTERFAZ – AUTENTIFICACIÓN.....	83
FIGURA 3.14 DISEÑO DE INTERFAZ – BÚSQUEDA DE USUARIOS	84
FIGURA 3.15 DISEÑO DE INTERFAZ – PRODUCTOS	84
FIGURA 3.16 DISEÑO DE INTERFAZ – PERFIL DE USUARIO.....	85
FIGURA 3.17 DISEÑO DE INTERFAZ – ODONTOGRAMA	86
FIGURA 3.18 DIAGRAMA ENTIDAD RELACIÓN.....	87
FIGURA 3.19 DIAGRAMA DE CLASES	88
FIGURA 3.20 MODELO FÍSICO DE DATOS	89
FIGURA 3.21 INGRESO AL SUBSISTEMA DE INFORMACIÓN ODONTOLÓGICA	90
FIGURA 3.22 GESTIÓN DE USUARIOS	91
FIGURA 3.23 BÚSQUEDA DE UNIDADES EDUCATIVAS.....	91
FIGURA 3.24 PERFIL DE UNIDAD EDUCATIVA.....	92
FIGURA 3.25 BÚSQUEDA DE PACIENTE	92
FIGURA 3.26 GESTIÓN DE PACIENTE.....	93

FIGURA 3.27 ODONTOGRAMA DE PACIENTE 1	93
FIGURA 3.28 ODONTOGRAMA DE PACIENTE 2	94
FIGURA 3.29 REPORTE DE ÍNDICES ODONTOLÓGICO DE PACIENTE	94
FIGURA 3.30 IMPLEMENTACIÓN DEL CIFRADO DTE A LA BASE DE DATOS	96
FIGURA 3.31 COPIA DE SEGURIDAD ENCRIPTADO.....	96

ÍNDICE DE TABLAS

TABLA 2.1 COMPARACIÓN DE HISTORIA CLÍNICA COMPUTARIZADA CON LA TRADICIONAL ...	23
TABLA 2.2 PRINCIPALES PROBLEMAS CON LOS MODELOS DE PROCESO SOFTWARE TRADICIONALES.....	28
TABLA 2.3 PRINCIPIOS DE LOS MÉTODOS ÁGILES.....	30
TABLA 2.4 CALIDAD APLICACIÓN WEB.....	53
TABLA 3.1 TAREAS PARA LA OBTENCIÓN DE REQUISITOS	62
TABLA 3.2 LISTA DE ACTORES	63
TABLA 3.3 FUNCIONES BÁSICAS DEL SISTEMA	64
TABLA 3.4 ESPECIFICACIÓN DE CASO DE USO INGRESAR A SISTEMA	69
TABLA 3.5 ESPECIFICACIÓN DE CASO DE USO GESTIONAR USUARIOS.....	71
TABLA 3.6 ESPECIFICACIÓN DE CASO DE USO GESTIONAR PACIENTES	73
TABLA 3.7 ESPECIFICACIÓN DE CASO DE USO ODONTOGRAMA	75
TABLA 3.8 ESPECIFICACIÓN DE CASO DE USO HISTORIAL CLÍNICO	77
TABLA 4.1 ÁRBOL DE REQUERIMIENTOS DE CALIDAD PARA EL DOMINIO DE LA APLICACIÓN WEB PARA EL GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO	102
TABLA 4.2 DESARROLLO DE LA METODOLOGÍA CON LOS PESOS DE AGREGACIÓN.....	104
TABLA 4.3 RESUMEN DE LOS RESULTADOS OBTENIDOS	108
TABLA 4.4 PRUEBAS UNITARIAS	110
TABLA 4.5 PUNTOS DE FUNCIÓN	111
TABLA 4.6 PUNTOS DE FUNCIÓN NO AJUSTADOS.....	112
TABLA 4.7 DETERMINACIÓN DE COMPLEJIDAD.	113
TABLA 4.8 CONVERSIÓN DE PUNTOS DE FUNCIÓN A KLDC	114
TABLA 4.9 COEFICIENTES AB Y CB Y LOS EXPONENTES BB Y DB	115

TABLA 4.10 COSTO SOFTWARE EXPRESS.....	116
TABLA 4.11 COSTO ELABORACIÓN DEL PROYECTO	116
TABLA 4.12 COSTO TOTAL.....	117

CAPÍTULO I
MARCO REFERENCIAL

1 MARCO REFERENCIAL

1.1 INTRODUCCIÓN

En la actualidad el avance de las Nuevas Tecnologías de Información y Comunicación (NTIC's)¹, ha hecho de los sistemas de información (SI) indispensables en cualquier institución que desee alcanzar altos niveles de calidad en los servicios que presta.

No se puede pensar en una institución que carezca de información en cualquier ámbito que plantee desarrollarse, ésta es la principal característica que definirá estrategias de cara al futuro y medirá productividad como efectividad de los procesos efectuados. [DAV05]

Actualmente la información es el activo más importante de cualquier organización. Por otro lado con el incremento de la tecnología, las organizaciones han optado por implementar sistemas de información, automatizados e integrados; como un proceso de detección de problemas y también de oportunidades que permita almacenar grandes cantidades de información, procesarlas de forma rápida y presentar resultados requeridos, para apoyar la toma de decisiones.

El Gobierno Autónomo Municipal de La Paz (GAMLP), tiene como misión institucional contribuir a la satisfacción de las necesidades colectivas de los habitantes del Municipio, mejorando la calidad de vida de la población en sus aspectos tangibles e intangibles. Considera que las condiciones fundamentales del desarrollo humano, entendido como el proceso de ampliación de capacidades y oportunidades de la población, es una vida saludable; en este sentido, ha establecido como meta y prioridad de gestión: el desarrollo integral de niños y niñas en edad escolar².

El GAMLP, está dividido en varias oficialías, las cuales tienen una misión específica. La Oficialía Mayor de Desarrollo Humano (OMDH) es una de ellas y tiene a su cargo la Dirección de Educación (DE); ésta dirección a su vez tiene a su cargo la Unidad de Alimentación Complementaria Escolar (UNACE) que está encargada de la entrega de

¹ Nuevas Tecnologías de Información y Comunicación. Agrupan los elementos y las técnicas utilizadas en el tratamiento y la transmisión de la información, principalmente de informática, internet y telecomunicaciones.

² Convención sobre los Derechos del Niño, Artículo 27.

raciones de alimento complementario escolar (desayuno escolar) en unidades educativas públicas de los 6 macrodistritos del Municipio de La Paz.

La alimentación adecuada es el Derecho Humano³ más importante y esencial para el desarrollo social y económico de un país. Una buena nutrición contribuye a mejorar la eficiencia y los resultados de acciones para el desarrollo: mayor capacidad de aprendizaje, menores gastos para el cuidado de la salud, mayor productividad, entre otros. Es decir, la nutrición como base para el desarrollo, está relacionada principalmente con la educación, salud, pobreza y género. [COM02]

Por esta razón, la Unidad de Alimentación Complementaria Escolar (UNACE), ha implementado en unidades educativas que tiene a su cargo, los programas: “Nutrición y salud escolar”, “Atención integral al menor de 6 años” y “Por una sonrisa sana y feliz del escolar”, basados en prevención, seguimiento y evaluación, para contribuir a mejorar el estado nutricional y salud integral del escolar. [GOB06, Pág. 7]

El presente trabajo propone un “Sistema Integrado de Salud”, sistema informático en plataforma Web bajo un enfoque RIA⁴ para la UNACE cuyo propósito es integrar el flujo de información de sus programas de salud en ejecución, anteriormente mencionados y producir información en tiempo real, a partir de la implementación de un “Subsistema de Información Odontológica” que pertenecerá al programa: “Por una sonrisa sana y feliz del escolar”; el desarrollo de éste subsistema constituirá la base para un Sistema Integrado de Salud, orientado a reemplazar el flujo de papeles por el flujo de información computarizada, basado en una Base de Datos Relacional, a partir de una única carga de datos y producir múltiples combinaciones de información: nutricional, odontológica, afecciones de enfermedades, etc., facilitando la generación de información con integridad, claridad, y confiabilidad.

³ Declaración de Derechos Humanos, Artículo 25; párrafo 1.

⁴ RICH INTERNET APPLICATIONS (RIA), término acuñado por Macromedia en marzo de 2002, las cuales son un nuevo tipo de aplicación Web cuyo objetivo es incrementar y mejorar las opciones y capacidades de las aplicaciones Web tradicionales.

1.2 ANTECEDENTES

1.2.1 Antecedentes de la Institución

La Unidad de Alimentación Complementaria Escolar (UNACE), es dependiente del Gobierno Autónomo Municipal de La Paz (GAMLP), sin fines de lucro, creada en 1998 por Resolución Municipal N° 0674.

El objetivo principal de la UNACE es la dotación de la alimentación complementaria escolar (programa de “Desayuno escolar”)⁵ y la salud integral del escolar (programa de “Nutrición y salud escolar”)⁶, con cobertura a 410 unidades educativas, entre unidades fiscales, unidades de convenios y centros especiales, de turnos: mañana, tarde y noche, llegando a 168.000 [GOB09, Pág. 3] estudiantes favorecidos en todo el Municipio de la ciudad de La Paz.

Con el propósito de obtener información actualizada sobre la situación de salud integral escolar, en las gestiones 2005-2006 se elabora el “Perfil Epidemiológico del Escolar” en el Municipio de La Paz⁷, proceso desarrollado a partir de la implementación del carnet de salud del escolar, como un producto de alianzas estratégicas promovidas por la Unidad de Alimentación Complementaria Escolar (UNACE) con el Ministerio de Salud y Deportes, Servicio Departamental de Salud (SEDES), Servicio Departamental de Educación (SEDUCA), Universidad Mayor de San Andrés (UMSA), además de la participación activa de representantes de la comunidad educativa.

Los principales resultados del “Perfil Epidemiológico del Escolar” se resumen en la Figura N° 1, con referencia a indicadores de estado nutricional, proliferación de afecciones y prevalencia de caries dentales. Realizando el diagnóstico general de las diferentes prevalencias en edad escolar, se comprueba que el 78,87% [GOB06, Pág. 6] de escolares sufren de una alta persistencia de caries dental, es decir de cada 100 escolares 80 tienen

⁵ Programa implementado a partir del año 2000, enmarcado en el tercer lineamiento estratégico “Municipio solidario, equitativo y afectivo”.

⁶ Programa de prevención, seguimiento y evaluación en salud integral escolar, implementado a partir de año 2005.

⁷ Estudio realizado en todas las unidades educativas públicas pertenecientes a los 6 macrodistritos del Municipio de la ciudad de La Paz.

problemas fundamentalmente de caries, lo que originó la necesidad de enfocar el tema de alimentación y salud de forma integral. Como consecuencia del “Perfil Epidemiológico del Escolar”⁸ desde junio del 2007, se implementó el programa “Por una Sonrisa Sana y Feliz del Escolar”, con sus tres pilares:

- Educación en nutrición, orientada en salud dental.
- Educación en higiene buco dental.
- Aplicación del TRA⁹ (Tratamiento Restaurativo Atraumático).

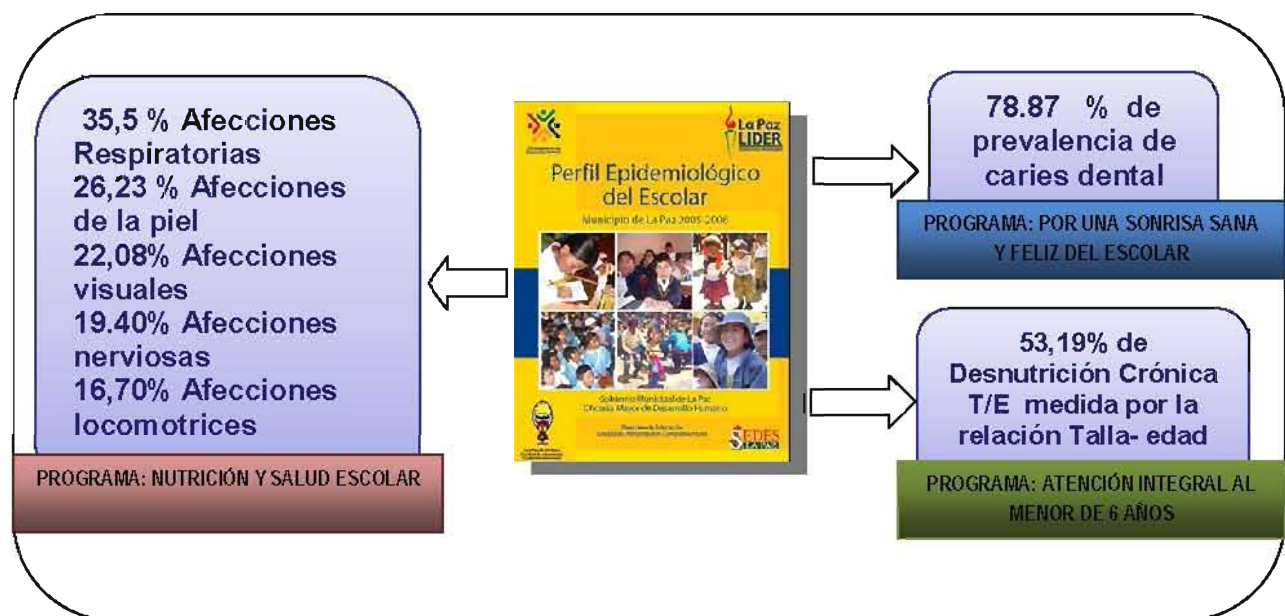


Figura 1.1 Representación de resultados del Municipio de la ciudad de La Paz.
Fuente: “Perfil Epidemiológico del Escolar”, Gobierno Autónomo Municipal de La Paz.

1.2.2 Antecedentes de Trabajos Similares

En la actualidad la UNACE cuenta con el sistema de información “Unace-SYS”, que proporciona información de unidades educativas, infraestructura, personal, detalle de

⁸ El Perfil Epidemiológico del Escolar constituyó en un referente para el diseño y ajuste de políticas, programas y proyectos orientados a mejorar la calidad de vida de nuestros niños, niñas y adolescentes en edad escolar.

⁹ Técnica sencilla y rápida para realizar obturaciones en caries de primer y segundo grado.

ejecución, información del desayuno escolar, otros. Unace-SYS es la integración de dos proyectos de grado realizado en el año 2004 y 2007. También cuenta con otro sistema de información independiente de “Unace-SYS”, para el manejo de información del programa: “Nutrición y salud escolar”, desarrollado como un proyecto de grado en el año 2006.

- Entre los proyectos de grado desarrollados recientemente en la Carrera de Informática respecto a la Unidad de Alimentación Complementaria Escolar (UNACE), se tiene:

1) Título Administración de Información de la Unidad de Nutrición y Alimentación Complementaria Escolar

Autor Denisse Janeth Ríos Quisbert

Año 2007

Descripción Sistema de Administración de la UNACE que se basa en el análisis y diseño de los módulos de seguimientos de actas, seguimientos de promotores, evaluación de personal, registro de unidades educativas, promotores, actas, asignación de promotores.

2) Título Sistema automatizado de tratamiento de la información de la evaluación nutricional de niños de edad escolar.

Autor Américo Serafín Soliz Sepúlveda

Año 2006

Descripción Sistema basado en el programa “Nutrición y salud escolar” que realiza evaluaciones nutricionales periódicas y colectivas, basado en la recolección de datos del carnet de salud escolar, estos datos ingresan a un procesamiento de evaluación y al compararlo con estándares nutricionales arrojarán resultados, útiles para los profesionales en nutrición de la unidad.

Los proyectos de grado mencionados anteriormente fueron desarrollados, e implementados como sistemas de información, de acuerdo a requerimientos y necesidades independientes, actualmente en funcionamiento como aplicación de escritorio.

- Entre los proyectos de grado desarrollados recientemente en la Carrera de Informática respecto al tema propuesto, se tiene:

3) Título Sistema de Contabilidad Integrada - UMSA

Autor Lizeth Jimena Coaquira Quisbert

Año 2007

Descripción Sistema que opera de forma común, única y uniforme, integrador de los registros presupuestarios, económicos, financieros y patrimoniales, orientado a determinar los costos de las operaciones, aplicando la teoría contable, principios de contabilidad integrada.

1.3 ELECCIÓN DE TEMA

En la actualidad la UNACE ha emprendido automatizar e integrar todos sus programas de salud integral escolar, buscando obtener información en línea y completa de todo el alumnado que tiene a cargo en los 6 macrodistritos del Municipio de La Paz.

Los sistemas de información implementados en la UNACE, no fueron desarrollados para una estructura integrada de información o plataforma Web, teniendo muchos inconvenientes y debiéndose realizar ajustes manuales, necesarios para salvar las situaciones de problemas, es así que el programa: “Por una sonrisa sana y feliz del escolar” no está automatizado en forma independiente o integrado en algún sistema informático de la UNACE.

Si bien la odontología tuvo un salto importante en cuanto a la modernidad y tecnología, el índice de caries, enfermedades periodontal y otras patologías de la cavidad bucal producidas por microorganismos como bacterias, virus y hongos siguen incidiendo en la salud general de nuestra población, especialmente en niños, niñas y adolescentes; a ellos se suman condiciones de pobreza en la que viven la mayoría de los hogares, y sufren además elevados niveles de desnutrición y mortalidad materna e infantil. [MIN09]

El presente trabajo denominado “Sistema Integrado de Salud”, será un sistema informático en plataforma Web bajo un enfoque RIA, que integrará datos, indicadores y

variables de índole odontológica y nutricional (caries dentales, obturaciones, tratamientos dentales, patologías, afecciones, alergias, peso, talla y otros), en forma ordenada, unificando en una aplicación Web la información que controla el programa “Por una sonrisa sana y feliz del escolar” y el programa “Nutrición y salud escolar”.

1.4 DEFINICIÓN DEL PROBLEMA

La UNACE está en un proceso de automatización de todos sus datos, mejoramiento y ampliación de sus programas, justamente para mejorar y agilizar todos sus procesos, en un sistema informático único que integre información oportuna, precisa y confiable, para el seguimiento, control y toma de decisiones de sus programas de salud implementados en escolares de los 6 macrodistritos de la ciudad de La Paz.

El programa “Por una sonrisa sana y feliz del escolar”, viene realizando una cobertura odontológica de 17.813 estudiantes con 40.665 [GOB09, Pág. 6] obturaciones de muelas y dientes en 96 unidades educativas, aplicando la técnica del Tratamiento Restaurativo Atraumático (TRA).

Según perspectivas de la UNACE las proyecciones en prevención, seguimiento y evaluación en salud integral escolar, se incrementará en los siguientes años. El trabajo con información no automatizada es cada vez un problema de escala mayor, basado en datos reales no confiables, existe un mayor margen de error en resultados de avances y exitosas toma de decisiones.

El programa “Por una sonrisa sana y feliz del escolar”, presenta muchas falencias en su trabajo operativo y necesita urgentemente la implementación de un sistema de información; todo su flujo de información y recolección de datos es manual, el procedimiento en manejo de información que hasta ahora se vino ejecutando trae consigo la dificultad en cuanto a proceso: de administración, control y seguimiento de información, los procesos también se los realiza de forma manual o en su caso migrados a hojas electrónicas Excel.

- Los datos generales recolectados de cada escolar atendido por el programa “Por una sonrisa sana y feliz del escolar”, se almacena manualmente en hojas físicas, lo que provoca demora en la búsqueda específica de un paciente.
- El diagnóstico odontológico o línea base que se realiza a cada escolar, es recolectado manualmente, no contar con un sistema que automatice esta información deriva en un conteo manual muy moroso para obtener índices odontológicos, proliferación de caries, etc., por cada registro recolectado.
- No se tiene información actualizada y confiable en tiempo real, de índices odontológicos, cuantificación de dientes sanos, caries dentales, obturaciones, etc., en cada unidad educativa y macrodistrito del Municipio.
- Los reportes y/o informes del programa difieren demasiado tiempo, cuantificar toda la información recolectada es torpe; por alumno, curso, unidad educativa y finalmente macrodistrito, obteniendo datos muy generales.
- Existe demasiada información sin digitalizar de escolares que fueron atendidos por el programa desde que el mismo inicio: antecedentes generales, curaciones realizadas, seguimiento, etc., sin describir una línea de avance real del trabajo que realiza el programa en gestiones anteriores, conllevando a trabajar en redundancia, sin avance y con fracasadas toma de decisiones.
- No existe un registro organizado de los odontólogos egresados¹⁰ que trabajan en el programa, a los cuales asignan individualmente: material de trabajo e insumos odontológicos.
- Existe demora en la búsqueda de información, desorganización en el almacenamiento y manipuleo de la información acumulada. Ocasionando pérdida de tiempo y deterioro de hojas físicas.
- La información es inconsistente y tienen mayor probabilidad de ser errónea.
- Existe duplicidad de esfuerzos como de tareas y un nivel de equivocación alto en la representación de datos reales.

¹⁰ Estudiantes egresados de la Carrera de Odontología de las Universidades de convenio.

- No existe coordinación, integración, flujo e intercambio de información con otros programas que ejecuta UNACE, y que tienen la misma misión: salud integral escolar, enfáticamente con el programa: “Nutrición y salud escolar”.
- No existe información oportuna o referente de un historial clínico de salud actualizado con antecedentes clínicos (enfermedades, alergias, situación nutricional y de salud), de cada escolar, indispensable a la hora de tomar medidas clínicas u curaciones odontológicas de los pacientes.

1.4.1 Formulación del problema

¿Un nuevo sistema logrará reducir el trabajo manual en el programa “Por una sonrisa sana y feliz del escolar”, integrando el mismo flujo e información que controla el programa “Nutrición y salud escolar”, eliminando duplicidad de datos y/o procesos, brindando información confiable oportuna y significativa a nivel de una aplicación Web para una buena toma de decisiones?

1.5 JUSTIFICACIÓN

1.5.1 Justificación técnica

El desarrollo del sistema traerá consigo beneficios al Gobierno Autónomo Municipal de La Paz mediante la Unidad de Alimentación Complementaria (UNACE), ya que permitirá tener a su alcance estadísticas¹¹ y datos reales en prevención, seguimiento y evaluación de la salud integral escolar en línea (internet), posibilitando una correcta toma de decisiones, mejores y nuevas proyecciones a largo plazo. El Sistema Integrado de Salud aplicará

¹¹ Al ser la Estadística una ciencia que conlleva otro estudio complejo y relacionado con la investigación científica, cuando nos referimos a estadísticas estaremos haciendo hincapié en la recolección, análisis e interpretación de datos, ya sea para ayudar en una toma de decisiones o para explicar condiciones regulares o irregulares de algún fenómeno o estudio aplicado.

tecnologías Web basadas en las RIAs¹² (AJAX, Java Script y JQuery), permitiendo una adecuada explotación del manejo de la información de manera fácil [WEI04].

La institución, actualmente cuenta con la infraestructura adecuada, tanto a nivel de hardware (equipos de computación y servidores) como de software (Visual Studio 2010, SQL Server 2008, IIS), para la implementación del presente proyecto, es decir, cuenta con los equipos necesarios para el desarrollo, pruebas y culminación del proyecto.

1.5.2 Justificación operativa

El desarrollo del Sistema Integrado de Salud tendrá como principal beneficiario a los niños que son atendidos por dichos programas; la agilización de los procesos permitirá trabajar a los odontólogos y nutricionistas con más énfasis en las unidades educativas que así lo requieran.

También beneficiará a odontólogos, nutricionistas y personal en general de la UNACE; el sistema facilitará el trabajo que desempeñan, realizando el trabajo operativo de una manera más eficiente obteniendo información confiable y actualizada.

1.5.3 Justificación económica

El desarrollo e implementación del sistema utilizará recursos de la institución, existe software actual con sus respectivas licencias, también se cuenta con computadoras de última generación, lo cual representan costos bajos y de fácil aplicación, así mismo sistematizar ciertos procesos, actividades y tareas que en la organización la realizan manualmente o semiautomáticamente, es de gran utilidad.

La información generada por el sistema será oportuna para la buena toma de decisiones, generando beneficios en la obtención de resultados y la entrega oportuna de información, satisfaciendo así las expectativas de la misión como el deseo de que la pérdida de tiempo sea un costo que influye negativamente en el cumplimiento oportuno de la misma.

¹² Los usuarios finales son los que experimentan las grandes ventajas de las RIAs. Cabe destacar que una RIA bien estructurada y diseñada disminuye el tiempo en que el usuario encuentre la información que necesita sin esperar a que una página nueva se cargue.

1.6 OBJETIVOS

1.6.1 Objetivo General

Desarrollar un Sistema Integrado de Salud, a partir de un Subsistema de Información Odontológica para el programa “Por una sonrisa sana y feliz del escolar” e integrar el sistema de información del programa “Nutrición y salud escolar”, mejorando el flujo de información, organización y seguimiento, eliminando duplicidad de datos, procesos y tareas.

1.6.2 Objetivos Específicos

- Desarrollar el Subsistema de Información Odontológica que estará implementado en el Sistema Integrado de Salud.
- Diseñar el portal Web¹³ de la UNACE, que será la puerta de entrada del Sistema Integrado de Salud.
- Realizar una reingeniería de procesos y de base de datos del sistema informático que contempla el programa: “Nutrición y salud escolar”, que será modelo para el desarrollo del Subsistema de Información Odontológica.
- Implementar una base de datos centralizada con el gestor de base de datos SQL Server 2008, que integre todos los datos necesarios de los programas de la UNACE en cuestión.
- Diseñar un historial clínico, para el control y seguimiento de los pacientes escolares atendidos en sus unidades educativas de acuerdo al tipo de patologías y curaciones que fueron registradas.
- Diseñar un Odontograma para tener una representación anatómica o esquemática de las piezas dentales, organizados por cuadrantes y sobre los cuales se señalan las patologías y/o tratamientos del paciente escolar.

¹³ Un portal Web es un sitio Web cuya característica fundamental es la de servir de puerta de entrada (única) para ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema. Principalmente un portal Web está dirigido a resolver necesidades de información específica de un tema en particular.

- Diseñar módulo de gestión de datos generales de: usuarios del sistema, pacientes, unidades educativas y productos; control de stock de insumos odontológicos, procesamiento de datos recolectados validados y depurados.
- Diseñar el módulo de reportes generales con gráficas estadísticas y obtención de resultados del trabajo operativo realizado por el programa, que brindará información explícita sobre escolares atendidos.
- Generar reporte de historial clínico y odontograma para una evaluación y seguimiento del paciente.
- Definir diferentes roles y accesos de seguridad de usuarios para mantener la integridad de la información que será almacenada en la base de datos central, previniendo cualquier adulteración posterior.

1.7 ALCANCES Y LÍMITES

1.7.1 Alcances

El proyecto se orienta al análisis, diseño, desarrollo y pruebas del Subsistema de Información Odontológica, donde se identificaron los siguientes alcances:

- El Subsistema de Información Odontológica que pertenece al programa: “Por una sonrisa sana y feliz del escolar” se desarrollará e implementará desde su fase inicial hasta su fase final, utilizando una metodología adecuada.
- El sistema de información del programa: “Nutrición y salud escolar” existente en la actualidad, se fusionará al Subsistema arriba mencionado, manteniendo todos sus parámetros y requerimientos en cuestión como sistema de información establecido, nos concentraremos en una reingeniería de software basado en su modelo relacional, entidad relación y estructura de relaciones que se integrará al modelo relacional del Subsistema de Información Odontológica, generando así una base de datos central del Sistema Integrado de Salud.
- Módulo de historial clínico.
- Módulo de stock de productos odontológicos proporcionados al programa.

- El sistema de información se utilizará en la Web extranet/intranet del Municipio a todo nivel de conexión de internet de una computadora.
- El sistema de información abarcará la información a nivel municipal, es decir los 6 macrodistritos de toda la ciudad de La Paz.

1.7.2 Límites

- La implementación del Sistema Integrado de Salud, solo integrará a 2 programas en cuestión que son los referentes en salud integral escolar que ejecuta la UNACE: programas: “Nutrición y Salud Escolar” y “Por una sonrisa sana y feliz del escolar”. No contemplará la implementación e integración de otros programas pertenecientes a la UNACE.
- Solo los usuarios con permisos acceden completamente al sistema, el resto navega en el portal a nivel de extranet del Municipio.
- No se contempla el control de personal, se diseñará un módulo de registro de odontólogos egresados¹⁴ que trabajan en el programa.
- Se diseñará un módulo de stock de insumos, esto no implica que es un módulo de inventarios, ya que significaría un módulo de mayor complejidad.
- El proyecto abarcará solo a unidades educativas públicas: unidades fiscales, unidades de convenios y centros especiales, que pertenecen a los 6 macrodistritos del Municipio de La Paz. El proyecto no contempla otras unidades educativas que estén fuera de este rango.
- Podrán beneficiarse solo aquellos usuarios que dispongan de una conexión a internet, por tratarse de un proyecto de plataforma Web.

¹⁴ Estudiantes egresados de la Carrera de Odontología de las diferentes universidades en convenio, que hacen su internado rotatorio como parte de su formación académica.

1.8 APORTES

1.8.1 Aporte Teórico

Dentro la ingeniería Web existen varias metodologías de desarrollo para aplicaciones Web, la presente propuesta hace énfasis a la Metodología Ingeniería Web basada en UML (UML-Based Web Engineering “UWE”), y un enfoque del desarrollo ágil del software basado en el modelo de proceso incremental con un proceso de software personal.

El aporte teórico del presente trabajo es la utilización de estos paradigmas como una metodología personalizada, puntualizando los pasos que éste requiere para su desarrollo, según las necesidades del proyecto; requiere de un práctico modelado Web por su integración con otros subsistemas como de una rápida implementación, entendiendo así el objetivo principal del sistema, y limitantes en cuanto a tiempo y recursos (económicos, humanos y tecnológicos).

En este sentido el presente Proyecto de Grado servirá como base para nuevos proyectos y sistemas, que contemplen estas características en el desarrollo e implementación de sistemas de información.

1.8.2 Aporte Práctico

La relevancia social sobre el tema de salud integral es muy importante y delicada, sobre todo por la falta de atención o cobertura del gobierno central [CAR00], la UNACE trabaja con varios programas y varias instituciones, especializados en la alimentación complementaria y salud integral, pero limitándose a la prevención, seguimiento y evaluación, lo cual es de trascendental importancia por el control y datos que se manejan dentro de esta unidad.

La información obtenida a través del Sistema Integrado de Salud servirá para sustentar, nuevas proyecciones, planes de contingencia y alcances (mediano y largo plazo), basados en la salud básica del niño, niña y adolescente contemplados como un derecho universal.

Los beneficiados directos del presente Proyecto, es el personal de la UNACE y los escolares atendidos por los programas en ejecución, el sistema facilitará el flujo e intercambio de información entre los programas de salud de manera rápida y segura, se automatizarán la mayoría de los procesos manuales, se solucionarán problemas de datos erróneos e inconsistentes, es decir, se obtendrán datos e información confiable, que son indispensables a la hora de nuevas decisiones: prevenciones y proyecciones basadas en la salud integral del escolar, gratificando a los escolares con mejores atenciones y rápido apoyo.

1.9 METODOLOGÍA Y HERRAMIENTAS

Las metodologías y herramientas que se utilizan para el desarrollo del presente proyecto son:

- Metodología Ingeniería Web basada en UML (UWE). El Modelado de aplicaciones Web con UML consiste en aplicar la metodología de análisis y diseño orientado a objetos al desarrollo de aplicaciones Web ya que hasta el día de hoy, lo más importante en el desarrollo de aplicaciones Web han sido las herramientas. La fácil creación de hojas HTML y en general de sitios Web, usando herramientas simples, ha hecho que el desarrollo de este tipo se haga sin un trabajo serio de análisis y diseño. Dado que las páginas Web son los principales componentes de la arquitectura Web. Usando UML podemos ver una página Web como un objeto, esto nos permitirá poder establecer los métodos y atributos de una página Web.
- La usabilidad se puede considerar como la capacidad del producto software para permitir que usuarios específicos logren realizar tareas específicas con productividad, efectividad, seguridad y satisfacción en determinados casos de uso. El objetivo de un producto es que posea la calidad necesaria y suficiente para que satisfaga las necesidades de usuario explícitas e implícitas. [USE11]
- Las Rich Internet Applications, o RIA (en español "aplicaciones de Internet enriquecidas"), son aplicaciones Web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador

Web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.

Las RIA surgen como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales. Buscan mejorar la experiencia del usuario.

- Servidor Windows Server 2003.
- Internet Information Server IIS.
- Servidor de Base de Datos SQL Server 2008.
- Lenguaje de Programación C#.
- Entorno de Desarrollo Integrado IDE Microsoft Visual Studio 2010.
- Crystal Reports.

CAPÍTULO II

MARCO TÉORICO

2 MARCO TEÓRICO

2.1 SISTEMAS DE INFORMACIÓN

Sistema de información (SI) es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Dichos elementos formarán parte de alguna de estas categorías:

- Elementos de un sistema de información.
- Personas.
- Datos.
- Actividades o técnicas de trabajo.
- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos (incluyendo procesos manuales y automáticos) dando lugar a información más elaborada y distribuyéndola de la manera más adecuada posible en una determinada organización en función de sus objetivos. [PRE06]

2.1.1 Sistema Integrado

Un Sistema Integrado de Información (SII) se refiere a un sistema que integra o centraliza la información misional de una organización facilitando su uso a lo largo y ancho de todas las áreas de la Organización. Los SII cubren los procesos de negocio de un tipo de organización específica.

Los sistemas como ERP¹⁵ pueden ser considerados como SII. Algunos proveedores ofrecen soluciones para sectores específicos como Gas, Telecomunicaciones, Producción, Finanzas, entre otros. [PRE06]

¹⁵ Enterprise Resource Planning, Software de Información Centralizada orientado a registrar e integrar la mayoría de los procesos de negocios.

2.1.2 Sistemas de Información de Salud (SIS)

La expansión y la transmisión de tecnologías y sus aplicaciones, comprenden diferentes campos de la actividad humana, entre ellos el área de la salud. El aumento de la automatización ha sido un elemento sumamente importante de estos adelantos tecnológicos.

La odontología, al igual que otras actividades del área de salud, no puede llevarse a cabo efectivamente sin contar con una información exacta y a tiempo, información inherente al paciente y sus problemas, procedimientos adecuados para su cuidado, beneficio y limitaciones. Es así que el funcionamiento de un sistema de salud, depende de lo bien que se utilicen sus recursos (maquinaria, dinero, personas, información, etc.), pero la información es, quizá, el recurso más valioso que cualquier institución posee, ya que sin ella todo lo demás es inmanejable.

Un diagnóstico rápido y el tratamiento eficaz de un paciente dependen de la transmisión exacta de órdenes y resultados entre varios servicios de salud. Las mejoras en la eficiencia con un sistema de proceso de datos funcionando adecuadamente, se pueden medir en miles de horas-hombre y en gran cantidad de dinero ahorrado. Igualmente importante, el manejo de información sanitaria automatizada mejora (o mejorará) significativamente la asistencia al paciente, al reducir errores, acelerando el flujo de órdenes y resultados, y haciendo disponible una información más completa para la toma de decisiones [ASS86].

En estos casos, las ventajas del uso de las Nuevas Tecnologías de la Información y la Comunicación (NTIC) en un sistema integrado por programas de salud son obvias:

- Acumular y organizar la información para que sea accesible inmediatamente.
- Mejorar la productividad al reducir el flujo de papel y la redundancia de la información.
- Ayudar en la planificación de recursos y mejora del funcionamiento del sistema.
- Permitir hacer revisiones de calidad de la asistencia.
- Gestionar la información para tratar a los pacientes más eficientemente.

Bajo estas premisas se define un SIS como un sistema global e integrado, diseñado para gestionar la administración, logística y aspectos clínicos de un sistema sanitario [VAN00]. Generalmente es un software compuesto por varios componentes o módulos que

gestionan una gran variedad de subsistemas en las especialidades médicas (Enfermería, Laboratorio, Cirugía, etc.). La función de un SIS es por tanto apoyar las actividades de cada establecimiento del sistema sanitario en los niveles operacional, táctico y estratégico.

2.1.3 Historia Clínica Computarizada (HCC) ¹⁶

La Historia Clínica Computarizada (HCC) supone incorporar las Nuevas Tecnologías de la Información y la Comunicación (NTIC) en el núcleo de la actividad sanitaria. Esto trae como consecuencia que la historia deje de ser un registro de la información generada en la relación entre un paciente y un profesional o un centro sanitario, para formar parte de un sistema integrado de información clínica.

El registro de salud constituye una recopilación sistemática de hechos pertinentes de la vida de un paciente y su historia de salud, incluso de enfermedad (es) y tratamiento (s) pasados y presentes, preparado por aquellos profesionales que contribuyen al cuidado del paciente y su bienestar. Estos registros han sido tradicionalmente basados en papel, que durante la mayor parte del siglo pasado ha sido el vehículo para coleccionar, almacenar y presentar la información del paciente, pero los mismos se han vuelto progresivamente inadecuados debido al incremento intensivo de la información en el ambiente de la salud.

En los últimos cinco años el volumen de la información se ha duplicado y se ha generado más información en los últimos 20 años que en los 5000 años anteriores, razón por la cual es indispensable usar nuevos medios de acceso a las fuentes de información y la tecnología contribuya a esto.

Según el Comité Europeo de Normalización (CEN), una HCC es un: Registro longitudinal y potencialmente multi-institución o multinacional de la atención sanitaria de un único sujeto (paciente), creado y almacenado en uno o varios sistemas físicos con el

¹⁶ En la literatura también se las puede encontrar como Historias Clínicas Electrónicas (HCE) o Historias Clínicas Digitalizadas (HCD). No obstante, si se acepta que la informática es una ciencia que mediante sistemas electrónicos y programas informáticos permite un manejo automático de la información, y en este caso se está hablando de manejo de la información sanitaria de pacientes, parece que el mejor termino que puede ajustarse a dicho concepto sea Historia Clínica Computarizada (HCC).

propósito de informar en la asistencia sanitaria futura del sujeto y proporcionar un registro médico-legal de la asistencia que se le ha suministrado [CEN04].

Por tanto se podría decir que una HCC no es más que un conjunto global y estructurado de información, relacionado con los procesos de la asistencia médico sanitaria de los pacientes, soportado en una (o varias) plataformas informáticas.

El registro de pacientes basado en computadoras (HCC) es el término aceptado para describir la información automatizada del paciente y sigue las pautas generales con respecto al contenido, estructura, privacidad, divulgación de la información, exactitud, autenticación de entrada y almacenamiento, vinculado a la naturaleza de la información en papel. La Historia Clínica Computarizada mejora la calidad de atención haciendo que el registro y su información asociada este siempre disponible para los profesionales, cuando estos lo necesiten, además de contribuir a hacer su trabajo más efectivo y realizar una correcta selección de los servicios prestados para que se adapten a las necesidades y circunstancias de cada paciente. Soluciona problemas físicos de almacenamiento y favorece la interconsulta en lugares remotos compartiendo la información en tiempo real.

La sustitución de la HC tradicional (en soporte papel) por una historia clínica informatizada (HCC) responde a varias necesidades:

1. Dar cumplimiento a las características y objetivos del documento HC en cuanto a los requerimientos del equipo sanitario, manteniendo la confidencialidad.
2. Resolver los dos problemas clásicos de los archivos de HC: el almacenamiento de grandes volúmenes documentales, y la seguridad frente a los riesgos de pérdida y de deterioro.
3. Permitir la transferencia rápida de la información sanitaria existente de un paciente a puntos lejanos, garantizando que cada paciente solo tenga un único expediente y éste pueda ser consultado desde distintos lugares.
4. Soportar las decisiones médico-asistenciales, mediante la interacción con bases de datos, que permitan una rápida consulta de las mejores prácticas, los protocolos de manejo y las evidencias reconocidas.
5. Poner a disposición de los educadores, investigadores y de los planificadores sanitarios esta información, en forma eficiente.

La Tabla 2.1 muestra una comparación de las características de la HC tradicional y la HCC.

Tabla 2.1 Comparación de Historia Clínica Computarizada con la Tradicional

Característica	HC Computarizada	HC Tradicional
Accesibilidad	Utilizable en todo momento y lugar vía internet.	Utilizable en un solo lugar.
Inviolabilidad	No puede ser adulterada, se protege por medio de una firma digital. Inserción de hora y fecha automáticas y técnicas de copia de seguridad adecuadas. Mecanismos de inserción de campos auto-numéricos.	Puede llegar a rehacerse total o parcialmente sin poder comprobarlo.
Privacidad	Garantizada por mecanismos de seguridad informáticos.	Garantizada por mecanismos de control de archivos.
Riesgo de pérdida	Seguridad garantizada con una correcta política de resguardo de la información (back-up).	Frecuentemente extraviada.
Integridad	La informatización racional garantiza que la información de un paciente no esté atomizada.	Frecuentemente se encuentra dividida en servicios, se suelen abrir varios números de historia clínica para un mismo paciente.
Durabilidad	Permanece inalterable en el tiempo para que su información pueda ser consultada.	Sufre deterioro con el tiempo, por su propio uso muchas veces.
Costos de personal	Puede ser operada y buscada por los mismos profesionales que requieren la información.	Requiere personal para el mantenimiento del archivo.

Fuente: [MAN04]

2.2 INGENIERÍA DEL SOFTWARE

A pesar de que cientos de autores han definido en forma individual la ingeniería del software, la definición que propuso Fritz Bauer en una conferencia fundamental sobre la materia es muy atractiva:

“[La ingeniería del software es] el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales.”

Según [PRE06], la ingeniería de software es una tecnología estratificada que debe estar sustentada en un compromiso de calidad. La base que soporta la ingeniería del software es un enfoque en la calidad.

La base de la ingeniería del software es el estrato del proceso. El proceso de la ingeniería del software es el elemento que mantiene juntos los estratos de la tecnología y que permite el desarrollo racional y a tiempo del software de computadora. El proceso define un marco de trabajo que debe establecerse para la entrega efectiva de la tecnología de la ingeniería del software.

Los métodos de la ingeniería del software proporcionan los “como” técnicos para construir software. Los métodos abarcan un amplio espectro de tareas que incluyen la comunicación, el análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y soporte. Los métodos de ingeniería del software se basan en un conjunto de principios básicos que gobiernan cada área de la tecnología e incluye actividades de modelado y otras técnicas descriptivas. [PRE06]

2.2.1 Ingeniería Web

La World Wide Web y la Internet que la alimentan son, posiblemente, los desarrollos más importantes en la historia de la computación. Estas tecnologías han llevado a todos a la era de la informática; además, se han convertido en parte integral de la vida diaria en la primera década del siglo XXI.

Para quienes pueden recordar un mundo sin la Web, el crecimiento caótico de la tecnología tiene su origen en otra era: los primeros días del software. Era una época de poca disciplina pero enorme entusiasmo y creatividad.

En este sentido, la ingeniería Web hace referencia a las metodologías, técnicas y herramientas que se utilizan para el desarrollo de aplicaciones Web complejas y de gran escala en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones. El desarrollo de aplicaciones Web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional. La ingeniería Web es multidisciplinar y aglutina contribuciones de diferentes áreas: arquitectura de la información, ingeniería de hipermedia/hipertexto, ingeniería de requisitos, diseño de interfaz de usuario, usabilidad, ingeniería de datos, indexado y recuperación de información, testeo, modelado y simulación, despliegue de aplicaciones, operación de sistemas y gestión de proyectos.

2.2.2 Crisis del software

De acuerdo con el estándar IEEE-729 [IEEE 729-1983] el software se define como un “conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación”. A partir de esta definición, el software va más allá de los programas de computadora, forman también parte del software los procedimientos, documentación y todos los productos que se relacionen con el desarrollo de un programa computarizado.

El software se ha convertido en un pilar fundamental en la evolución de los sistemas y productos informáticos. En las últimas décadas, el software ha pasado de ser una solución a problemas específicos y una herramienta para el manejo de información, a ser una industria autónoma. Sin embargo, a la par de este crecimiento ha surgido un conjunto de problemas que persisten hasta el día de hoy. La demanda en cantidad y calidad de software ha ido creciendo de forma muy rápida en las últimas décadas, lo que ha desatado un gran auge en la industria software, sin embargo, esta industria aún se encuentra evolucionando de una forma artesanal hacia una disciplina de ingeniería. Esto ha generado problemas tales como: el software es (casi) siempre entregado más tarde de lo previsto, es más caro de lo planeado, y

tiene una funcionalidad distinta a la esperada; estos son los principales factores que dieron pie a la llamada “crisis del software” y que por lo visto, siguen presentándose. [STA08]

De acuerdo con [PRE06] muchas de las causas de la “crisis del software” se deben a mitos y teorías que surgieron durante los primeros años del desarrollo de software. Los mitos del software propagaron información errónea y confusa. La razón que hace peligrosos estos mitos es su origen, ya que la mayoría de estos surgieron debido a las experiencias frente a ciertos eventos, tuvieron un sentido intuitivo y frecuentemente fueron promulgados por expertos de aquel tiempo.

La ingeniería del software (IS) es la medida que surge a raíz de la “crisis del software”, la cual establece la premisa de ofrecer métodos y técnicas para desarrollar y mantener software de calidad.

Por lo tanto, un proceso software es un conjunto de actividades y resultados asociados que producen un producto software, son actividades genéricas que pueden organizarse de diferentes formas y describirse en diferentes niveles de detalle para diferentes tipos de software. [SOM05]

El proceso software define el enfoque que se adopta mientras el software está en desarrollo, en tanto que la IS, además del proceso de software, abarca también las tecnologías que requiere dicho proceso (p.e. métodos, técnicas y herramientas automatizadas). Aunque existen muchos procesos diferentes para el desarrollo de software, éste cuenta con fases genéricas que son comunes para todos ellos:

1. Especificación. Se debe definir la funcionalidad del software y las restricciones en su operación.
2. Diseño e implementación. Se debe producir software que cumpla su especificación.
3. Validación. Se debe validar el software para asegurar que hace lo que el cliente desea.
4. Mantenimiento. Se centra en el cambio que va asociado con la corrección de errores debido a las mejoras producidas por los requisitos cambiantes del cliente.

Para resolver un problema con un desarrollo de software, se debe de contar con una estrategia que acompañe al proceso, métodos, herramientas y las fases genéricas. Estas

estrategias son llamadas modelo del proceso software. Se selecciona un modelo del proceso según la naturaleza del proyecto.

Por otro lado existen diferentes “modelos tradicionales” de desarrollo, entre los más conocidos se encuentran: modelo en cascada, modelo de construcción de prototipos, desarrollo rápido de aplicaciones (DRA), modelo incremental y modelo espiral.

Estos “modelos” son ampliamente utilizados en el área de la IS ya que proporcionan una base para el proceso de desarrollo de software, y además son significativamente más efectivos que un enfoque hecho al azar. Sin embargo, representan debilidades; la razón de esto se debe a que algunos de estos son modificaciones de otros anteriores. Pero estas fallas no dependen solamente del modelo, sino del proyecto al que se apliquen, la elección de un modelo para un proceso software está ligada a la naturaleza del proyecto que se pretende realizar. La Tabla 2.2 nos indica que los principales problemas de los modelos de desarrollo han creado una gran incertidumbre entre los grupos de desarrolladores ya que se crean interrogantes referentes a: qué tipo de proyecto se intenta realizar, qué modelo es el mejor para este tipo de proyecto, y si se obtendrán los resultados esperados con la implantación del modelo elegido.

Tabla 2.2 Principales problemas con los modelos de proceso software tradicionales

Modelo de Proceso de Software	Principales Problemas
Cascada	<ul style="list-style-type: none"> • Proyectos reales raras veces se acoplan al modelo secuencial. • Requiere exponer explícitamente todos los requisitos. • Un cambio en los requisitos iniciales, produce confusión y es difícil de tratarse. • Una versión del programa estará disponible hasta que el trabajo se encuentre muy avanzado, un error grave detectado en esta etapa resulta desastroso para el proyecto.
Construcción de Prototipos	<ul style="list-style-type: none"> • Problemas con sistemas grandes y complejos. • El cliente puede confundirse con la versión final y el prototipo. • La gestión del desarrollo es muy lenta. • El desarrollador elige herramientas para hacer que los prototipos funcionen rápidamente, olvidándose de las características del sistema global.
DRA	<ul style="list-style-type: none"> • Requiere mayor cantidad de recursos humanos. • Este modelo no es aplicable si no existe un compromiso de realizar rápidamente las actividades entre clientes y desarrolladores. • Inadecuado para proyectos con riesgos técnicos altos.
Incremental	<ul style="list-style-type: none"> • Si un avance incremental del producto no cumple las características que el cliente requiere, será trabajo perdido ya que se tiene que volver a desarrollar dicho avance. • Ciclo de vida largo, ya que con la presentación de avances al cliente pueden surgir nuevos requerimientos que no estaban contemplados. • Mayor exigencia del cliente, el nuevo avance tiene que ser mucho mejor que el anterior.
Espiral	<ul style="list-style-type: none"> • Contar con un alto nivel de habilidad para la evaluación de riesgos. • Difícil de comprender y aplicar. • En caso de riesgos no descubiertos a tiempo, se convierten en problemas graves.

Fuente: Elaboración Propia.

2.2.3 DESARROLLO ÁGIL

A principios de la década del '90, surgió un enfoque que fue bastante revolucionario para su momento ya que iba en contra de la creencia de que mediante procesos altamente definidos se iba a lograr obtener software en tiempo, costo y con la requerida calidad. El enfoque fue planteado por primera vez por [Martin91] y se dio a conocer en la comunidad de ingeniería del software con el mismo nombre que su libro, RAD o Rapid Application Development. RAD consistía en un entorno de desarrollo altamente productivo, en el que participaban grupos pequeños de programadores utilizando herramientas que generaban código en forma automática tomando como entradas sintaxis de alto nivel. En general, se considera que este fue uno de los primeros hitos en post de la agilidad en los procesos de desarrollo como mencionaremos a continuación. Cabe mencionar que las metodologías ágiles no inventaron la noción de los procesos iterativos e incrementales, los cuales eran usados desde décadas pasadas inclusive en momentos en que el modelo en cascada era el estándar.

En 2001, Kent Beck y otros 16 notables desarrolladores, escritores y consultores (conocidos como la “Alianza Ágil”) firmaron el “Manifiesto para el desarrollo ágil de software”, estableciendo 12 principios para quienes quieren alcanzar la agilidad.

Los métodos ágiles se centran en el software mismo en vez de su análisis y diseño. Los métodos ágiles dependen de un enfoque iterativo para la especificación, desarrollo y entrega del software, y principalmente fueron diseñados para apoyar al desarrollo de aplicaciones de negocio donde los requerimientos del sistema normalmente cambiaban rápidamente durante el proceso de desarrollo, están pensados para entregar software funcional de forma rápida a los clientes, quienes pueden proponer que se incluyan en iteraciones posteriores del sistema nuevos requerimientos o cambios en los mismos [SOM05]. Los métodos ágiles se basan en la noción de desarrollo y entregas incrementales, pero proponen procesos diferentes para alcanzar sus objetivos. Sin embargo, tienen principios en común que proporcionan las bases para un método ágil (véase Tabla 2.3).

Tabla 2.3 Principios de los Métodos Ágiles.

Principio	Descripción
Participación del cliente	<ul style="list-style-type: none"> • Los clientes deben de estar fuertemente implicados en todo el proceso de desarrollo. Su papel es proporcionar y priorizar nuevos requerimientos del sistema y evaluar las iteraciones del sistema.
Entrega Incremental	<ul style="list-style-type: none"> • El software se desarrolla en incrementos, donde el cliente especifica los requerimientos a incluir en cada incremento.
Personas, no procesos	<ul style="list-style-type: none"> • Se deben reconocer y explotar las habilidades del equipo de desarrollo. Se les debe dejar desarrollar sus propias formas de trabajar, sin procesos formales.
Aceptar el cambio	<ul style="list-style-type: none"> • Se debe contar con que los requerimientos del sistema cambian, por lo que el sistema debe de dar cabida a estos cambios.
Mantener la simplicidad	<ul style="list-style-type: none"> • Se deben de centrar en la simplicidad tanto en el software a desarrollar como en el proceso de desarrollo. Donde sea posible, se trabaja activamente para eliminar la complejidad del sistema.

Fuente: Elaboración Propia

2.2.4 Principales Modelos de Gestión Ágil

Si hubiera que determinar cuál es el origen de la gestión ágil de proyectos, a falta de mejor información, habría que situarlo en las prácticas adoptadas en los 80 por empresas como Honda, 3M, Canon, Fuji, Nec, Xerox, HP o Epson para el desarrollo de nuevos productos.

La industria del software ha sido la primera en seguir su adopción, y muchos de sus profesionales han documentado y propagado las formas particulares en las que han implementado los principios de la agilidad en sus equipos de trabajo.

De esta forma han aparecido en las últimas décadas los nombres:

- AD - Agile Database Techniques
- AM - Agile Modeling
- ASD - Adaptive Software Development
- AUP - Agile Unified Process

- Crystal
- FDD - Feature Driven Development
- DSDM - Dynamic Systems Development Method
- Lean Software Development
- Scrum
- TDD - Test-Driven Design
- XP - eXtreme Programming

Éstos son los modelos que se encuentran inscritos en la organización Agile Alliance (www.agilealliance.org) para promocionar y difundir su conocimiento.

Cada una de ellos expone formas concretas de aplicación de principios ágiles en el desarrollo de software.

Algunos determinan cómo realizar las pruebas, o la duración que emplean para desarrollar cada iteración, o el protocolo para realizar las reuniones de trabajo.

Unos métodos cubren áreas concretas de la ingeniería del software (diseño, desarrollo y pruebas), como es caso de AD, AM o XP, y otros se centran en la gestión del proyecto.

Éstos últimos son:

- ASD - Adaptive Software Development
- AUP - Agile Unified Process
- Crystal
- DSDM - Dynamic Systems Development Method
- Scrum

2.2.5 Programación Extrema

La programación extrema (PE) es el método ágil más conocido y ampliamente utilizado. Debe su nombre al uso de prácticas reconocidas, como el desarrollo iterativo, y a la participación del cliente en niveles “extremos”. En la PE, todos los requerimientos se expresan como escenarios, los cuales se implementan directamente como una serie de tareas. Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el nuevo código se integre al sistema. Existe un pequeño espacio de tiempo entre las entregas del sistema. La PE

implica varias prácticas (véase Figura 2.1), que se ajustan a los principios de los métodos ágiles.

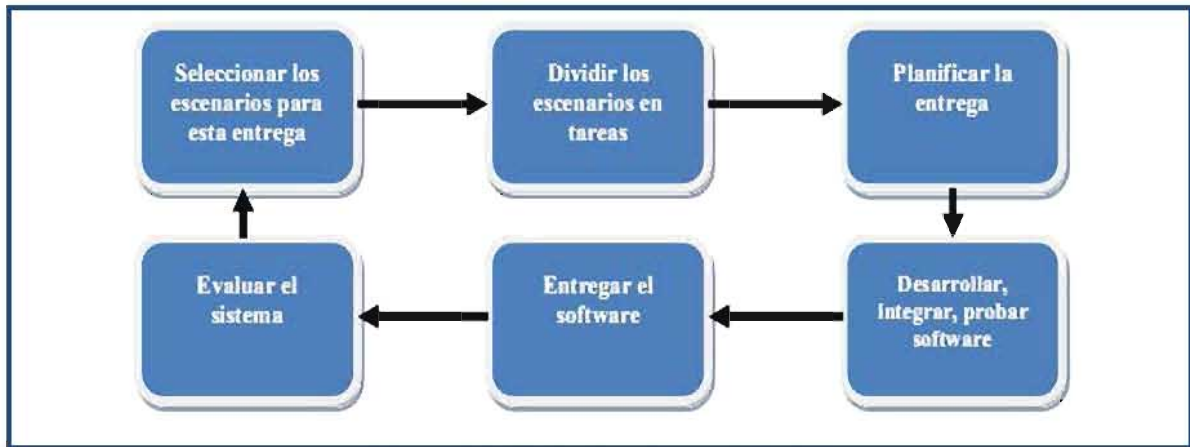


Figura 2.1 Ciclo de entrega de la PE

Fuente: [URL-FIG-1]

La PE utiliza un enfoque orientado a objetos como su paradigma de desarrollo preferido. La PE abarca un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades del marco de trabajo: planeación, diseño, codificación y pruebas. En la Figura 2.2 se ilustra el proceso de la PE y se observa algunas de las ideas y tareas clave asociadas con cada actividad del marco de trabajo. [BEC1b]

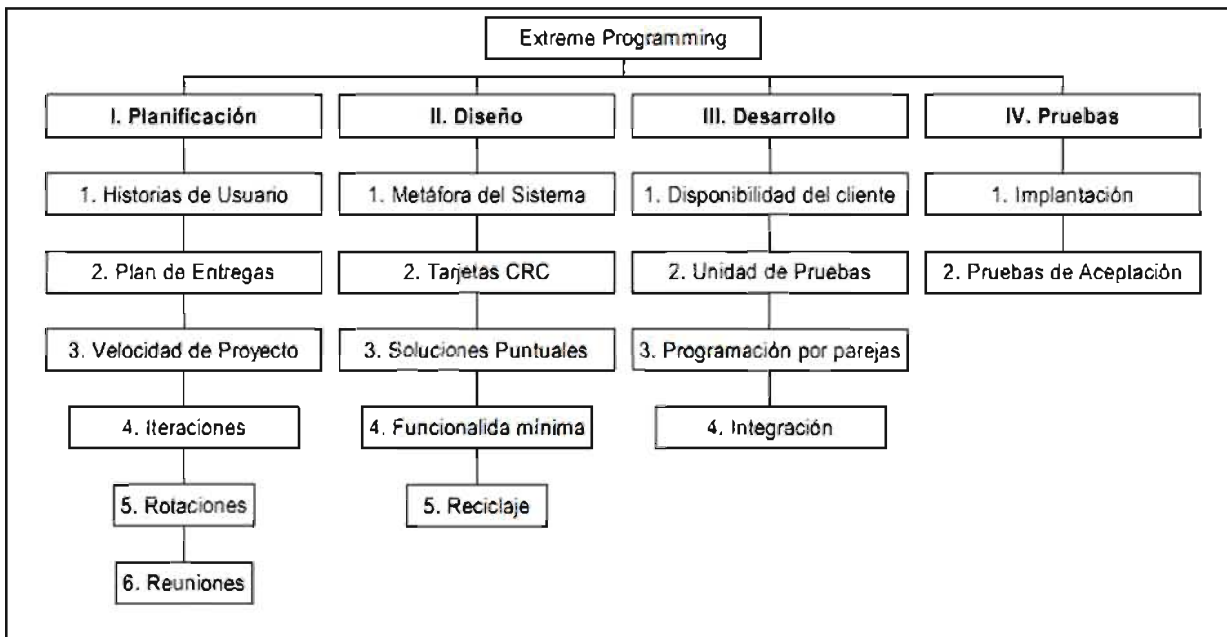


Figura 2.2 El proceso de la Programación Extrema

Fuente: [URL-FIG-1]

Sin embargo, ninguna de las soluciones mencionadas anteriormente ha sido capaz por sí sola de proporcionar alivio a la “crisis del software”, ya que si bien el foco de interés de los esfuerzos se ha situado en diversas áreas, con frecuencia se ha dirigido a la búsqueda de soluciones óptimas a problemas particulares. Muestra de lo anterior se refleja en los estudios realizados por el Standish Group [STA04], el informe Chaos Extreme¹⁷, establece que del año 2000 al 2006 el abandono de proyectos software y los proyectos software que no cumplían con los requisitos del usuario siguen siendo problemas latentes y redundantes dentro de la industria software, también indica el aumento paulatino del porcentaje de proyectos software terminados (véase Figura 2.2).

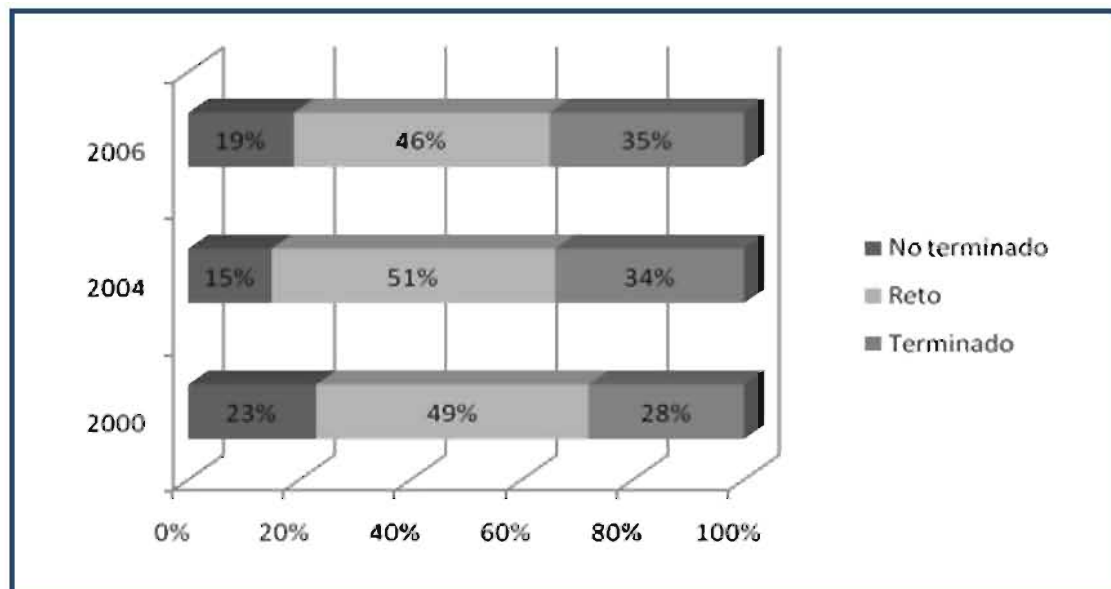


Figura 2.3 Resultados del estudio CHAOS (2000-2006)

Fuente: [STA04]

El mismo reporte indica que el 35% de los proyectos software fueron terminados, el 19% no fueron terminados y un 46% no cumplió con los requisitos establecidos por el cliente.

¹⁷ Desde 1994 el reporte del Standish Group, el famoso Chaos Report, se ha vuelto en el patrón dorado sobre la calidad de los proyecto software. Estos estudios demuestran la situación de Europa y Estados Unidos en relación al éxito y/o fracaso de los proyectos informáticos. Para más información consultar www.standishgroup.com.

Estos puntos concretos establecieron la adopción de las mejores prácticas del software, ampliando la gama de ventajas, haciendo un mejor uso de los métodos y tecnologías actualmente disponibles en el desarrollo software. Los modelos de mejora buscan reconocer las oportunidades y establecer un programa de mejoramiento mediante la comprensión y el conocimiento de los costos y beneficios que conlleva una implementación de estas características. Un modelo de procesos de referencia describe cuáles son reconocidas como las mejores prácticas que una organización debe de implementar para el desarrollo de software.

De acuerdo a [WAL03] “las mejores prácticas son todas aquellas metodologías y herramientas que mejoran consistentemente la productividad y calidad de los productos software cuando son implementadas, y al aplicarlas se tiene un beneficio directo en el proyecto”. Por lo tanto, las mejores prácticas de software son aquellas prácticas que los ingenieros de software implementan con una experiencia reconocida en el área específica, esta contribución se hace significativa para el buen término del proyecto software [ADA04].

2.3 REINGENIERÍA

Según [ART9], Reingeniería es: “la modificación de un producto de software, o de ciertos componentes, usando para el análisis del sistema. Existen técnicas de ingeniería inversa y, para la etapa de reconstrucción, herramientas de ingeniería directa”.

La reingeniería trabaja todos los niveles de abstracción (desde la implementación hasta el diseño) para conseguir realizar cambios más drásticos preservando los valores de negocio del sistema.

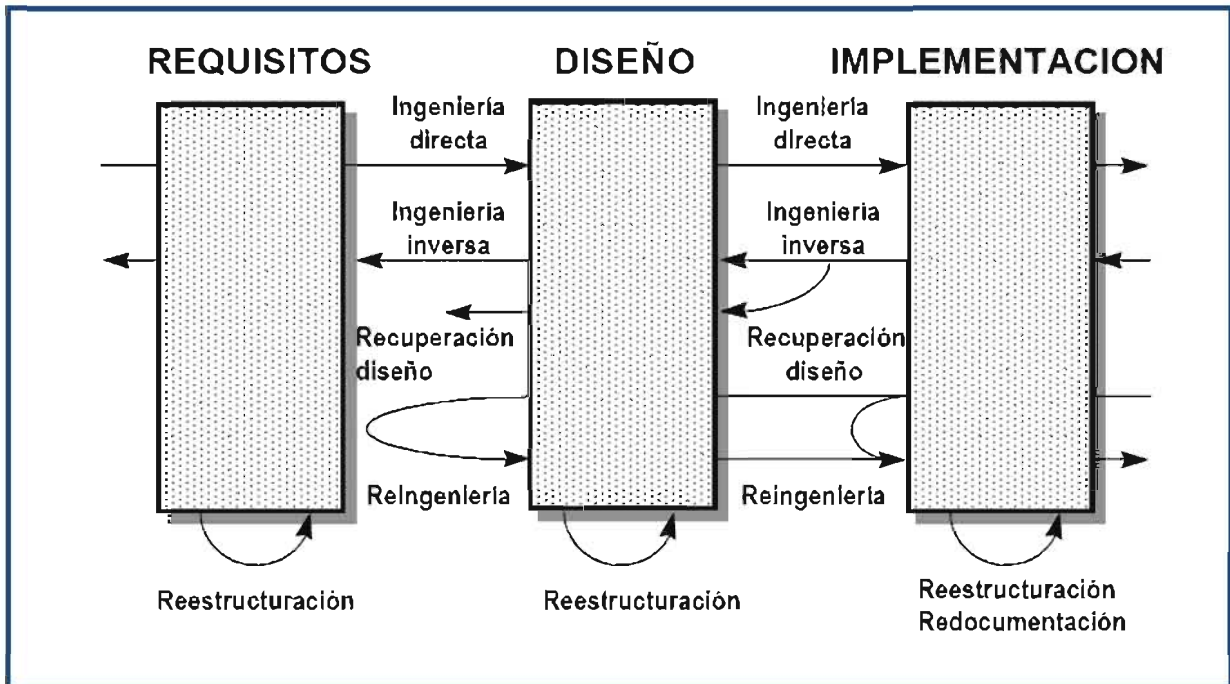


Figura 2.4 Reingeniería

Fuente: [ART13]

2.3.1 Ingeniería Inversa

El término ingeniería inversa tiene sus orígenes en el mundo del hardware. Una compañía desensambla un producto de hardware de un competidor con la finalidad de comprender sus “secretos” de diseño y fabricación. Tales secretos podrían comprenderse fácilmente si se obtuviesen las especificaciones de diseño y fabricación del competidor. Pero dichos documentos están patentados y no están disponibles para la compañía que realiza ingeniería inversa. [PRE06]

La ingeniería inversa del software es el proceso de analizar un programa con la finalidad de crear una representación del programa en un mayor grado de abstracción que el código fuente. La ingeniería inversa es un proceso de recuperación de diseño. Las herramientas de la ingeniería inversa obtienen información del diseño de datos, arquitectónico y de procedimientos a partir de un programa existente. [PRE06]

2.3.2 Ingeniería Directa

En un mundo ideal, las aplicaciones se reconstruirían empleando un motor de reingeniería automatizado, el programa antiguo sería insertado en el motor, analizado, reestructurado y luego regenerado en una forma que exhibiese los mejores aspectos de la calidad de software. A corto plazo es improbable que tal motor aparezca, pero las empresas han introducido herramientas que mejoran un limitado subconjunto de dichas capacidades que aborden dominios de aplicación específicos (por ejemplo las aplicaciones que se implementan mediante un sistema de base de datos específico).

La ingeniería directa, también llamada renovación o reclamación, no solo recupera la información de diseño a partir del software existente con la finalidad de mejorar su calidad global. En la mayoría de los casos el software sometido a reingeniería vuelve a implementar la función del sistema existente y también añade nuevas funciones o mejora al desempeño global. [PRE06]

2.4 LENGUAJE UNIFICADO DE MODELADO

Cualquier rama de ingeniería o arquitectura ha encontrado útil desde hace mucho tiempo la representación de los diseños de forma gráfica. Desde los inicios de la informática se han estado utilizando distintas formas de representar los diseños de una forma más bien personal o con algún modelo gráfico. La falta de estandarización en la manera de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores.

Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño. El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con

los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.

2.4.1 ¿Qué es UML?

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema [HER01].

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware [HER01].

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

2.4.2 Diagramas UML

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas [BOO00]:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.

Los diagramas más interesantes (y los más usados) son los de casos de uso, clases y secuencia, por lo que nos centraremos en éstos.

- **Diagrama de casos de usos** representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. En la Figura 2.4 se muestra un ejemplo de casos de uso.
- **Diagrama de clases** muestra un conjunto de clases, interfaces y sus relaciones. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. En la Figura 2.5 se muestran las clases globales, sus atributos y las relaciones de una posible solución al problema de la venta de entradas.

- **Diagrama de secuencia** se muestra la interacción de los objetos que componen un sistema de forma temporal. Siguiendo el ejemplo de venta de entradas, la Figura 2.6 muestra la interacción de crear una nueva sala para un espectáculo.

El resto de diagramas muestran distintos aspectos del sistema a modelar. Para modelar el comportamiento dinámico del sistema están los de interacción, colaboración, estados y actividades. Los diagramas de componentes y despliegue están enfocados a la implementación del sistema.

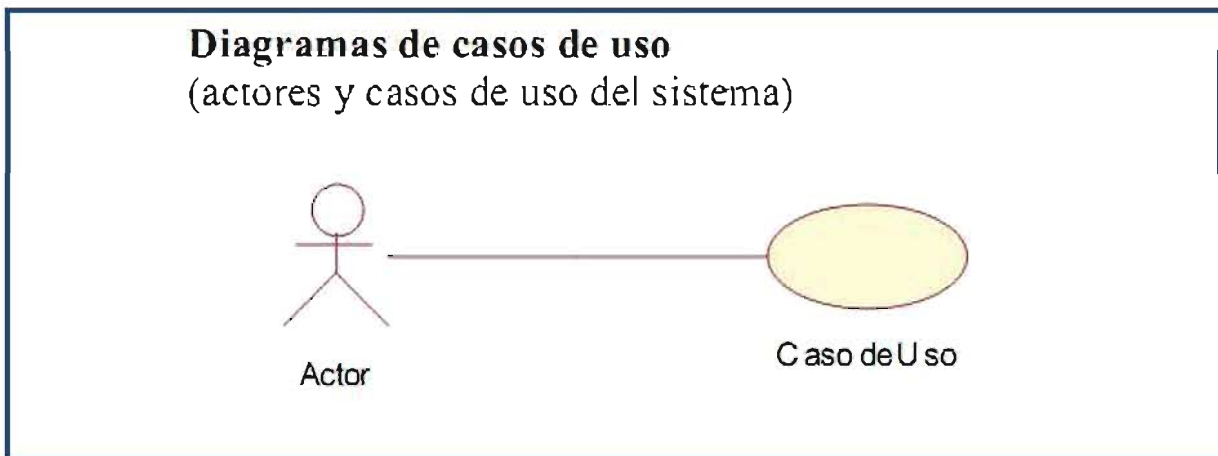


Figura 2.5 Casos de Uso

Fuente: [HER01]

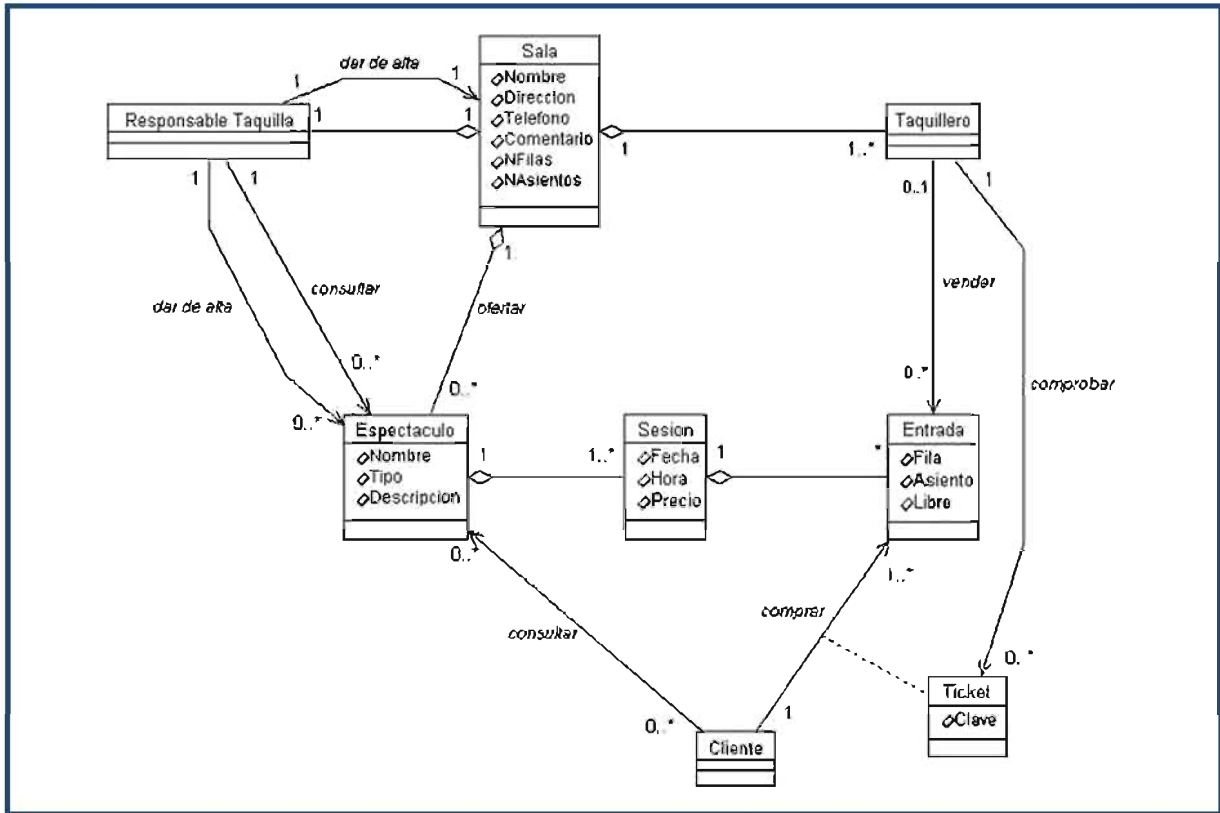


Figura 2.6 Diagrama de Clases

Fuente: [HER01]

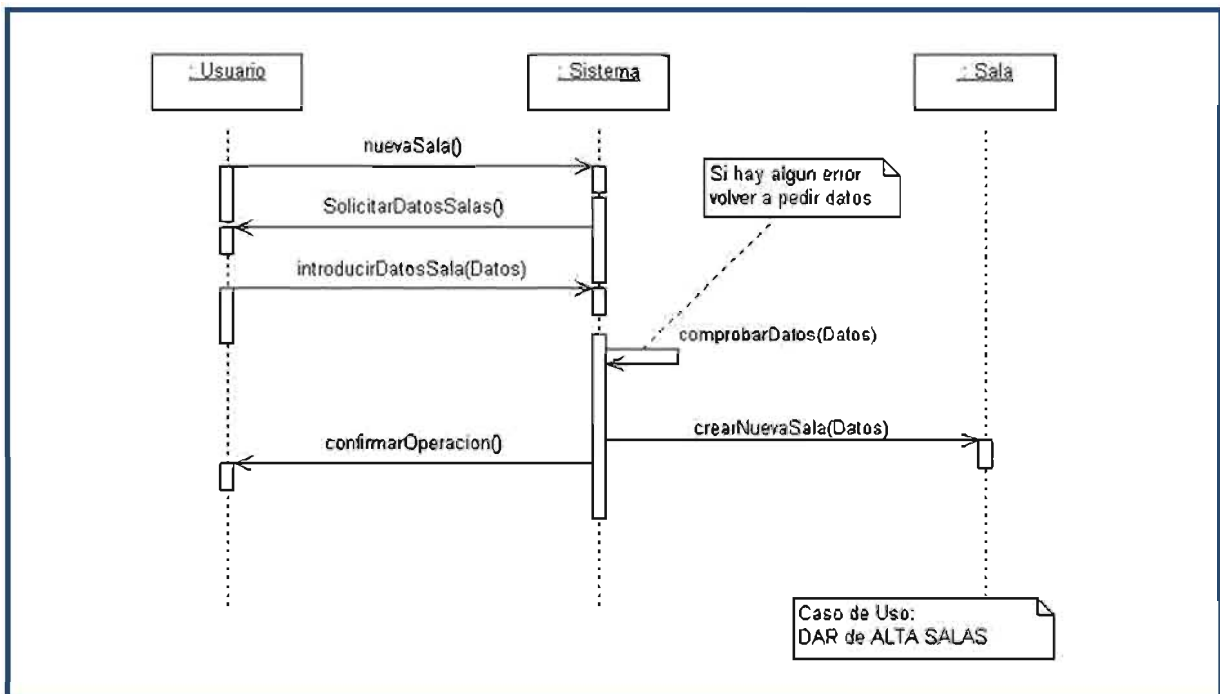


Figura 2.7 Diagrama de Secuencia

Fuente: [HER01]

2.5 UWE (UML-Based Web Engineering)

UWE (Ingeniería Web Basada en UML, de sus siglas en inglés UML-based Web Engineering) apareció a inicios de los 90s, con la idea de encontrar un estándar para construir modelos de análisis y diseño basados en los métodos de OOHDM (Método de Diseño Hipermedia Orientado a Objeto, de sus siglas en inglés Object Oriented Hypermedia Design Method) y WSDM (Gestión de Servicios Web Distribuidos, de sus siglas en inglés Web Services Distributed Management). UWE, al igual que RUP, es un acercamiento orientado a objetos que cubre en su totalidad el ciclo de vida de las aplicaciones Web, se mueve a través de una serie de iteraciones e incrementos y usa notaciones y diagramas de UML. Se utiliza diagramas de actividades UML para visualizar las actividades de los flujos de trabajos y los productos desarrollados por los trabajadores.

Resumiendo, UWE es un método sistemático, prescriptivo, enfocado en el usuario, basado en UML, iterativo e incremental para sistemas hipermedia adaptativos [KOC07]. Entiéndase que nos referimos a un sistema hipermedia como una aplicación la cual utiliza relación asociativa entre la información contenida dentro de múltiples medios, con el propósito de facilitar tanto el acceso como la manipulación de la información encapsulada.

Comenzando con un análisis de requisitos realizado bajo la técnica de casos de uso que se centra en la fase de diseño, el modelo conceptual de la aplicación es usado como la pauta para modelar el modelo de navegación. El modelo de navegación es el siguiente en realizarse, el cual muestra cómo navegar por el espacio de navegación de la herramienta utilizando elementos de acceso como índices, consultas y menús. Por último, un modelo de presentación es construido basado en el modelo de navegación [KOC07].

- Modelo de Casos de Uso.
- Modelo Conceptual.
- Modelo de Navegación.
- Modelo de Presentación.

La Figura 2.7 muestra los modelos representados como paquetes UML y mostrando la relación que existe entre cada uno de ellos.

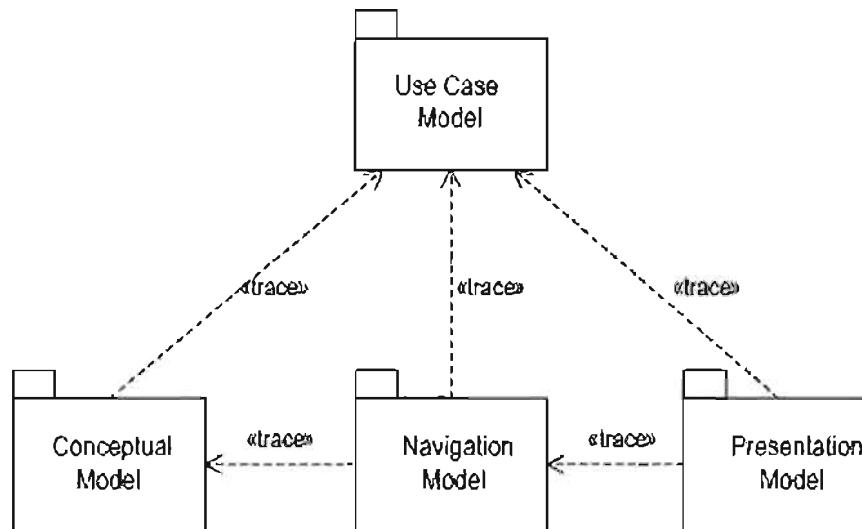


Figura 2.8 Modelos utilizados en el método de creación UWE.

Fuente: [KOC07]

El objetivo del análisis de requisitos es encontrar los requerimientos funcionales de la aplicación Web y representar estos requisitos mediante un diagrama de casos de uso.

El objetivo del diseño conceptual es construir un modelo conceptual del dominio de la aplicación, teniendo en cuenta los requisitos capturados con los casos de uso. Las técnicas orientadas a objetos tradicionales se utilizan para construir el modelo conceptual, como la creación de clases, asociaciones y la definición de estructuras de herencia. El modelo se representa mediante un diagrama UML ordinario de clases.

Basado en el modelo conceptual el método de navegación propone un conjunto de directrices para construir un modelo de navegación que representa el espacio de navegación y la estructura de navegación mediante la adición de elementos de acceso que pueden ser utilizados para la navegación. El método incluye un conjunto de elementos de modelado UML estereotipados para el diseño de navegación, como índices, visitas guiadas, consultas a la base de datos y menús. Estos estereotipos se utilizan para representar el modelo de navegación de una forma más clara y sencilla.

El modelo de presentación tiene como objetivo el diseño de interfaces de usuario abstractas y el diseño de la interacción del usuario con la aplicación Web. [ROS03]

2.5.1 UWE y su relación con UML

UWE define una extensión del Lenguaje Unificado de Modelado (UML). Ésta, es considerada como una extensión ligera de peso e incluye en su definición tipos, etiquetas de valores y restricciones para las características específicas del diseño Web, las cuales, unidas a las definiciones de UML forman el conjuntos de objetos de modelado que se usarán para el desarrollo del modelo utilizado en UWE.

Las funcionalidades que cubren UWE abarcan áreas relacionadas con la Web como la navegación, presentación, los procesos de negocio y los aspectos de adaptación.

Una de las ventajas de que UWE extienda el estándar UML es la flexibilidad de éste para la definición de un lenguaje de modelado específico para el dominio web y sobretodo la aceptación universal de dicho estándar en el campo de la ingeniería del software.

Otra gran ventaja es que actualmente existen múltiples herramientas CASE basadas en UML, con lo cual es relativamente sencillo su utilización y ampliación para utilizar los objetos de modelado definidos en UWE.

2.5.2 Modelos de UWE

2.5.2.1 Modelo de casos de uso

El método de creación para UWE propone los casos de uso para la captura de requisitos del sistema. Es una técnica centrada en el usuario que obliga a definir quienes son los usuarios (actores) de la aplicación y ofrece una forma intuitiva para representar las funcionalidades que tiene que cumplir una aplicación para cada actor.

El método de creación aplica los pasos sugeridos por muchos autores para la construcción del modelo de casos de uso de una aplicación Web. Estos pasos son:

1. Establecer los actores.
2. Para cada actor describa las actividades que tendrá que realizar.
3. Agrupar las actividades por casos de uso.
4. Establecer relaciones entre actores y casos de uso.
5. Establecer relaciones “include” y “extends” entre casos de uso.

6. Simplificar el modelo de casos de uso mediante la definición de relaciones de herencia entre los actores y/o entre casos de uso.

En el diagrama mostrado en la Figura 2.8 se propone un ejemplo simple de caso de uso, para ilustrar las funcionalidades de una aplicación web.

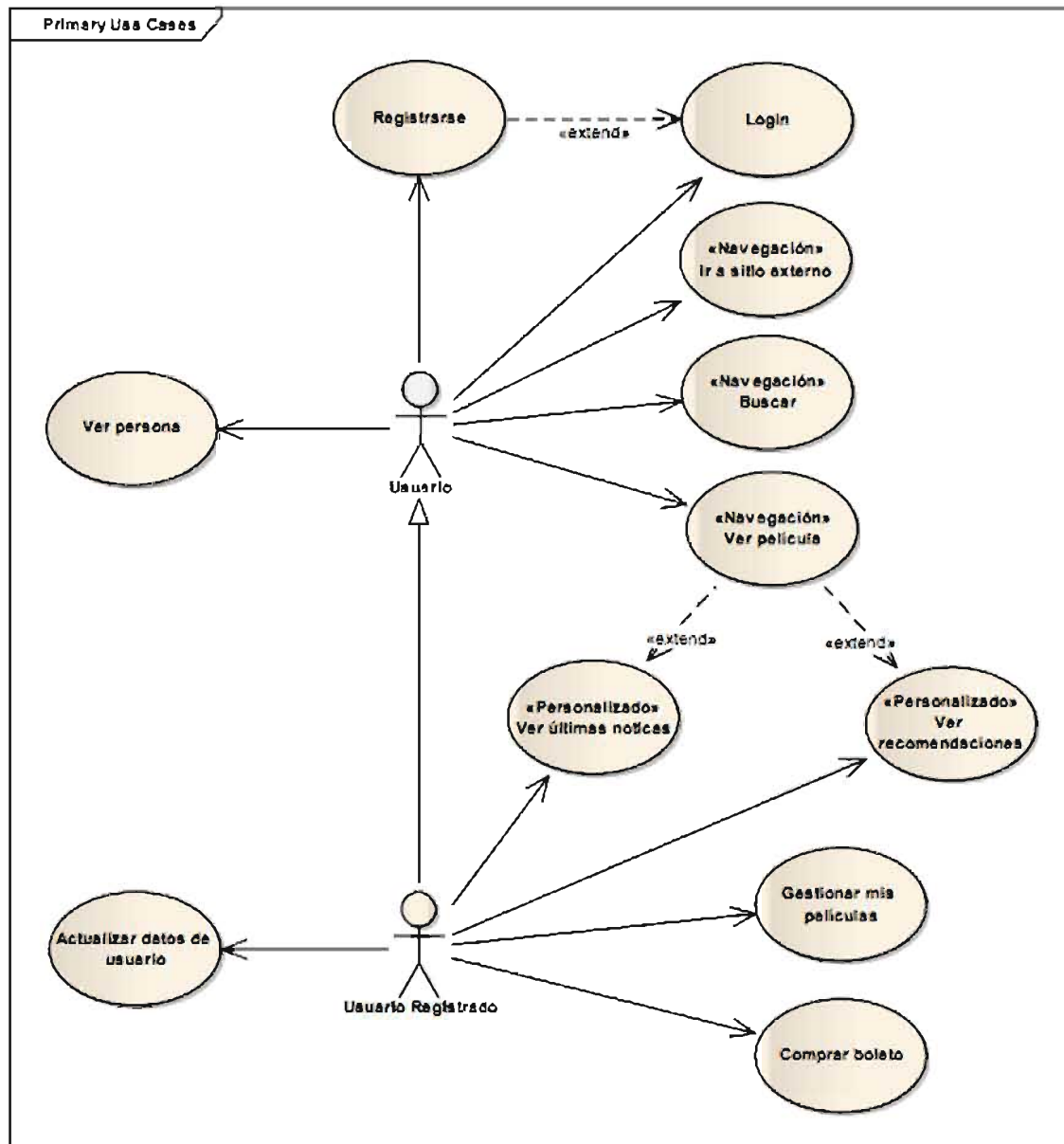


Figura 2.9 Diagrama de Casos de Uso

Fuente: [ROS08]

2.5.2.2 Modelo Conceptual

El diseño conceptual se basa en el análisis de requisitos de la etapa anterior. Incluye los objetos implicados en la interacción entre el usuario y la aplicación. El diseño conceptual tiene como objetivo construir el modelo de clases con estos objetos, ignorando las rutas de navegación, presentación y aspectos de la interacción en lo posible. Estos aspectos se posponen a los pasos de navegación y presentación del proceso de creación UWE [KOC07].

El diagrama de contenido se refiere al modelo conceptual de la herramienta. Un diagrama de clases en UML se utiliza para representar gráficamente un modelo conceptual como visión estática que demuestre una colección de los elementos estáticos del dominio [OCA04]. Un diagrama de contenido proporciona una especificación del dominio de la información relevante para el software Web. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la herramienta, es decir, los objetos que son relevantes para la realización de una actividad o que son el resultado de una de ellas.

El desarrollador puede seguir técnicas reconocidas de modelado orientado a objetos para construir el modelo de clases UML que representen el diagrama de contenido, como:

1. Búsqueda de clases del sistema.
2. Especificación de los atributos más importantes.
3. Determinación de las asociaciones entre clases.
4. Identificar agregación y composición de clases.
5. Definir herencia entre clases.
6. Definir condiciones.

La Figura 2.9 muestra un ejemplo de este modelo.

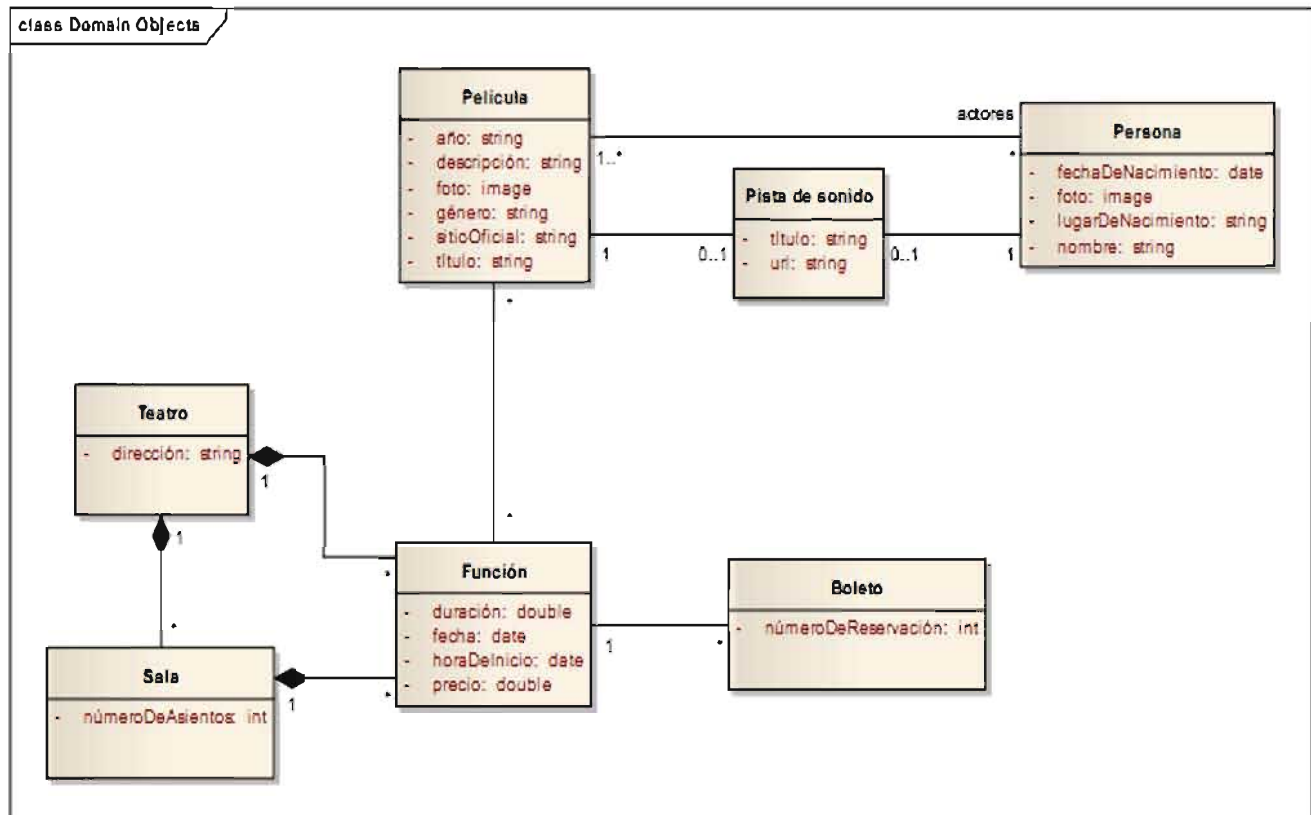


Figura 2.10 Diagrama de Contenido

Fuente: [ROS08]

2.5.2.3 Modelo de navegación

Basados en los requisitos y en el modelo de contenido, el modelo de navegación de la aplicación Web se construye para especificar la estructura del hipertexto del sistema, que se da por nodos y enlaces. En este modelo se describe cómo es la navegación con el apoyo de elementos de acceso, como índices, visitas guiadas, consultas a la base de datos y menús [KOC07].

Este modelo comprende la especificación de qué objetos pueden ser visitados mediante la navegación a través de la aplicación Web y las asociaciones entre ellos. En el marco de UWE se destaca como el más importante, pues con él se pueden representar elementos estáticos, a la vez que se pueden incorporar lineamientos semánticos de referencia para las funcionalidades dinámicas de una aplicación Web [OCA04].

En la Figura 2.10 se muestra el modelo de navegación basado en el modelo de casos de uso y modelo de contenido de la Figura 2.8 y Figura 2.9.

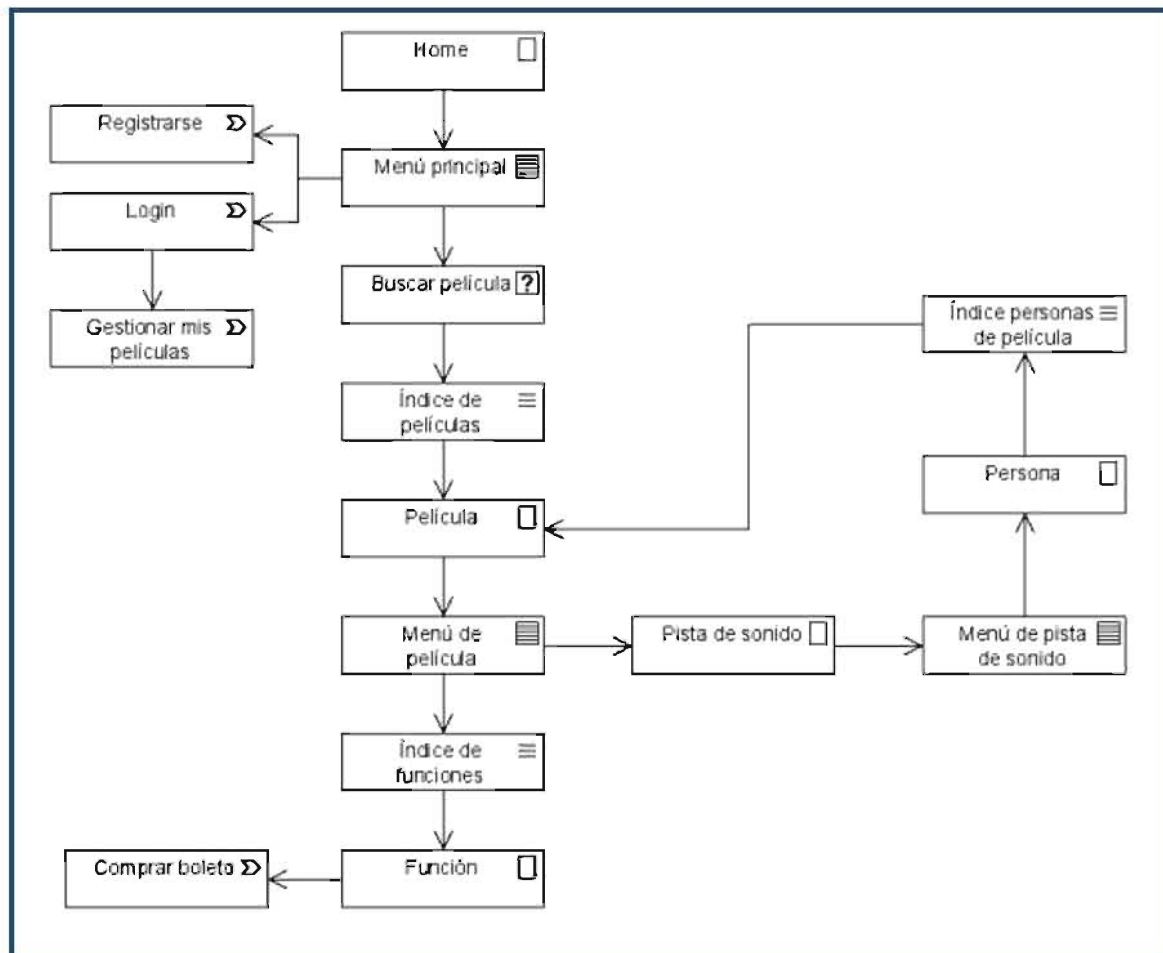


Figura 2.11 Diagrama de Navegación

Fuente: [ROS08]

2.5.2.4 Modelo de Presentación

El modelo de presentación pretende proporcionar una representación abstracta de la interfaz de usuario final y definir la interacción de las clases navegables con el usuario. Esta etapa está basada en el modelo de navegación. El objetivo principal de esta capa es conseguir un modelo abstracto de la interfaz de usuario Web 2.0, que permita mostrar la forma en que la estructura de navegación es presentada al usuario. El modelo de interfaz es un diseño a grandes rasgos de lo que será la interfaz final, decisiones acerca de detalles como

tamaños, colores, fuentes u otros objetos de interfaz específicos no se consideran por ser parte de la implementación. El objetivo es visualizar la organización de la estructura de la aplicación Web de una manera más intuitiva que como lo hace el modelo de navegación [KOC07].

La Figura 2.11 muestra un modelo abstracto de presentación de la interfaz de usuario en base al modelo de casos de uso de la Figura 2.8.



Figura 2.12 Modelado Abstracto de la interfaz de usuario

Fuente: [ROS08]

2.6 RICH INTERNET APPLICATIONS (RIA)

En la actualidad, las aplicaciones Web no solo deben de ofrecer variedad de funciones, sino que además deben de ser lo más óptimas posibles y sus interfaces gráficas deben ser mucho más impactantes y cómodas para los usuarios [Farré05]. En los últimos años, la demanda de aplicaciones basadas en Internet continúa creciendo y, en general, es bastante diferente a la demanda que se presentaba en la década de los noventa. Los usuarios finales y las empresas ahora exigen más de sus inversiones en tecnología Internet. La capacidad para distribuir valores reales a los usuarios está obligando a muchas empresas a buscar modelos de aplicaciones de Internet más ricas; modelos que combinan la potencia de las computadoras de sobremesa tradicionales, ricas en dispositivos, con la utilización y la naturaleza rica en contenido de las aplicaciones Web [Tapper09], [Davis08].

En Marzo de 2002 Macromedia acuñó el término Aplicaciones Ricas de Internet (RIA, Rich Internet Applications) las cuales son un nuevo tipo de aplicación Web cuyo objetivo es el incrementar y mejorar las opciones y capacidades de las aplicaciones Web tradicionales [Tapper09], [Farré05]. De acuerdo a [URL-1], las RIA ofrecen una experiencia sofisticada y atractiva que mejora la satisfacción del usuario y aumenta su productividad. Gracias al amplio alcance de Internet, las RIA pueden implementarse en navegadores, escritorios y dispositivos.

Las limitaciones en la capa de presentación de los actuales navegadores Web y del lenguaje HTML han motivado a los desarrolladores a utilizar las aplicaciones tipo RIA que permiten, entre otras cosas, mejorar la experiencia entre el usuario y la aplicación, la ejecución de contenido multimedia y la carga de aplicaciones online/offline, dependiendo de la tecnología que se utilice [Farré05]. La idea de las RIA es poder realizar programas con las funcionalidades de una aplicación de escritorio común, con la gran ventaja de que no es necesario que el usuario tenga la aplicación instalada en su PC, sino que sean accesibles desde cualquier navegador Web, con lo cual se obtiene una aplicación multiplataforma que funciona en cualquier computadora con un navegador Web y conexión a Internet (véase Figura 2.14).

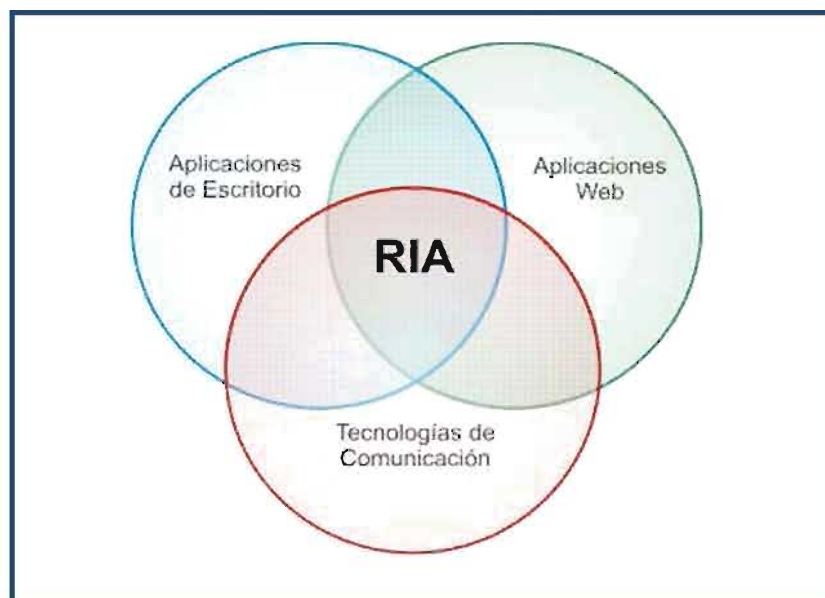


Figura 2.13 Idea principal de un enfoque RIA.

Fuente: [Farré05]

Las RIA son aplicaciones Web que transfieren la mayor parte de la carga de procesamiento de la interfaz de usuario para el cliente Web, mientras que la parte predominante de los datos (desde el control y el mantenimiento a los datos) permanece en el servidor de aplicaciones [Martinez-Ruiz06]. La Figura 2.15 muestra una arquitectura RIA estándar.

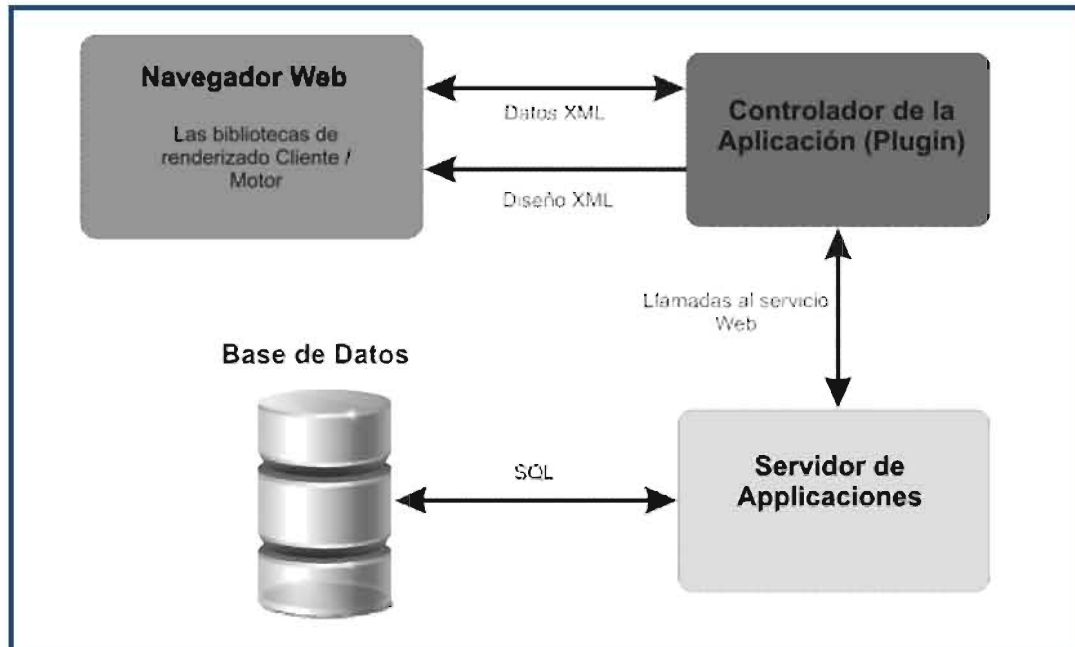


Figura 2.14 Arquitectura típica de RIA.

Fuente: [Martinez-Ruiz06]

Una de las principales ideas de este tipo de aplicaciones es el mejorar el método en que se envía y recibe el flujo de información entre el cliente y el servidor, las aplicaciones Web tradicionales cargan todo el procesamiento de datos en el servidor, recargando toda la página con cualquier acción que haga el usuario, aún si el cambio realizado es mínimo; mientras que bajo un enfoque RIA sólo es necesario cargar la interfaz solo una vez, y para cada acción del usuario se realiza una petición específica de la información solicitada al servidor sin la necesidad de recargar toda la página. De esta manera, se consigue reducir el tiempo de espera del usuario, el ancho de banda consumido y la carga de trabajo para el servidor [Tapper09], [Pérez08], [Davis08], [Brown08]. En la actualidad existen muchas opciones tecnológicas para construir una aplicación tipo RIA. Las opciones más populares son las iniciativas basadas en HTML como AJAX (Asynchronous Javascript and XML, Javascript y XML Asíncronos) [URL-2], y las opciones basadas en plugin como Adobe Flash [URL-3],

Adobe Flex [URL-4] y otros que se ejecutan en Flash Player. También existen nuevas tecnologías de Microsoft, como Windows Presentation Foundation (WPF) [URL-5] y Silverlight [URL-6].

2.7 CALIDAD DEL PRODUCTO SOFTWARE

En la actualidad, la comunidad de Ingeniería del Software reconoce que las aplicaciones y/o producto Web poseen características que las diferencian de los sistemas tradicionales. Estas características se deben al tamaño y complejidad de las aplicaciones, el carácter multidisciplinar del equipo de desarrollo, la tasa apresurada de entrega del proyecto, etc. Éstas características hacen con que los procesos, modelos y métricas existentes para evaluar la calidad tengan que ser adaptados para considerar los cambios impuestos por las nuevas tecnologías.

La norma **ISO 8402** define la calidad como el conjunto de propiedades y características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas. [URL-8]

Podemos considerar a la calidad Web como la capacidad del producto software para permitir que usuarios específicos logren realizar tareas específicas con productividad, efectividad, seguridad y satisfacción, en determinados escenarios de uso. El objetivo de un producto es que posea la calidad necesaria y suficiente para que satisfaga las necesidades de usuario explícitas e implícitas.

La calidad del producto software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad del producto software es medible y varía de un sistema a otro o de un programa a otro.

Observamos que la calidad es sinónimo de unos factores que pueden ser medidos:

- Eficiencia
- Flexibilidad
- Facilidad de uso
- Portabilidad
- Seguridad, etc.

2.7.1 ISO 9126

La **ISO 9126** [basada en el modelo de Mc Call] plantea un modelo normalizado que permite evaluar y comparar productos sobre la misma base. [URL-8]

Aquí la calidad queda definida a un alto nivel de abstracción por seis características:

- **Funcionalidad:** Las funciones satisfacen necesidades declaradas o implícitas.
- **Fiabilidad:** Capacidad de un sistema para mantener su nivel de rendimiento.
- **Usabilidad:** Esfuerzo necesario para el uso y la valoración individual de tal uso, por parte de un conjunto de usuarios.
- **Portabilidad:** Es la capacidad de un sistema para ser transferido de un entorno a otro.
- **Mantenibilidad:** Es el esfuerzo necesario para realizar modificaciones específicas.
- **Eficiencia:** Es la relación entre el nivel de prestaciones de un sistema y el volumen de recursos utilizados en condiciones declaradas.

2.7.2 Web QEM

La evaluación de productos Web no es una tarea sencilla. Es difícil considerar todas las características y atributos deseables y obligatorios de una aplicación o sitio Web si no se cuenta con un modelo de calidad que permita a los evaluadores especificar ordenadamente dichas características y atributos. La metodología Web QEM (Quality Evaluation Method), parte de un modelo jerárquico de calidad de producto basado en el estándar ISO 9126. Es decir, la calidad de producto queda definida a un alto nivel de abstracción por las características denominadas usabilidad, funcionalidad, fiabilidad, eficiencia, mantenibilidad y portabilidad (ver Tabla 3). Luego, se emplea un proceso de descomposición recursivo basado en sub-características y atributos, en consideración de una meta de evaluación y perfil de usuario dados. [URL-9]

Tabla 2.4 Calidad Aplicación Web

Requisito de Calidad	Características
Usabilidad	<ul style="list-style-type: none"> • Capacidad de comprensión del sitio global • Servicios de ayuda y realimentación en línea • Capacidades estéticas y de interfaz • Servicios especiales
Funcionalidad	<ul style="list-style-type: none"> • Capacidad de recuperación y de búsqueda • Servicios de búsqueda y navegación • Servicios relacionados con el dominio de la aplicación
Fiabilidad	<ul style="list-style-type: none"> • Proceso correcto de enlace • Recuperación de errores • Validación y recuperación de la entrada del usuario
Eficiencia	<ul style="list-style-type: none"> • Rendimiento del tiempo de respuesta • Velocidad de generación de páginas • Velocidad de generación de gráficos
Mantenimiento	<ul style="list-style-type: none"> • Facilidad de corrección • Adaptabilidad • Extensibilidad

Fuente: Elaboración Propia

2.7.3 Usabilidad

A la hora de calificar la calidad de una aplicación web, uno de los factores más importantes es la usabilidad. Es el atributo más visible ya que determina el grado de satisfacción del usuario respecto de la aplicación web y si no es fácil de usar no será bien empleado por el usuario y por lo tanto, nunca podrá ser aprovechado en su totalidad. [URL-10]

La Usabilidad determina hasta qué punto una aplicación web puede ser empleado por los usuarios, de tal manera que los apoye en el desarrollo de sus actividades y en el logro de sus metas.

2.7.4 Principios o reglas de usabilidad

Según Jakob Nielsen¹⁸ existen 8 reglas básicas sobre la usabilidad en la Web, a saber:

1. En Internet el usuario es el que manda.
2. En Internet la calidad se basa en la rapidez y la fiabilidad. En Internet cuenta que una página sea más rápida que bonita, fiable que moderna, sencilla que compleja.
3. Seguridad.
4. La confianza es algo que cuesta mucho ganar y se pierde con un mal enlace.
5. Simplificar, reducir, optimizar. La gente no va aprender un sitio por mucho que se insista, así que debe ser sencillo, reutilizable en todos los elementos que se pueda, para que de este modo los usuarios se sientan cómodos y no se pierdan cada vez que necesiten encontrar algo en el sitio.
6. Poner las conclusiones al principio. El usuario se sentirá más cómodo si ve las metas al principio. De esta forma no tendrá que buscar lo que necesita y perderá menos tiempo en completar su tarea. Si completa su tarea en menos tiempo se sentirá cómodo y quizás se dedique a explorar el sitio o quizás se lo recomiende a un amigo.
7. No hacer perder el tiempo a la gente con cosas que no necesitan.
8. Buenos contenidos. Escribir bien para Internet es todo un arte. Pero siguiendo las reglas básicas de (1) poner las conclusiones al principio y (2) escribir como un 25% de lo que se pondría en un papel, se puede llegar muy lejos. Leer en pantalla cuesta mucho, por lo que, en el caso de textos para Internet, reducir y simplificar todo lo que se pueda.

¹⁸ Jakob Nielsen (nacido en 1957, en Copenhague, Dinamarca) es una de las autoridades más respetadas en el ámbito mundial sobre usabilidad en la web.

2.8 HERRAMIENTAS INFORMÁTICAS

2.8.1 Tecnología de la plataforma .NET

.NET Framework; es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con esta plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows. Se basa en el entorno de ejecución de lenguaje común CLR (Common Language Runtime). Este motor de ejecución proporciona un conjunto de servicios comunes para los proyectos generados en Visual Studio .Net, con independencia del lenguaje. Estos servicios proporcionan bloques de desarrollo fundamentales para aplicaciones de cualquier tipo, para todas las capas y niveles de la arquitectura de aplicaciones. Microsoft Visual Basic, Microsoft Visual C++, Microsoft Visual C# y otros lenguajes de programación se han mejorado para beneficiarse de estos servicios. [URL-12]

2.8.2 Visual Studio 2010

Visual Studio 2010 es la última versión estable de la plataforma de Microsoft, ofrece un entorno de desarrollo de alto nivel para desarrollar aplicaciones de escritorio, Web ASP móviles, que se ejecutan sobre el .Net Framework, proporciona las tecnologías fundamentales para simplificar la creación, implantación y evolución constante de aplicaciones Web y servicios Web XML seguros, escalables y de alta disponibilidad. [URL-13]

2.8.3 SQL Server 2008

SQL Server 2008 es la siguiente edición de su predecesor en el mercado, SQL Server 2005, el cual tuvo muy buena aceptación en las empresas por su alta calidad.

Con esta última versión se ha conseguido mejorar aún más SQL Server 2005, mejorando la fiabilidad, escalabilidad, rendimiento y manejo. Muchas de las empresas

controlan sus redes locales con el conocido sistema operativo Windows Server 2008, siendo uno de los más extendidos y preferidos por la mayoría de empresas. Este sistema operativo de Microsoft es el entorno ideal para la instalación de SQL Server 2008, convirtiéndose en la mejor pareja posible para la administración.

Además la reciente aparición de SQL Server 2008, conlleva que esté preparado para la expansión por la red de redes (Internet) ya que por ejemplo es capaz de generar automáticamente documentos XML, se trata del formato estándar de datos que facilita la transmisión de datos en Internet.

Una base de datos de SQL Server consta de una colección de tablas en las que se almacena un conjunto específico de datos estructurados. Una tabla contiene una colección de filas, también denominadas tuplas o registros, y columnas, también denominadas atributos. Cada columna de la tabla se ha diseñado para almacenar un determinado tipo de información; por ejemplo, fechas, nombres, importes en moneda o números.

2.8.3.1 Características de SQL Server 2008 [URL-14]

- Servidor de base de datos, de gran rendimiento.
- RDBMS que pueden ser instalados tanto en sistemas de usuarios como Windows XP/Vista/7, máquinas de multiprocesador de 64/32 bits, redes de ordenadores.
- La administración se facilita mediante interfaz gráfica de usuario.
- Capacidad de tener varias instancias del servidor en una única máquina.
- Acceso directo a datos desde página Web, gracias a la generación automática de documentos XML, consiguiendo una completa integración con Internet.
- Posibilidades de data warehousing y data mining, para almacenar y analizar datos, funcionando como Online Transaction Processing (OLTP) y con servicios Online Analytical Processing (OLAP).
- Comunicación perfecta con otras aplicaciones Microsoft, pudiendo presentar información en hojas de Excel, por citar un ejemplo.
- Integración perfecta con herramientas de desarrollo de software como Visual Studio 2008/2010.
- Lenguaje T-SQL para ampliar las posibilidades de las tareas a realizar.

- Capacidad para interpretar funciones realizadas con CLR (Common Language Runtime) de plataformas .NET, esto nos permite realizar funciones en lenguajes muy conocidos como Visual Basic o C#.

2.8.4 Arquitectura de tres capas

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño (ver Figura 2.16).

Llamado también Modelo Vista Controlador (MVC); Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista [Brown08].



Figura 2.15 Arquitectura de Tres Capas

Fuente: Elaboración Propia.

Capa de Presentación (Vista).- es la que ve el usuario, presenta el sistema de usuario, le comunica la información del usuario dando un mínimo de proceso. Esta capa se

comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable para el usuario.

Capa de negocio (Controlador).- esta capa es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

Capa de datos (Modelo).- esta capa es donde residen los datos y es la encargada de acceder a los datos. Está formada por uno o más gestores de base de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador. Si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación, las capas de negocios y datos pueden residir en el mismo ordenador o servidor, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

2.9 SEGURIDAD

2.9.1 Cifrado de datos transparente (TDE)

Hasta que salió SQL Server 2008 no había un método nativo para asegurar los ficheros físicos de la base de datos. Se podía usar el sistema operativo para asegurar directorios o ficheros o se podían asegurar celdas usando el cifrado de la versión 2005. También existía alguna herramienta para el cifrado de ficheros, pero eran adicionales y externos a SQL Server.

Podríamos definir el cifrado como el proceso de transformar información directamente inteligible, usando un algoritmo que lo hace ininteligible a cualquier otra persona que no tenga la clave. Hay varios métodos (algoritmos matemáticos) de cifrado,

algunos mejores que otros. Aunque no son perfectos, en seguridad hay un principio muy importante: saber cuánto cuesta romper la clave y cuánto se obtendría de su ruptura.

Para definir correctamente el cifrado de datos transparente (TDE por su nombre en inglés: «Transparent Data Encryption») recurriremos a sus creadores, Microsoft: «El cifrado de dato transparente (TDE) realiza el cifrado y descifrado de E/S en tiempo real de los datos y los archivos de registro. El cifrado utiliza una clave de cifrado de la base de datos (DEK), que está almacenada en el registro de arranque de la base de datos para que esté disponible durante la recuperación. La DEK es una clave simétrica protegida utilizando un certificado almacenado en la base de datos maestra del servidor o una clave asimétrica protegida por un módulo EKM. TDE protege los datos “en reposo”, es decir, los archivos de datos y de registro».

SQL Server 2008 admite algoritmos de cifrado que están, a su vez, basados en la API cifrado (CAPI) del sistema operativo. Estos algoritmos tienen nombres como DES, Triple DES, DESX, RC2, RC4, RC4 de 128 bits, AES de 128 bits, 192 bits AES y AES de 256 bits. Cada sistema operativo soporta una lista específica de algoritmos de cifrado.

La principal ventaja respecto a otras tecnologías, es que no necesita cambios en la aplicación que se tenga desarrollada. Este permite a los administradores de bases de datos codificar la base de datos usando cualquier algoritmo de cifrado admitido sin tener que hacer ningún cambio a la aplicación cliente ni a los objetos de base de datos. **[TSJ11]**

CAPÍTULO III
MARCO APLICATIVO

3 MARCO APLICATIVO

3.1 ANÁLISIS Y DISEÑO DEL SISTEMA DE INFORMACIÓN

Uno de los principales problemas que enfrenta un proyecto de desarrollo de software, es la corta posibilidad de saber con exactitud los verdaderos y completos requerimientos del sistema al inicio del desarrollo, por más exhaustivo que sea el levantamiento de los requerimientos y más detallado el análisis y diseño inicial, en cualquier etapa del proyecto surgirán necesidades complementarias al inicio del proyecto que son imprescindibles para la conclusión del mismo. Las fases que se desarrollan a continuación contemplan prácticas de modelado en análisis y diseño del sistema utilizando la metodología UWE (modelo de casos de uso, conceptual, de presentación y navegacional), la codificación se realizará en base a una planeación de 3 iteraciones (entregas de producto software en base a los requerimientos esenciales del usuario) para la implementación de un producto software completo que controle el flujo de información del programa “Por una sonrisa sana y feliz del escolar” del Gobierno Autónomo Municipal de La Paz. Por las características del proyecto, se expondrá las fases y modelos más relevantes, manteniendo los modelos esenciales y descriptivos que facilitarán el análisis y diseño del sistema.

3.2 OBTENCIÓN DE REQUISITOS

La tarea de ingeniería de requisitos es fundamental para que un sistema sea exitoso, en este sentido para la realización del presente proyecto se realizaron las actividades que indica la Tabla 3.1.

Tabla 3.1 Tareas para la obtención de requisitos

Tarea	Característica
Entrevista	Se realizaron entrevistas con: <ul style="list-style-type: none"> ▪ Jefe de la UNACE. ▪ Encargado del programa “Por una sonrisa sana y feliz del escolar” ▪ Encargado del programa “Nutrición y salud escolar”
Observación	Todos los procesos y trabajos operativos que realiza el programa “Por una sonrisa sana y feliz del escolar”, son manuales. El sistema informático del programa “Nutrición y salud escolar”, está en funcionamiento como una aplicación de escritorio, pero no es utilizada en integridad, ya que no cumple con las expectativas del usuario y solo es administrado por el encargado del programa, derivando en repetidos procesos manuales por parte de los nutricionistas.
Documentación	Se tuvo acceso a toda la documentación física.
Cliente	Encargado de establecer y negociar requerimientos, decide cuando un requisito está satisfecho. También establece la prioridad de las funcionalidades a implementar, interfaz gráfica del usuario (GUI) y criterios de usabilidad.

Fuente: Elaboración Propia

3.2.1 Definición de Actores

La Tabla 3.2 muestra, la lista de actores, junto con una descripción de sus actividades relacionadas con la utilización del sistema.

Tabla 3.2 Lista de Actores

No.	Actor	Descripción
1	Administrador	Encargado de toda la configuración inicial y mantenimiento de los datos con los que funcionará el sistema. Gestiona a los usuarios (odontólogos profesionales y egresados), materiales e insumos que se le asignan.
2	Odontólogo Profesional	Encargado de administrar unidades educativas y pacientes, generar reportes específicos para los informes mensuales.
3	Odontólogo Egresado	Encargado de gestionar a pacientes, con su respectivo diagnóstico y tratamiento odontológico realizado.
4	Visitante	Es la persona que visita el portal de UNACE, y puede consultar información relevante acerca de los programas en cuestión.

Fuente: Elaboración Propia

3.2.2 Lista de Requerimientos del Sistema

La Tabla 3.3 muestra las características que necesita el sistema, a partir de la información obtenida como parte de las tareas de obtención de requisitos detallado en la Tabla 3.1.

Tabla 3.3 Funciones Básicas del Sistema

Ref.	Función	Categoría
R1	Control de acceso seguro y diferenciado de usuarios	Evidente
R2	Gestión de usuarios	Evidente
R3	Desplegar vistas y menú de acuerdo a rango de usuario	Oculto
R4	Gestión de pacientes	Evidente
R5	Gestión de odontograma	Evidente
R6	Gestión de historial clínico	Evidente
R7	Cálculo de índices dentales	Oculto
R8	Generación de reportes	Evidente
R9	Gestión unidades educativas	Evidente
R10	Gestión de productos	Evidente
R11	Cálculo de stock de productos	Oculto
R12	Búsqueda de usuarios, pacientes, unidad educativa	Evidente

Fuente: Elaboración Propia

3.2.3 Definición de procesos

Después de la obtención de requerimientos del sistema se explica los procesos más relevantes del sistema que cada actor espera gestionar a través de la aplicación Web.

➤ Administrador

- Gestión de usuarios: el administrador basado en una lista de odontólogos profesionales y egresados que trabajan en el programa, realizará un registro de usuarios, asignándole: nombre de usuario, contraseña, tipo de profesional y rango para tener un control de acceso seguro y diferenciado.

Dar de alta a un usuario

Dar de baja a un usuario

Actualizar perfil de usuario

Listar usuarios activos

Buscar usuario

Actualizar perfil personal

- Gestión de productos: con la lista actualizada de materiales e insumos que le asignan mensualmente al programa para realizar todo el trabajo odontológico en las respectivas unidades educativas, el administrador registrará todos los productos en el sistema, para tener un control de stock según sea requerido.

A partir de la carga de productos (materiales e insumos) dentro del sistema, podrá asignar en el sistema los materiales e insumos que son entregados a cada odontólogo, según stock, registrando fecha y cantidad del respectivo proceso.

Dar de alta a un producto

Dar de baja a un producto

Actualizar información y stock de producto

Listar productos

Buscar producto

Productos asignados a usuario

Estado de entrega de producto a usuario

➤ Odontólogo profesional

- Gestión de pacientes: después del trabajo de recolección de información general, diagnóstico (línea base) y tratamiento odontológico que se realiza a cada escolar (paciente) manualmente, este usuario podrá registrar al escolar atendido en caso de no estar registrado en el sistema.

Buscar un paciente

Dar de alta a un paciente

Dar de baja a un paciente

Actualizar información de paciente

- Generación de reportes: mensualmente se entrega un informe de avance y de estadísticas según el diagnóstico específico de cada pieza dental y los tratamientos realizados por el programa. Este usuario podrá generar reportes por grupos etáreos (edad y sexo), tipo de dentición y graficas estadísticas de las curaciones realizadas en cualquier unidad educativa o macrodistrito que trabajaron.

- Odontólogo egresado
 - Gestión de odontograma: cada escolar registrado tiene un odontograma y una ficha medica computarizada, este usuario registrará el diagnóstico de cada pieza dental en un odontograma interactivo y de fácil manejo, si se realizó un tratamiento específico en alguna pieza dental se registrara dicha curación en otro odontograma interactivo y de fácil manejo.

3.2.4 Herramientas de Desarrollo

Para el desarrollo del proyecto se utilizará las siguientes herramientas:

Herramientas de modelado del sistema:

- Plug-in open source ArgoUWE (www.pst.ifi.lmu.de/projekte/uwe) para la herramienta también open source ArgoUML (www.argouml.org), para las fases de navegación y presentación.
- Gliffy (www.gliffy.com), herramienta profesional online para el modelo de contenido, diagramas UML, diagrama entidad relación.
- LovelyCharts (www.lovelycharts.com/), herramienta profesional online para el modelado del mapa de navegación.
- Balsamiq Mockups (www.balsamiq.com), herramienta profesional online para el diseño de interfaz.

Herramientas para el desarrollo del sistema:

- Adobe Dreamweaver CS4
- Visual Studio 2010
- Sql Server 2008 R2
- Cristal Reports

Herramientas RIA:

- Ajax
- JavaScript
- JQuery

3.3 MODELO DE CASOS DE USO

3.3.1 Caso de Uso Principal

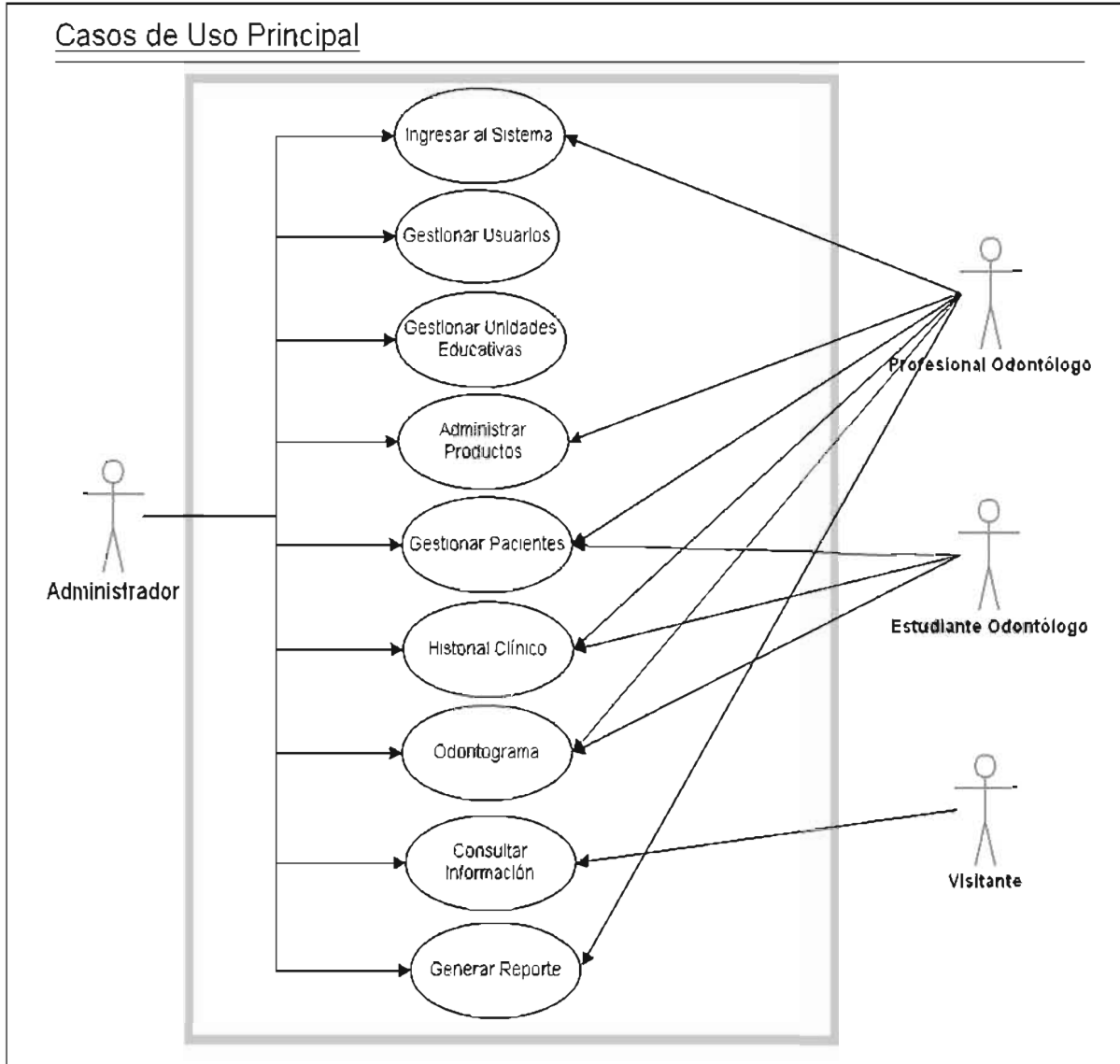


Figura 3.1 Caso de Uso Principal

Fuente: Elaboración Propia

3.3.2 Descripción de Casos de Uso

3.3.2.1 Caso de Uso Ingresar a Sistema

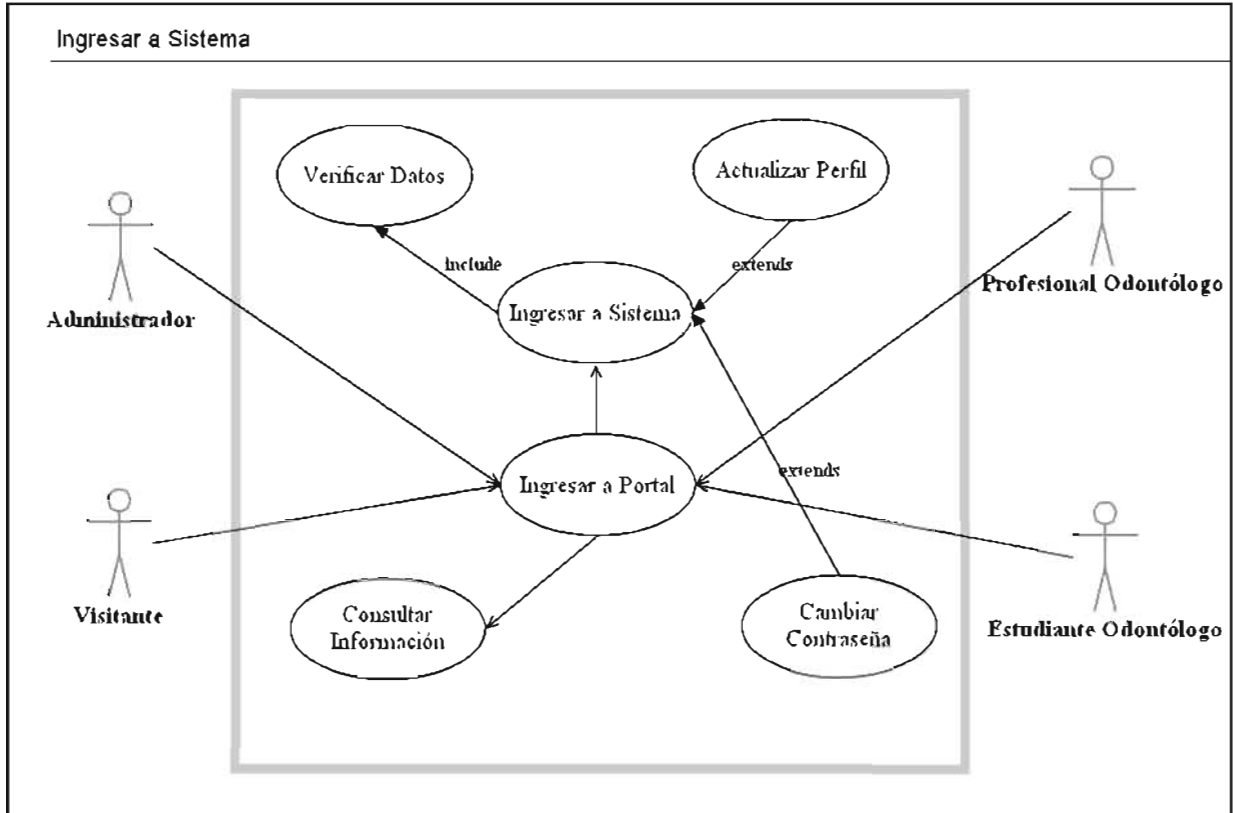


Figura 3.2 Caso de Uso Ingresar a Sistema

Fuente: Elaboración Propia

Tabla 3.4 Especificación de Caso de Uso ingresar a sistema

[CU-01]	Ingresar a Sistema
Objetivo en contexto	Describe el proceso de ingresar al sistema a través del portal web.
Precondiciones	Ninguna
Actores	GUI ¹⁹ del Portal, Servidor, Base de Datos
Secuencia normal	<p>Escenario Principal (Login)</p> <ol style="list-style-type: none"> 1. Ingresa dirección url del portal web. 2. Selecciona la opción de login de sistema. Si no es usuario registrado a S1. 3. Inserta nombre de usuario y contraseña. 4. El sistema accede a la estructura de datos comprueba nombre de usuario y contraseña, información correcta accede a sistema. Si no puede acceder, error S2. <p>Escenario Alterno 1 (Cambiar contraseña). Continúa en el punto 4) del escenario principal.</p> <p>Escenario Alterno 2 (Actualizar perfil) Continúa en el punto 4) del escenario principal.</p>
Secuencia alternativa	<p>S1. Usuario no registrado puede navegar por el portal y consultar información relevante y específica de índices odontológicos e índices nutricionales por alumno y unidad educativa.</p> <p>S2. Mensaje de error porque no se puede acceder a la Base de Datos o el nombre de usuario y/o contraseña son incorrectos.</p>

Fuente: Elaboración Propia

¹⁹ GUI (del inglés graphical user interface) Interfaz Gráfica de Usuario.

3.3.2.2 Caso de Uso Gestionar Usuarios

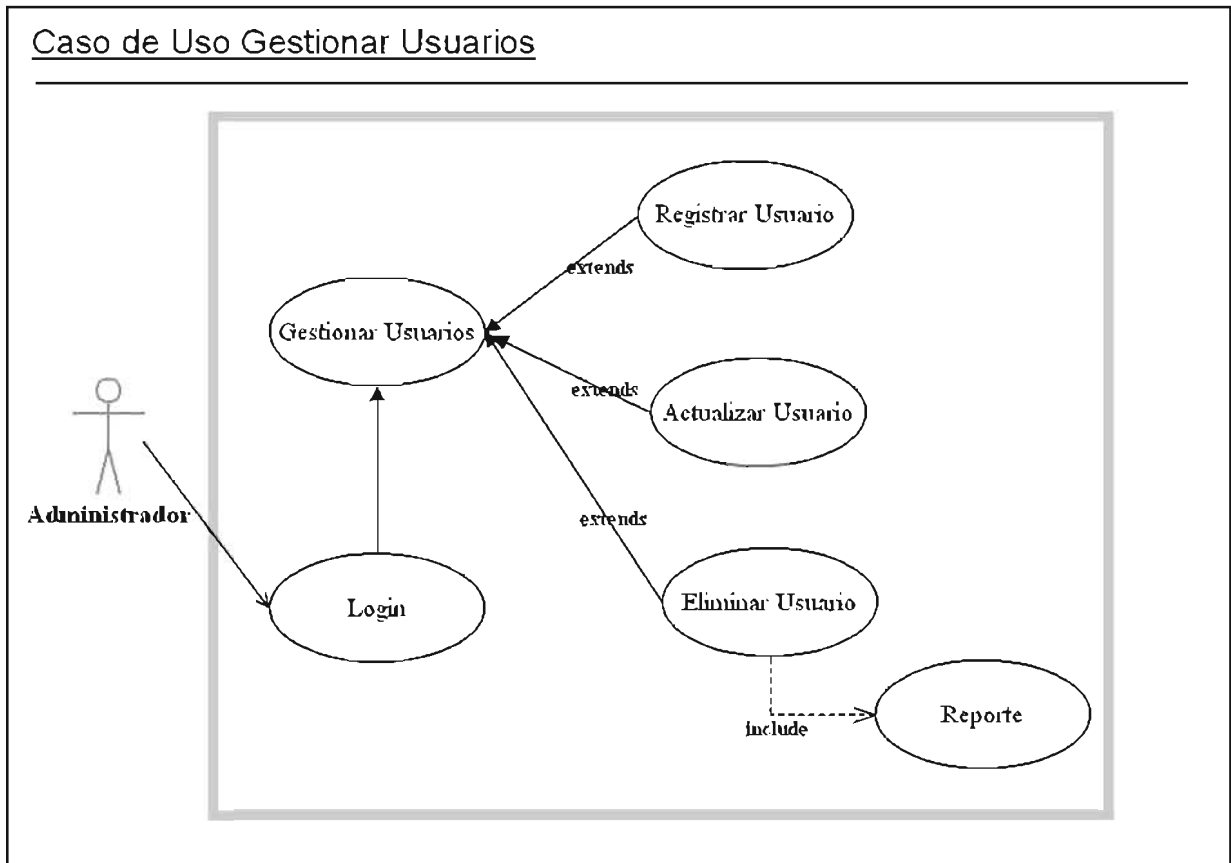


Figura 3.3 Caso de Uso Gestionar Usuarios

Fuente: Elaboración Propia

Tabla 3.5 Especificación de Caso de Uso gestionar usuarios

[CU-02]	Gestionar usuarios en la aplicación
Objetivo en contexto	Permite gestionar (ingresar, eliminar o actualizar) a través de la aplicación web, los usuarios que tendrán acceso al sistema.
Precondiciones	El administrador es el que realiza la operación y por tanto tiene que loguearse primero.
Actores	GUI del Administrador, Servidor, Base de Datos
Secuencia normal	<p>Escenario Principal (Actualizar usuario)</p> <ol style="list-style-type: none"> 1. Selecciona la opción usuarios. 2. El sistema muestra la lista de usuarios. 3. Selecciona opción ver usuario. 4. Mostrar la pantalla perfil de usuario y pulsa la opción actualizar datos. Si no se puede mostrar ir a S2 5. Mostrar pantalla de inserción de datos del usuario mostrando datos actuales. 6. Una vez modificados, el sistema comprueba que estén correctamente especificados los datos, guarda y envía datos. Si no, error S1 7. El sistema accede a la estructura de datos y actualiza el usuario. Si no puede acceder, error S2. <p>Escenario Alternativo 1 (Ingresar nuevo usuario)</p> <p>Escenario Alternativo 2 (Eliminar usuario existente)</p>
Secuencia alternativa	<p>S1. Campos incompletos o tipo de dato incorrecto. Se solicita al administrador que introduzca los datos de nuevo.</p> <p>S2. Mensaje de error porque no se puede acceder a la base de datos.</p>

Fuente: Elaboración Propia

3.3.2.3 Caso de Uso Gestionar Pacientes

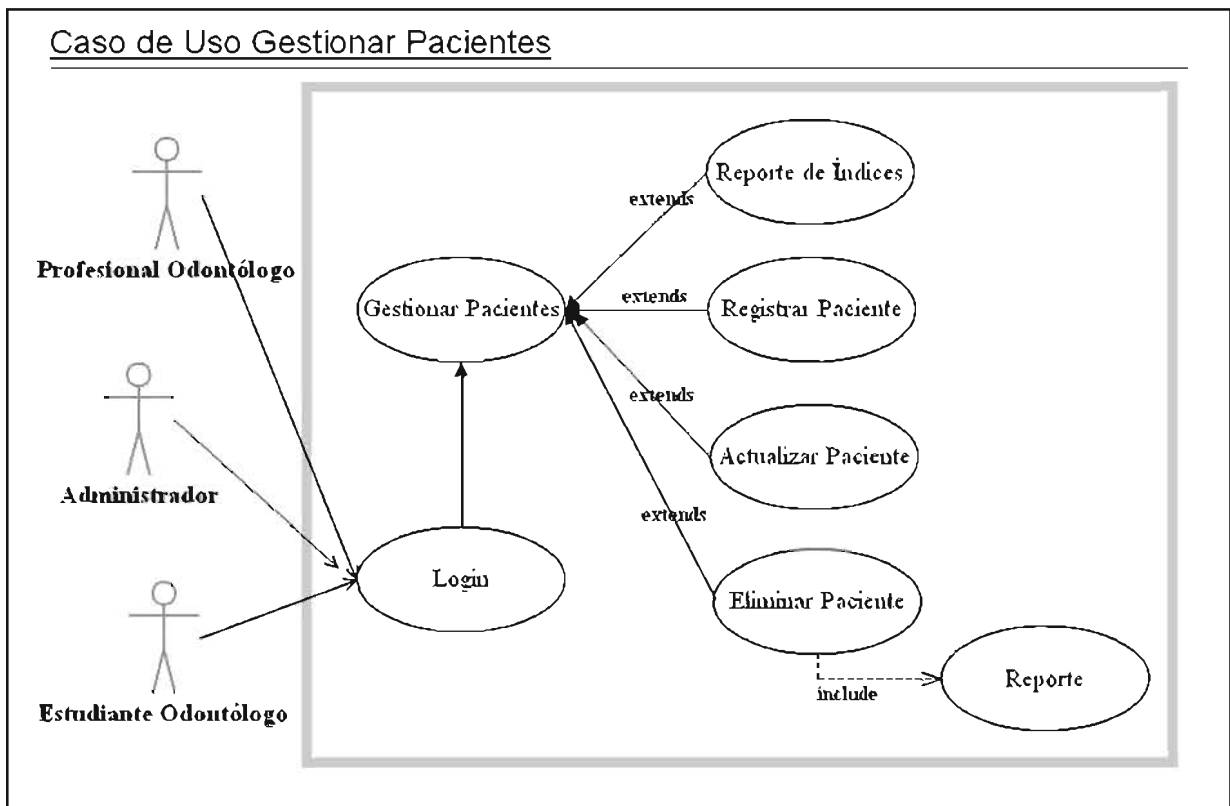


Figura 3.4 Caso de Uso Gestionar Pacientes

Fuente: Elaboración Propia

Tabla 3.6 Especificación de Caso de Uso gestionar pacientes

[CU-03]	Gestionar pacientes en la aplicación
Objetivo en contexto	Permite gestionar (ingresar, eliminar o actualizar) a través de la aplicación web, a toda la población escolar.
Precondiciones	El Administrador, Odontólogo Profesional y Estudiante son los que realizan la operación y por tanto tienen que loguearse primero.
Actores	GUI del Administrador, GUI del Profesional Odontólogo, GUI del Estudiante Odontólogo, Servidor, Base de Datos
Secuencia normal	<p>Escenario Principal (Actualizar paciente)</p> <ol style="list-style-type: none"> 1. Seleccionar la opción paciente. 2. El sistema muestra la pantalla de búsquedas de pacientes. 3. Busca al paciente por unidad educativa, macrodistrito o todos. 4. Seleccionar la opción ver paciente. 5. Mostrar la pantalla perfil de paciente y pulsar la opción actualizar datos. Si no se puede mostrar ir a S2 6. Mostrar pantalla de actualización de datos del paciente mostrando datos actuales. 7. Una vez modificados, el sistema comprueba que estén correctamente especificados los datos, guarda y envía los datos. Si no, error S1 8. El sistema accede a la estructura de datos y actualiza la información del paciente. Si no puede acceder, error S2. <p>Escenario Alternativo 1 (Ingresar nuevo paciente)</p> <p>Escenario Alternativo 2 (Eliminar paciente existente)</p>
Secuencia alternativa	<p>S1. Campos incompletos o falta de datos. Se solicita al administrador que introduzca los datos de nuevo.</p> <p>S2. Mensaje de error porque no se puede acceder a la base de datos.</p>

Fuente: Elaboración Propia

3.3.2.4 Caso de Uso Odontograma

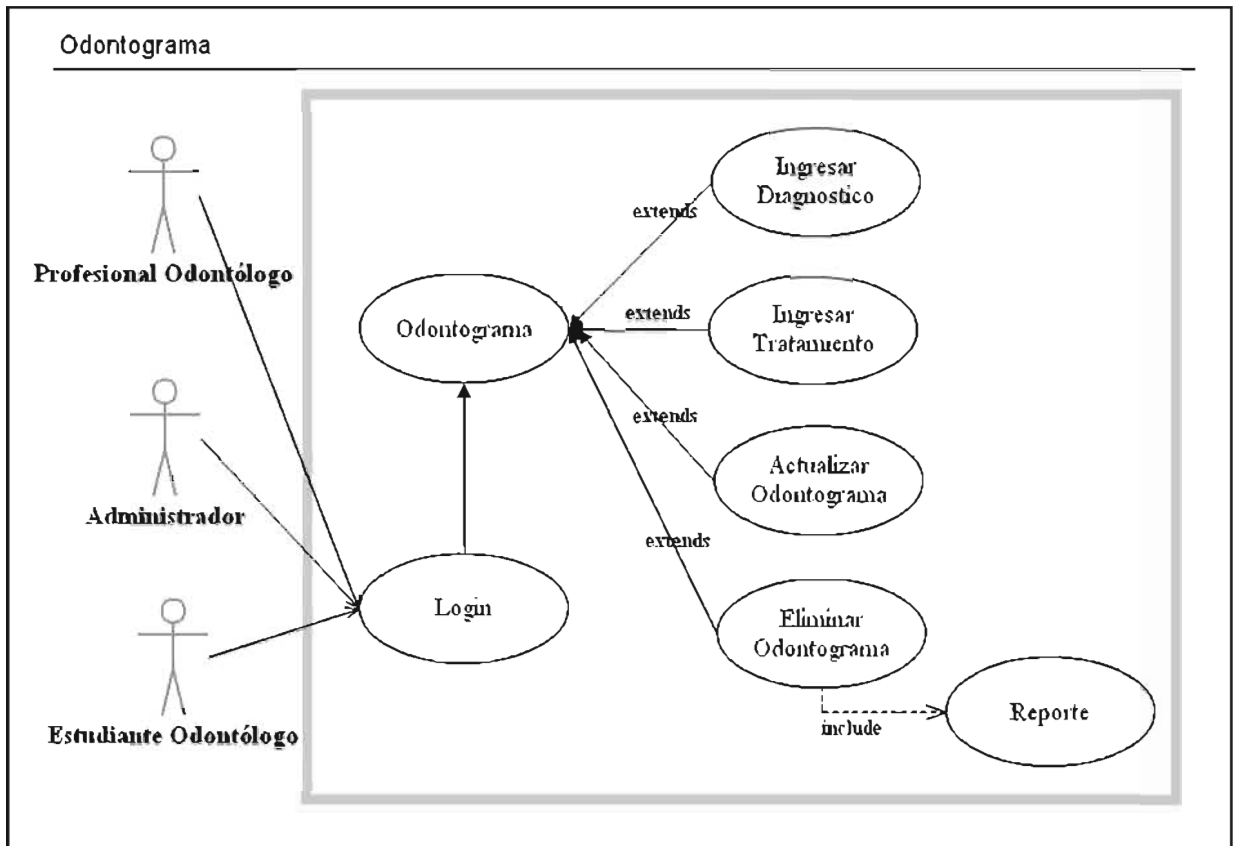


Figura 3.5 Caso de Uso Odontograma

Fuente: Elaboración Propia

Tabla 3.7 Especificación de Caso de Uso odontograma

[CU-04]	Odontograma
Objetivo en contexto	Describe el proceso de incluir datos referentes al levantamiento de información (diagnóstico y tratamiento dental) a través de la aplicación web, del odontograma dental de cada paciente atendido.
Precondiciones	El Administrador, Profesional y Estudiante Odontólogo son los que realizan la operación y por tanto tienen que loguearse primero.
Actores	GUI del Administrador, GUI del Profesional Odontólogo, GUI del Estudiante Odontólogo, Servidor, Base de Datos
Secuencia normal	<p>Escenario Principal (Ingresar Odontograma)</p> <ol style="list-style-type: none"> 1. Seleccionar la opción paciente. 2. El sistema muestra la pantalla de búsquedas de pacientes. 3. Buscar al paciente por unidad educativa, macrodistrito o todos. 4. Selecciona la opción ver paciente 5. Muestra la pantalla perfil de paciente y pulsa la opción odontograma. Si no se puede mostrar ir a S2 6. Muestra pantalla de inserción de datos para el diagnóstico preliminar y su respectivo tratamiento que se aplicó al paciente. 7. El sistema comprueba que estén especificados los datos necesarios, guarda y envía datos. Si no a S1. 8. El sistema accede a la estructura de datos registra y calcula índices de higiene dental de la entrada del nuevo odontograma. Si no puede acceder a S2. <p>Escenario Alternativo 1 (Actualizar odontograma)</p> <p>Escenario Alternativo 2 (Eliminar odontograma)</p>

Secuencia alternativa	<p>S1. Campos incompletos o falta de datos. Se solicita al administrador que introduzca los datos de nuevo y correctamente.</p> <p>S2. Mensaje de error porque no se puede acceder a la base de datos.</p>
------------------------------	--

Fuente: Elaboración Propia

3.3.2.5 Caso de Uso Historial Clínico

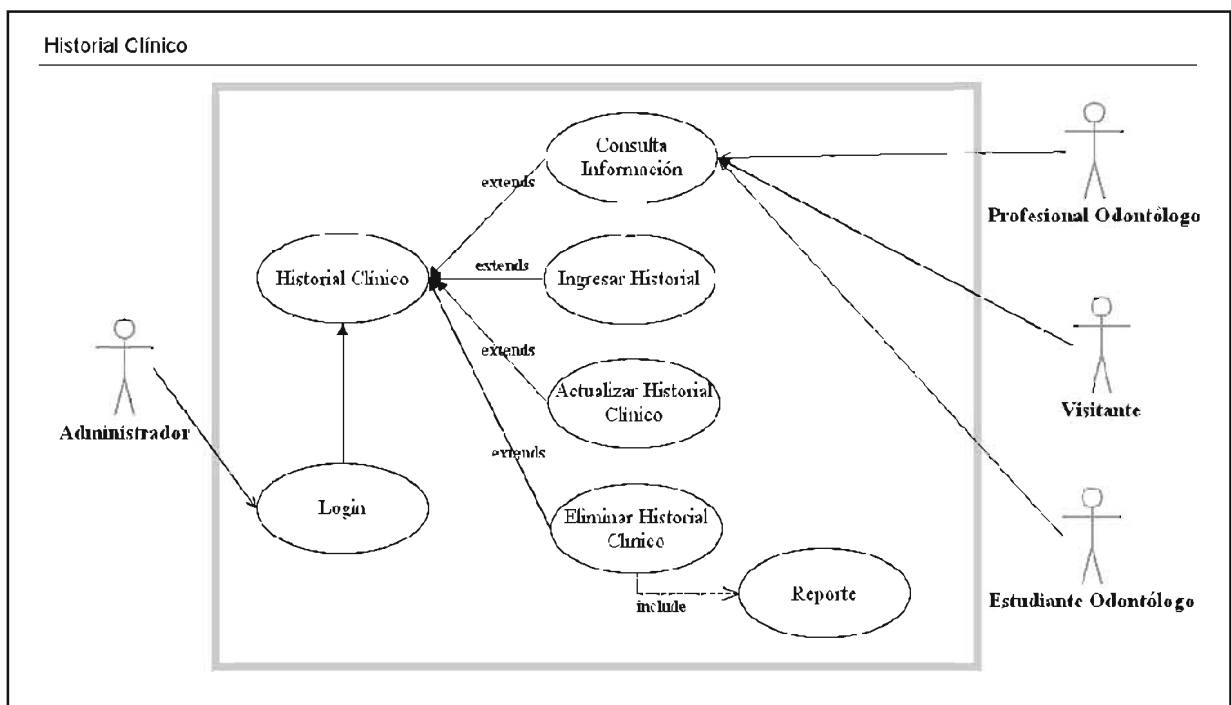


Figura 3.6 Caso de Uso Historial Clínico

Fuente: Elaboración Propia

Tabla 3.8 Especificación de Caso de Uso historial clínico

[CU-05]	Historial Clínico
Objetivo en contexto	Describe el proceso de incluir datos referentes al levantamiento de información a través de la aplicación web, del historial médico general y nutricional de cada paciente atendido.
Precondiciones	El Administrador es que realiza la operación y por tanto tienen que loguearse primero.
Actores	GUI del Administrador, Servidor, Base de Datos
Secuencia normal	<p>Escenario Principal (Ingresar Historial Clínico)</p> <ol style="list-style-type: none"> 1. Seleccionar la opción paciente. 2. El sistema muestra la pantalla de búsquedas de pacientes. 3. Busca al paciente por unidad educativa, macrodistrito o todos. 4. Selecciona la opción ver paciente 5. Mostrar la pantalla perfil de paciente y pulsa la opción historial médico. Si no se puede mostrar ir a S2 6. Mostrar pantalla de inserción de datos para el historial médico general con todos sus antecedentes que se obtuvo del paciente. 7. El sistema comprueba que estén especificados los datos necesarios, guarda y envía datos. Si no a S1. 8. El sistema accede a la estructura de datos registra y calcula índices nutricionales de la entrada del nuevo historial clínico. Si no puede acceder a S2. <p>Escenario Alternativo 1 (Actualizar Historial Clínico)</p> <p>Escenario Alternativo 2 (Eliminar Historial Clínico)</p>

Secuencia alternativa	<p>S1. Campos incompletos o falta de datos. Se solicita al administrador que introduzca los datos de nuevo y correctamente.</p> <p>S2. Mensaje de error porque no se puede acceder a la base de datos.</p>
------------------------------	--

Fuente: Elaboración Propia

3.4 MODELO DE CONTENIDO

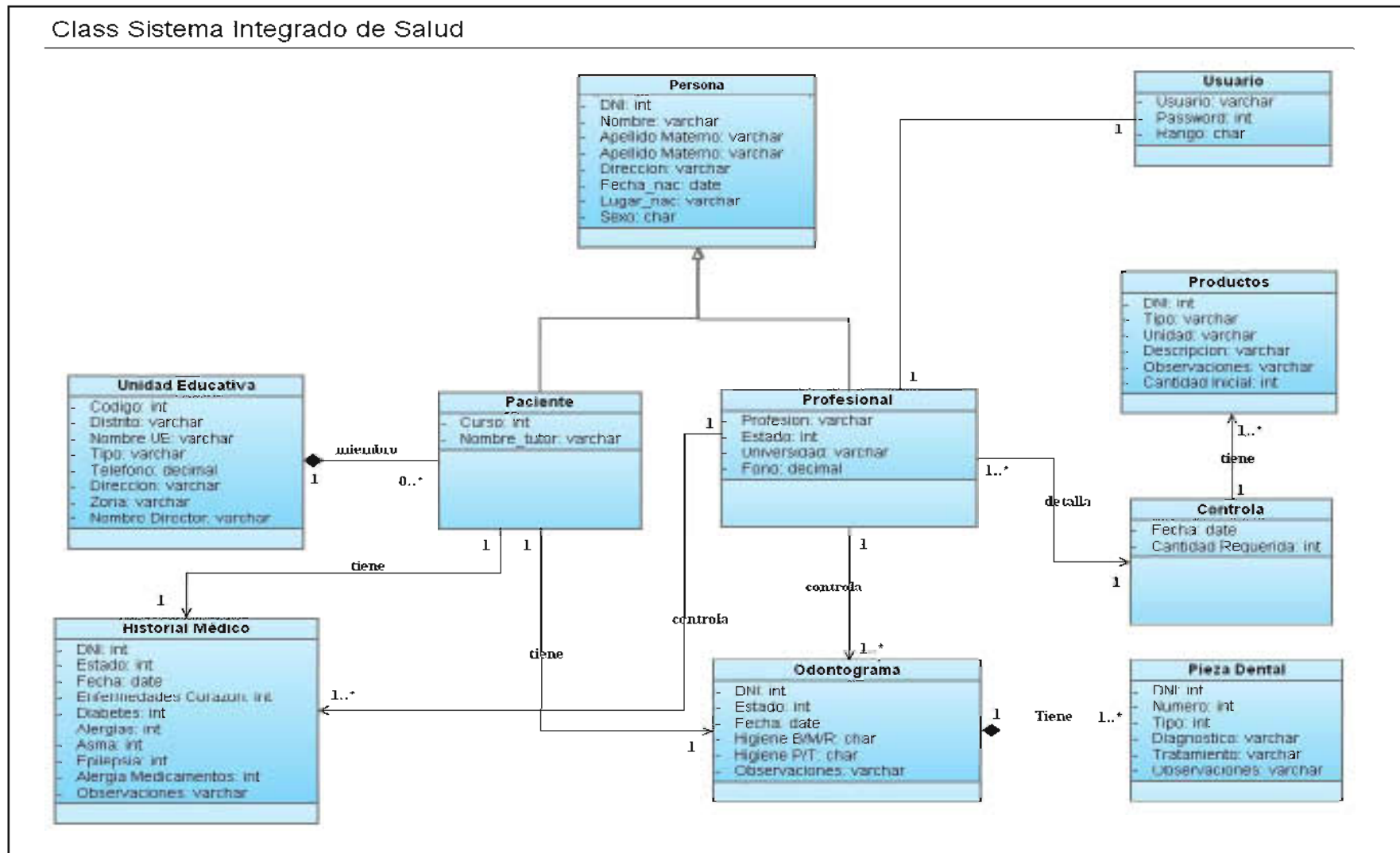


Figura 3.7 Clases de Análisis

Fuente: Elaboración Propia

3.5 MODELO DE NAVEGACIÓN

Los siguientes diagramas presentan el flujo de navegación de diferentes usuarios según el rango asignado en la aplicación Web. Los límites y alcances de navegación de cada usuario.

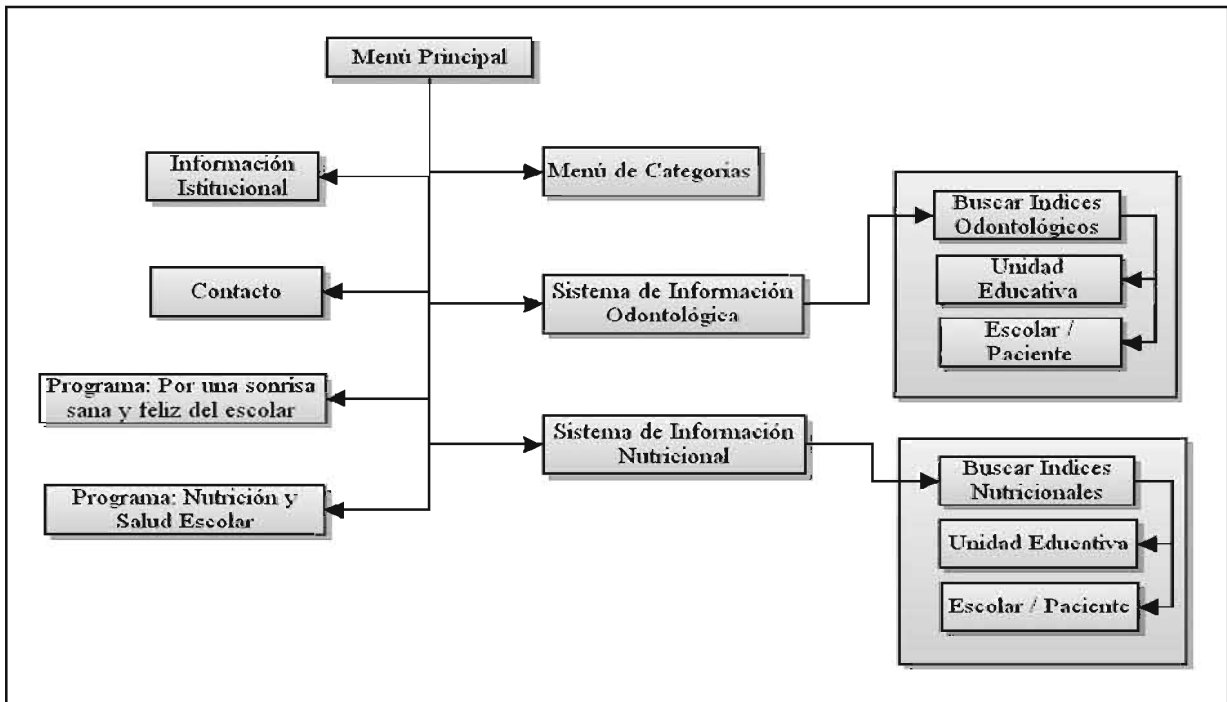


Figura 3.8 Diagrama de Navegación para Visitante

Fuente: Elaboración: Propia

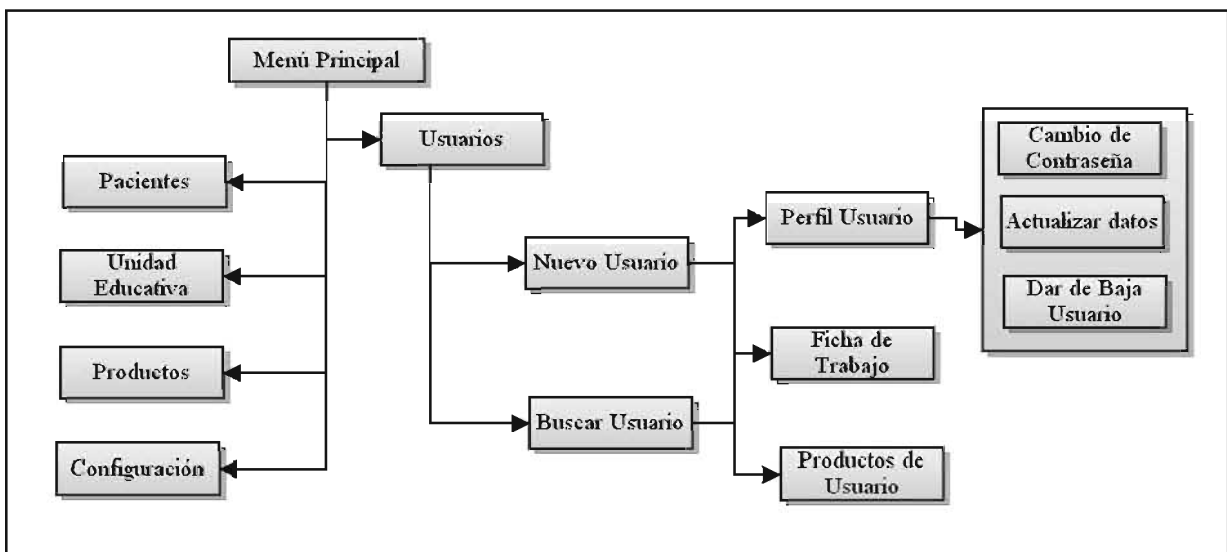


Figura 3.9 Diagrama de Navegación para Administrador

Fuente: Elaboración: Propia

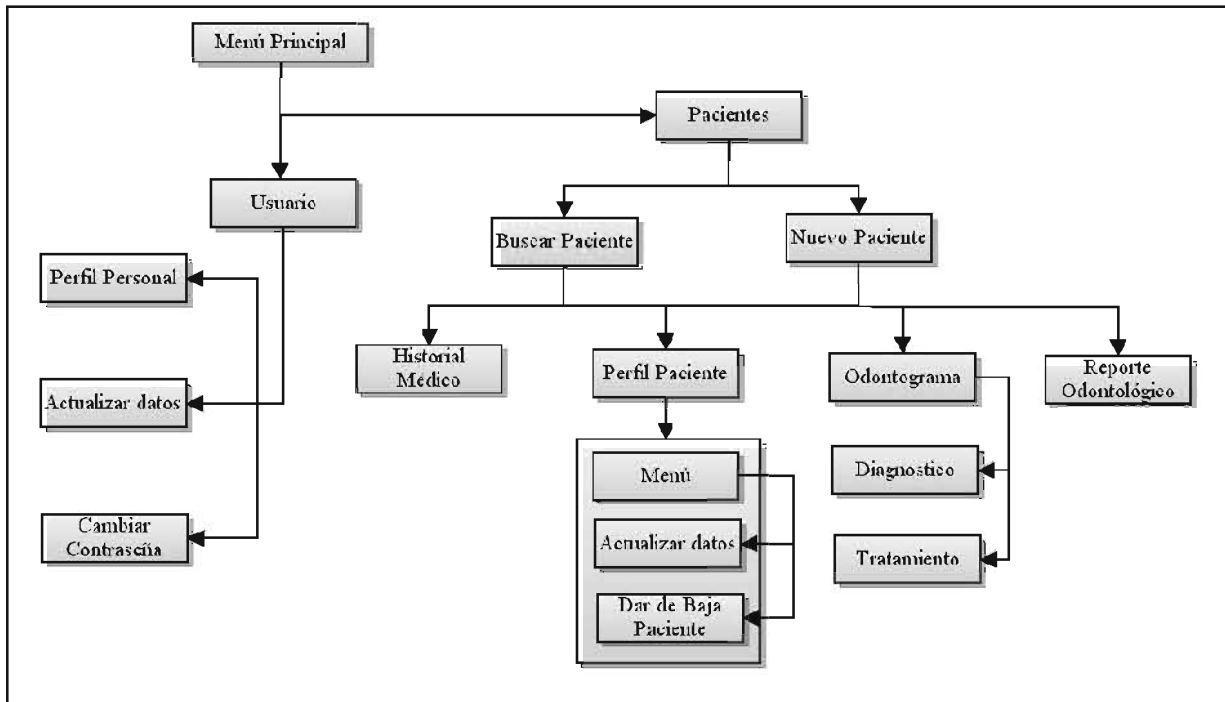


Figura 3.10 Diagrama de Navegación para Estudiante Odontólogo

Fuente: Elaboración: Propia

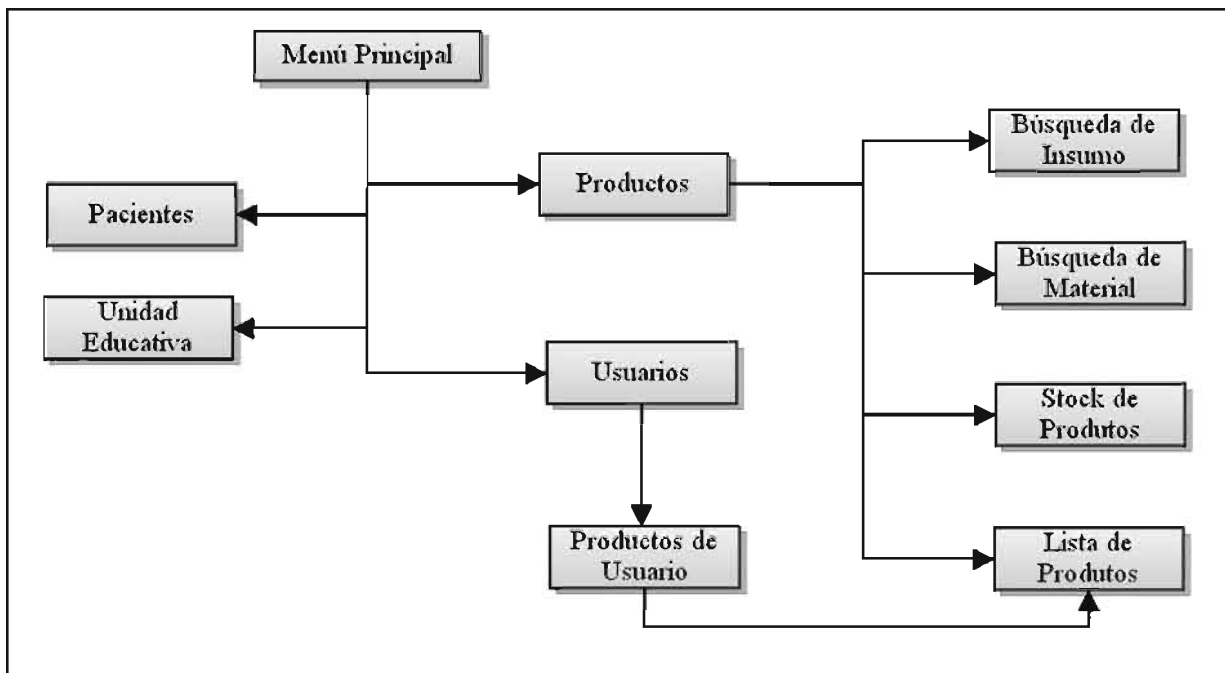


Figura 3.11 Diagrama de Navegación del Profesional Odontólogo

Fuente: Elaboración: Propia

3.5.1 Mapa de Navegación GUI Administrador

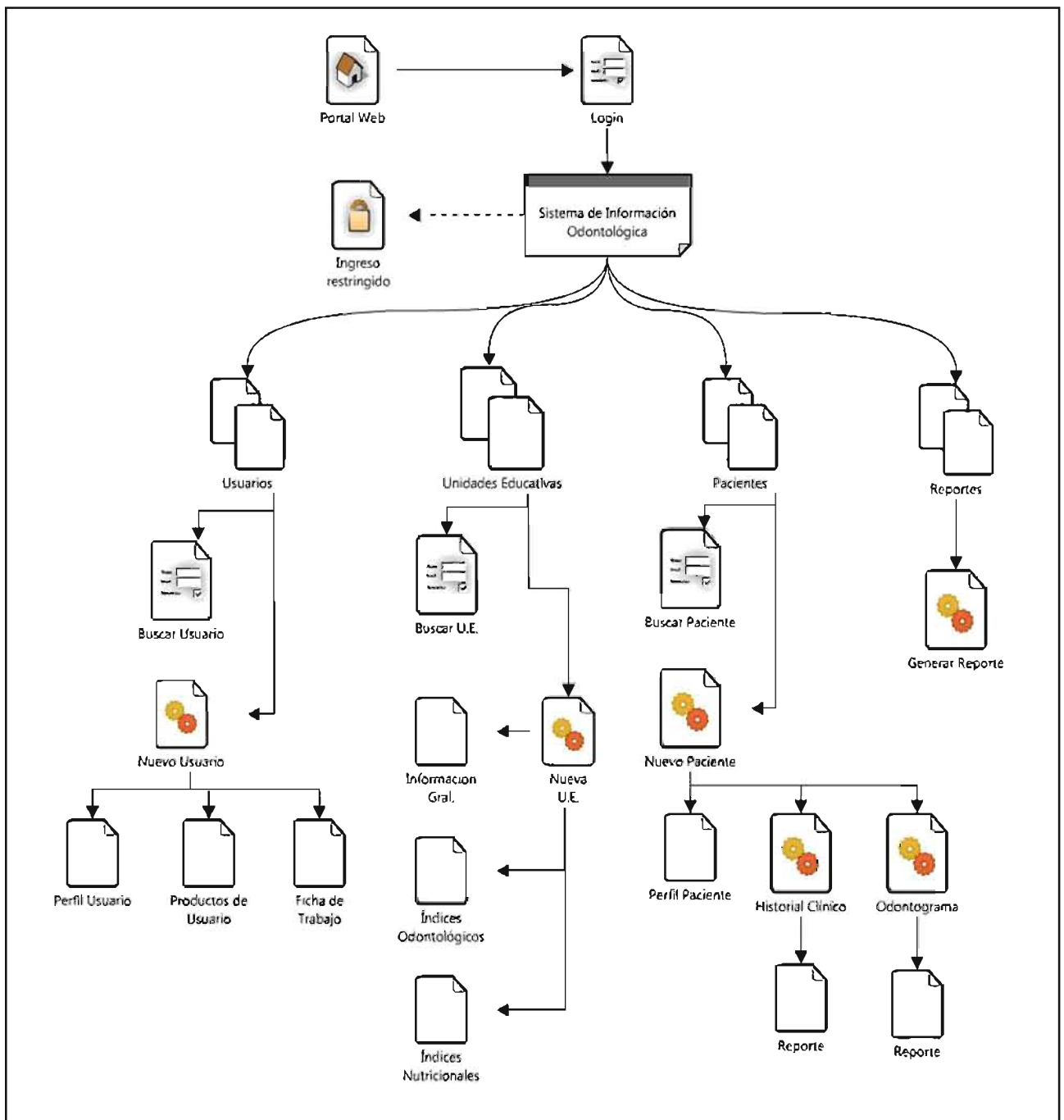


Figura 3.12 Mapa de Navegación GUI Administrador

Fuente: Elaboración: Propia

3.6 MODELO DE PRESENTACIÓN

Los diagramas de este modelo nos permiten observar como la aplicación Web presentará la vista de estructura y organización de los datos.


Sistema Integrado de Salud
Módulo Odontológico

Logotipo

▼ MENÚ PRINCIPAL

- Inicio
- Indices
- Manual de Usuario
- Módulo Odontológico
- Módulo Nutrición y Salud
- Contacto

Acceder

 LOGIN

Nombre de Usuario

Contraseña

No cerrar Sesión

Usuario Nuevo? [Contactese con el Administrador](#)

© Copyright 2011 Gobierno Autónomo Municipal de La Paz [Inicio](#) | [Contacto](#)

Figura 3.13 Diseño de Interfaz – Autenticación

Fuente: Elaboración: Propia

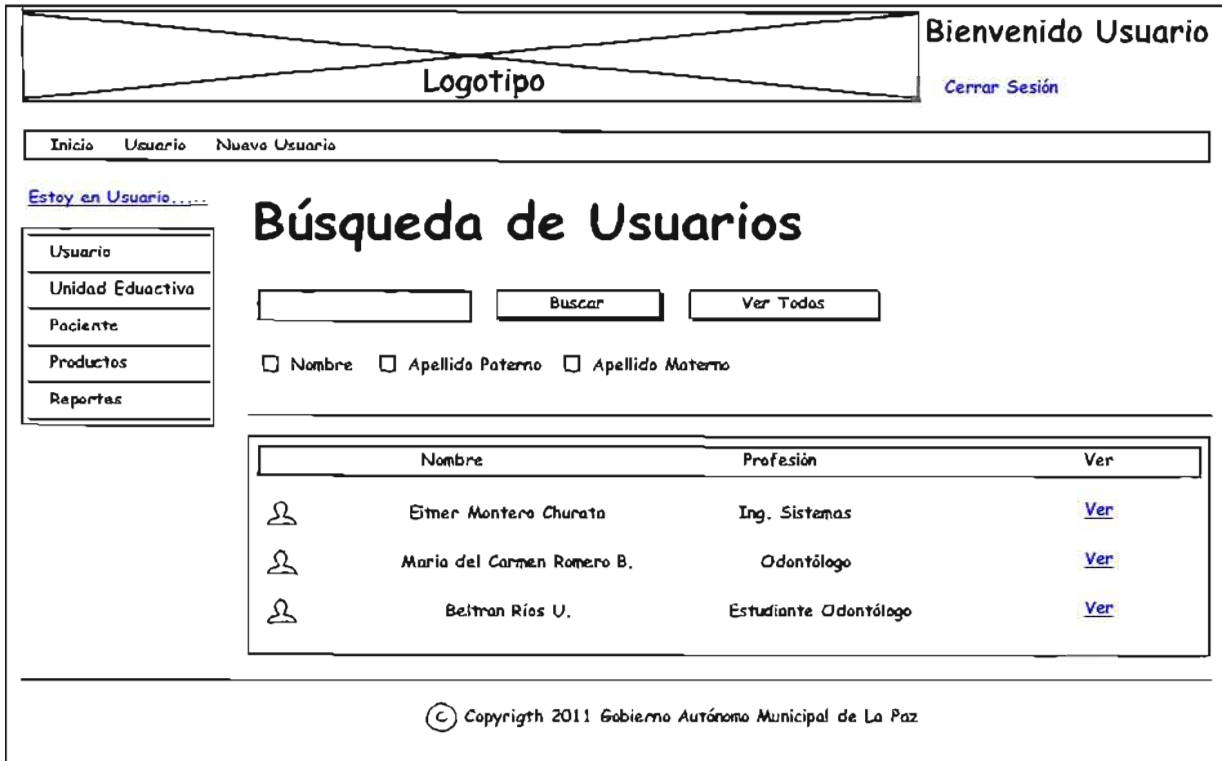


Figura 3.14 Diseño de Interfaz – Búsqueda de Usuarios

Fuente: Elaboración: Propia

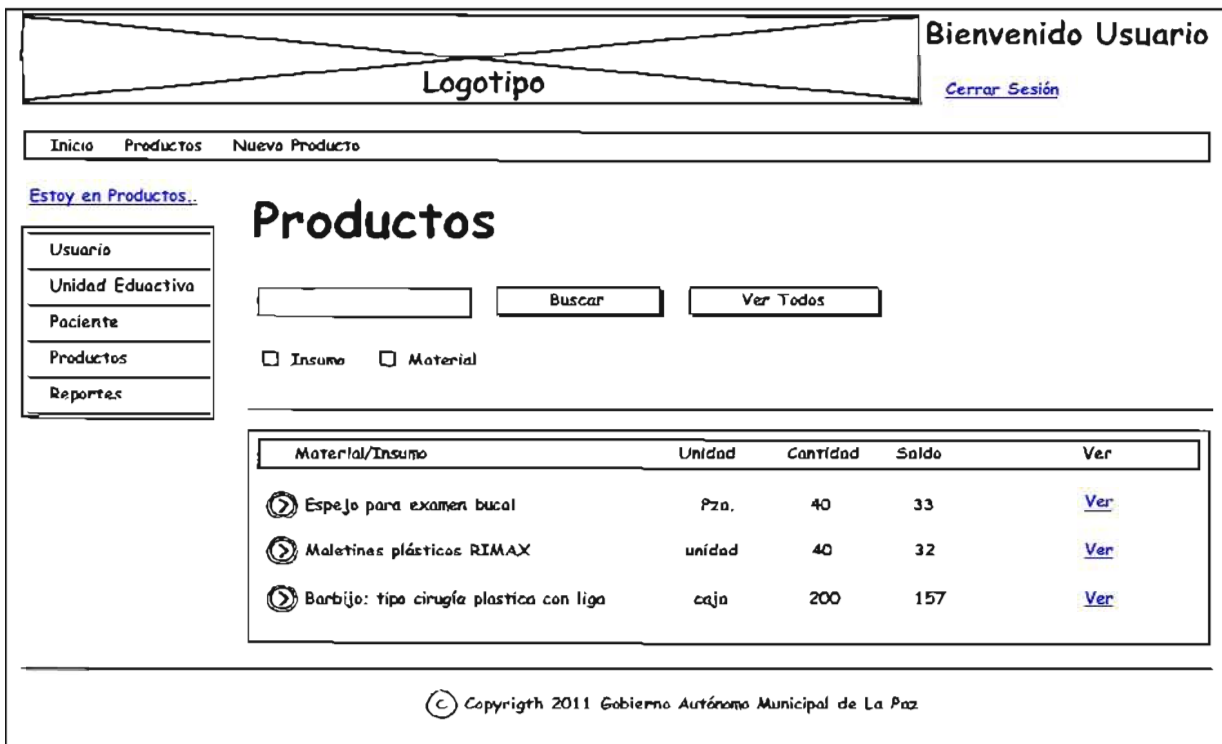



Figura 3.15 Diseño de Interfaz – Productos

Fuente: Elaboración: Propia



Bienvenido Usuario

[Cerrar Sesión](#)

Inicio
Usuario
Perfil Usuario
Producto de Usuario
Ficha de Trabajo

[Estoy en Perfil de Usuario....](#)

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Usuario</td></tr> <tr><td style="text-align: center;">Unidad Educativa</td></tr> <tr><td style="text-align: center;">Paciente</td></tr> <tr><td style="text-align: center;">Productos</td></tr> <tr><td style="text-align: center;">Reportes</td></tr> </table>	Usuario	Unidad Educativa	Paciente	Productos	Reportes	<table style="width: 100%;"> <tr> <td style="width: 30%;">Nombre</td> <td style="width: 30%;"><input type="text" value="Ermer"/></td> <td style="width: 10%; text-align: right;">*</td> </tr> <tr> <td>Apellido Paterno</td> <td><input type="text" value="Montero"/></td> <td></td> </tr> <tr> <td>Apellido Materno</td> <td><input type="text" value="Churata"/></td> <td></td> </tr> <tr> <td>C.I.</td> <td><input type="text" value="5808569"/></td> <td style="text-align: right;">*</td> </tr> <tr> <td>Dirección</td> <td><input type="text" value="Z- Miraflores"/></td> <td></td> </tr> <tr> <td>Fecha de Nacimiento</td> <td><input type="text" value="16/06/1985"/></td> <td style="text-align: right;">DD/MM/AAAA</td> </tr> <tr> <td>Lugar de Nacimiento</td> <td><input type="text" value="Tarija"/></td> <td></td> </tr> <tr> <td>Teléfono</td> <td><input type="text" value="70029687"/></td> <td></td> </tr> <tr> <td>Profesión</td> <td><input type="text" value="Informático"/></td> <td></td> </tr> <tr> <td>Universidad</td> <td><input type="text" value="UMSA"/></td> <td></td> </tr> </table>	Nombre	<input type="text" value="Ermer"/>	*	Apellido Paterno	<input type="text" value="Montero"/>		Apellido Materno	<input type="text" value="Churata"/>		C.I.	<input type="text" value="5808569"/>	*	Dirección	<input type="text" value="Z- Miraflores"/>		Fecha de Nacimiento	<input type="text" value="16/06/1985"/>	DD/MM/AAAA	Lugar de Nacimiento	<input type="text" value="Tarija"/>		Teléfono	<input type="text" value="70029687"/>		Profesión	<input type="text" value="Informático"/>		Universidad	<input type="text" value="UMSA"/>	
Usuario																																				
Unidad Educativa																																				
Paciente																																				
Productos																																				
Reportes																																				
Nombre	<input type="text" value="Ermer"/>	*																																		
Apellido Paterno	<input type="text" value="Montero"/>																																			
Apellido Materno	<input type="text" value="Churata"/>																																			
C.I.	<input type="text" value="5808569"/>	*																																		
Dirección	<input type="text" value="Z- Miraflores"/>																																			
Fecha de Nacimiento	<input type="text" value="16/06/1985"/>	DD/MM/AAAA																																		
Lugar de Nacimiento	<input type="text" value="Tarija"/>																																			
Teléfono	<input type="text" value="70029687"/>																																			
Profesión	<input type="text" value="Informático"/>																																			
Universidad	<input type="text" value="UMSA"/>																																			

<input type="button" value="Actualizar Información"/>	<input type="button" value="Dar de Baja"/>
---	--

Usuario	<input type="text" value="MCE-160685"/>	
Password	<input type="text" value="1234567"/>	<input type="button" value="Cambiar Contraseña"/>
Rango	<input checked="" type="radio"/> Administrador <input type="radio"/> Profesional Odontólogo <input type="radio"/> Estudiante Odontólogo	

© Copyright 2011 Gobierno Autónomo Municipal de La Paz

Figura 3.16 Diseño de Interfaz – Perfil de Usuario

Fuente: Elaboración: Propia

Bienvenido Usuario

[Cerrar Sesión](#)

Inicio
Paciente
Perfil Paciente
Odontograma
Historial Médico
Reporte Odontológico

Estoy en Odontograma...

Usuario

Unidad Educativa

Paciente

Productos

Reportes

Diagnostico Odontológico - Línea Base

16 / 06 / 1985

Guardar Diagnostico

Índice del Paciente

CPD 5

Tratamiento Odontológico

Índice IHOS

17/16	11	26/27	?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
47/46	31	36/37	

Higiene

B	R	M	P	T
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Guardar

Actualizar

© Copyright 2011 Gobierno Autónomo Municipal de La Paz

C - Cariado (Diente Permanente)
 P - Pérdido (Diente Permanente)
 El - Extraído Indicado (Diente Permanente)
 O - Obturada (Diente Permanente)
 c - cariado (Diente Temporal)
 ei - extraído indicado (Diente Temporal)
 a - obturada (Diente Temporal)

Figura 3.17 Diseño de Interfaz – Odontograma

Fuente: Elaboración: Propia

3.7 DIAGRAMA ENTIDAD RELACIÓN

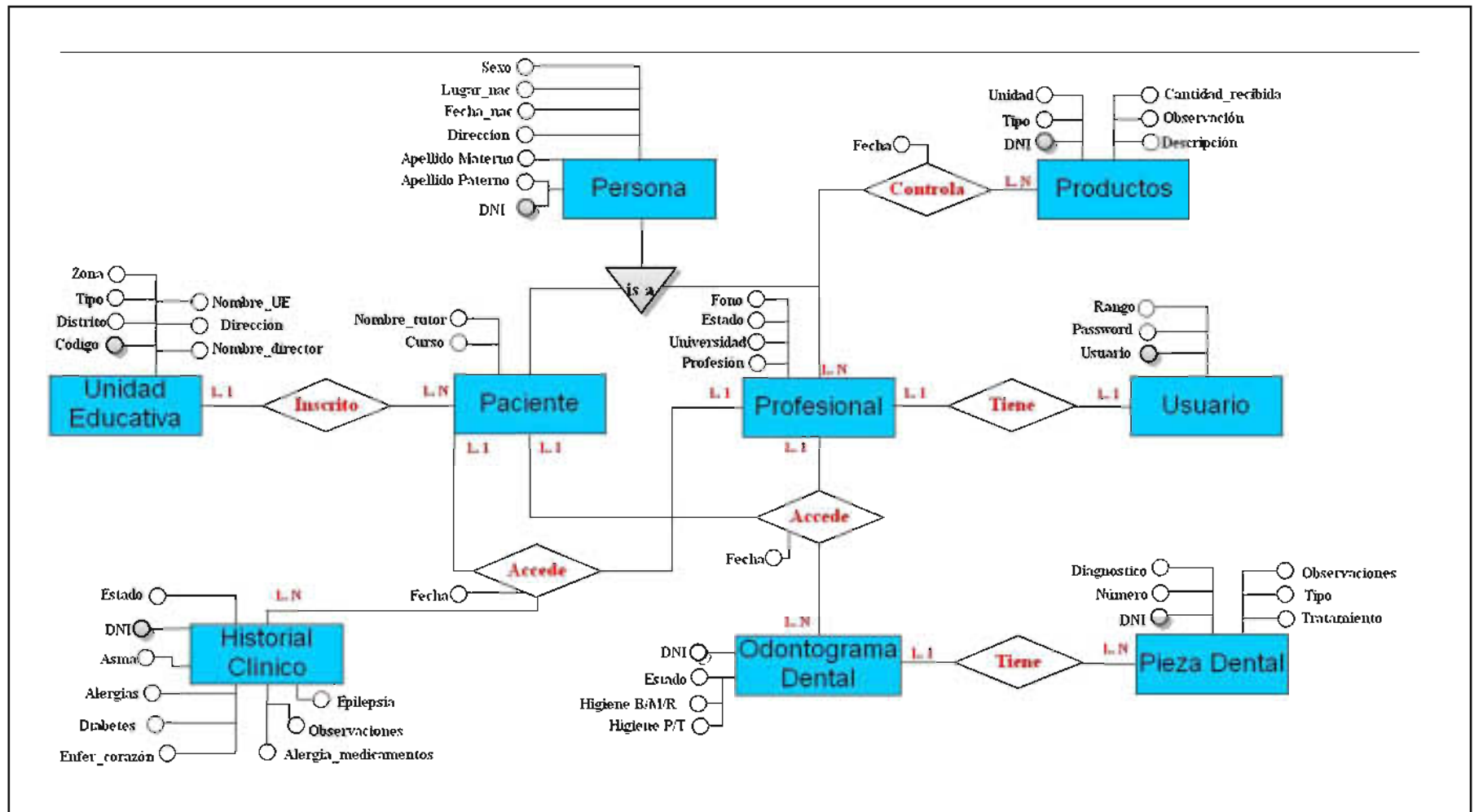


Figura 3.18 Diagrama Entidad Relación

Fuente: Elaboración: Propia

3.8 DIAGRAMA DE CLASES



Figura 3.19 Diagrama de Clases

Fuente: Elaboración: Propia

3.9 MODELO FÍSICO

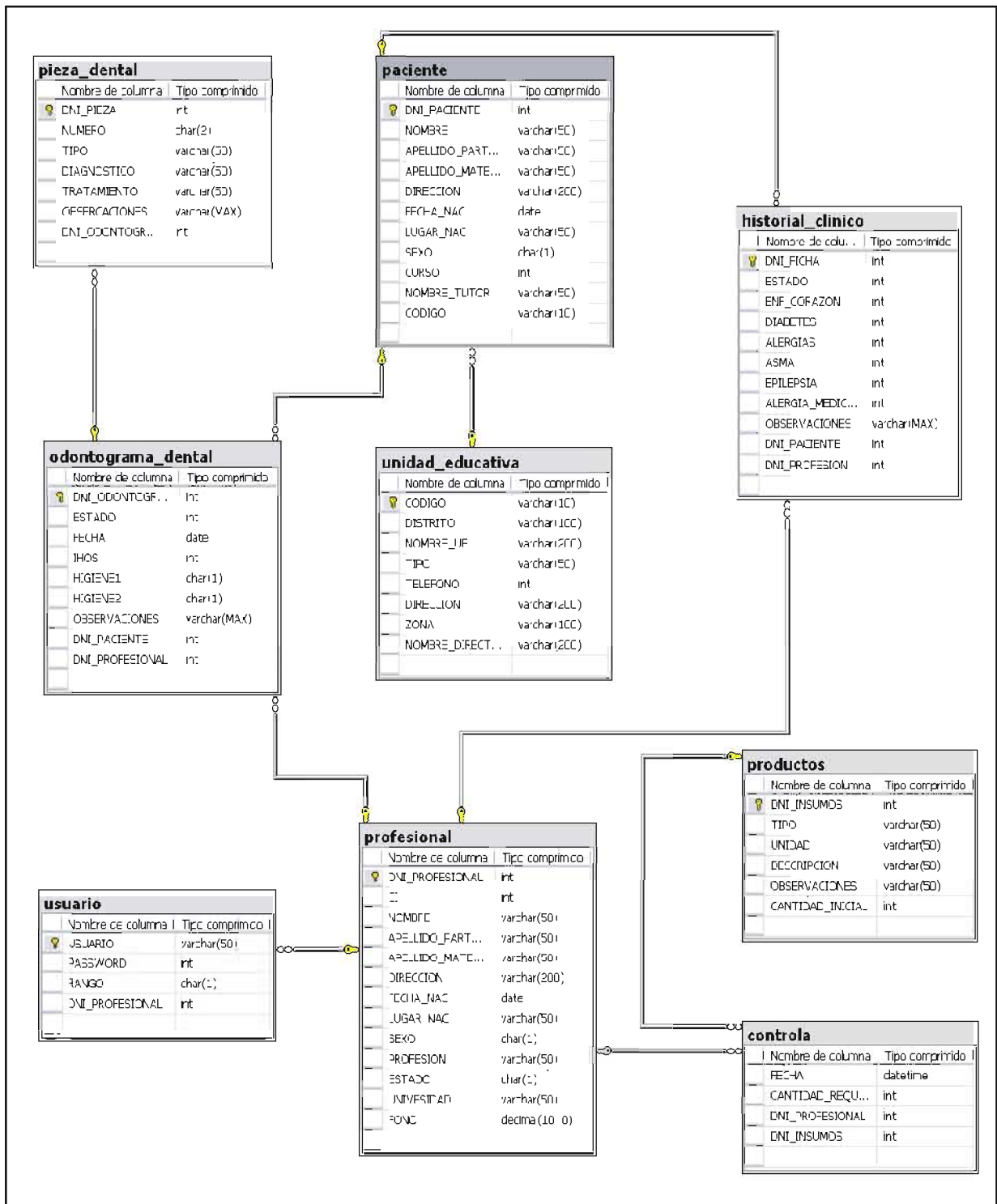


Figura 3.20 Modelo Físico de Datos

Fuente: Elaboración: Propia

3.10 IMPLEMENTACIÓN

3.10.1 Captura de ventanas del sistema

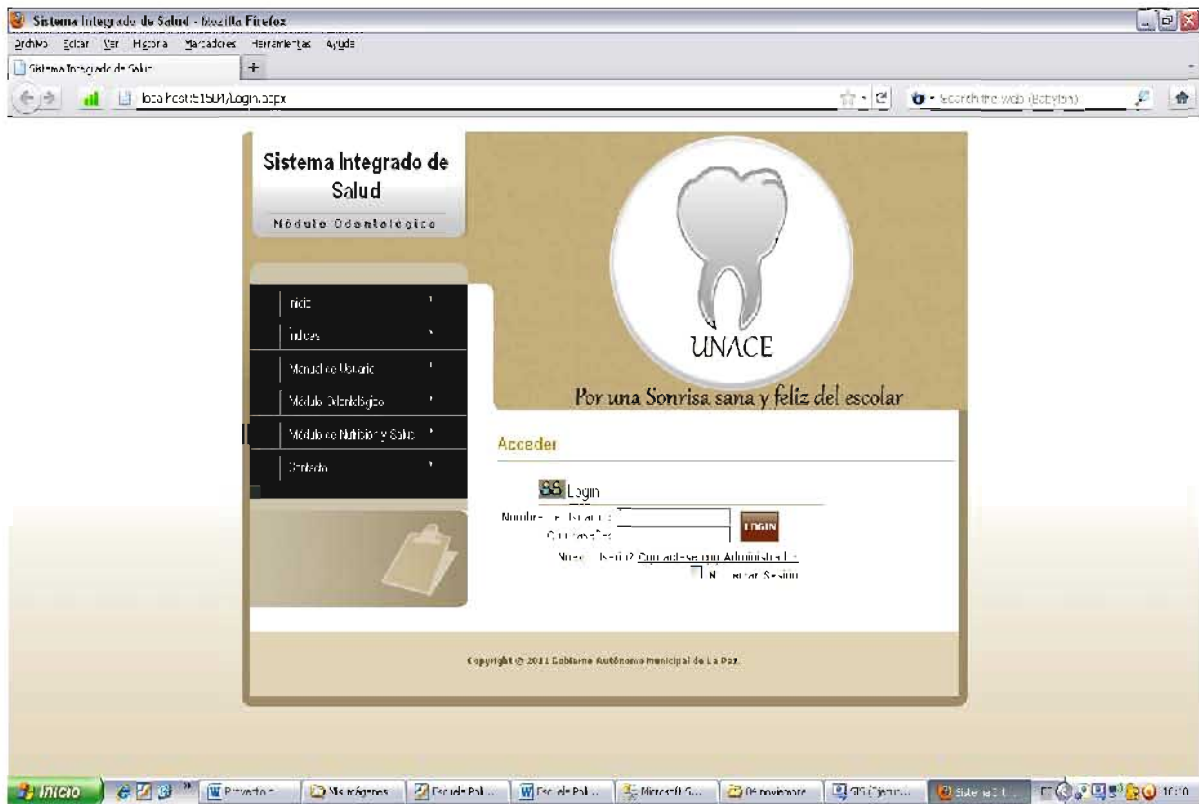


Figura 3.21 Ingreso al Subsistema de Información Odontológica

Fuente: Elaboración: Propia

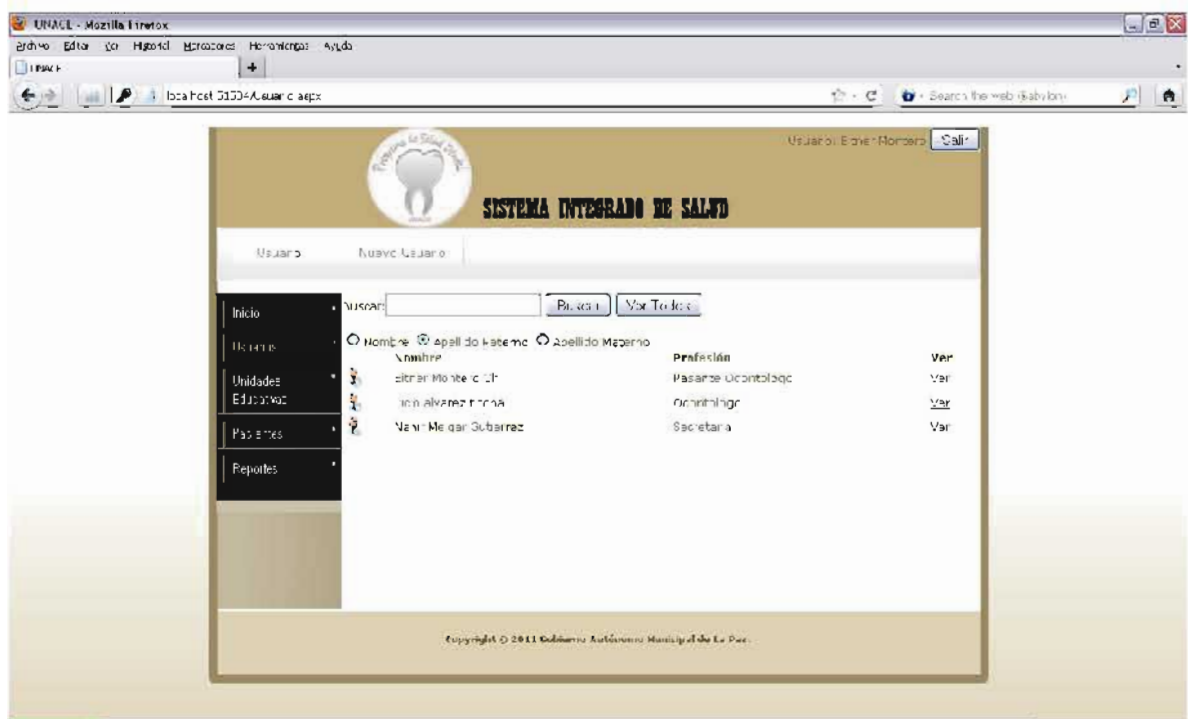


Figura 3.22 Gestión de Usuarios

Fuente: Elaboración: Propia

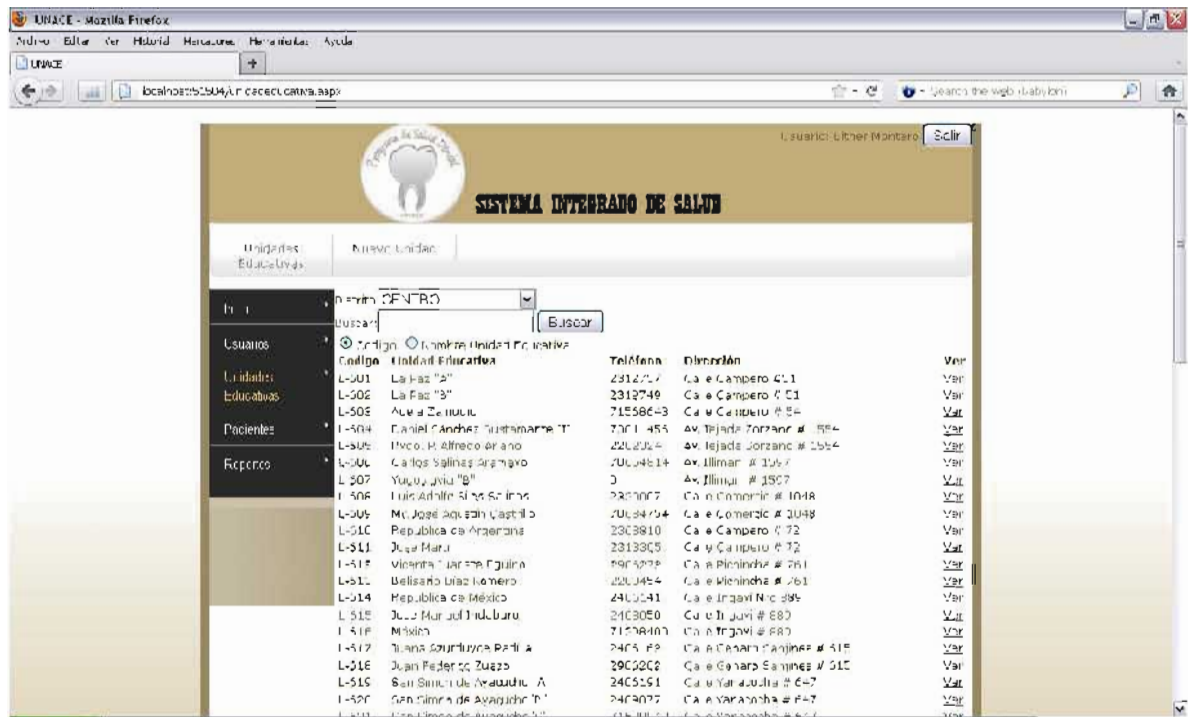


Figura 3.23 Búsqueda de Unidades Educativas

Fuente: Elaboración: Propia



Figura 3.24 Perfil de Unidad Educativa

Fuente: Elaboración: Propia

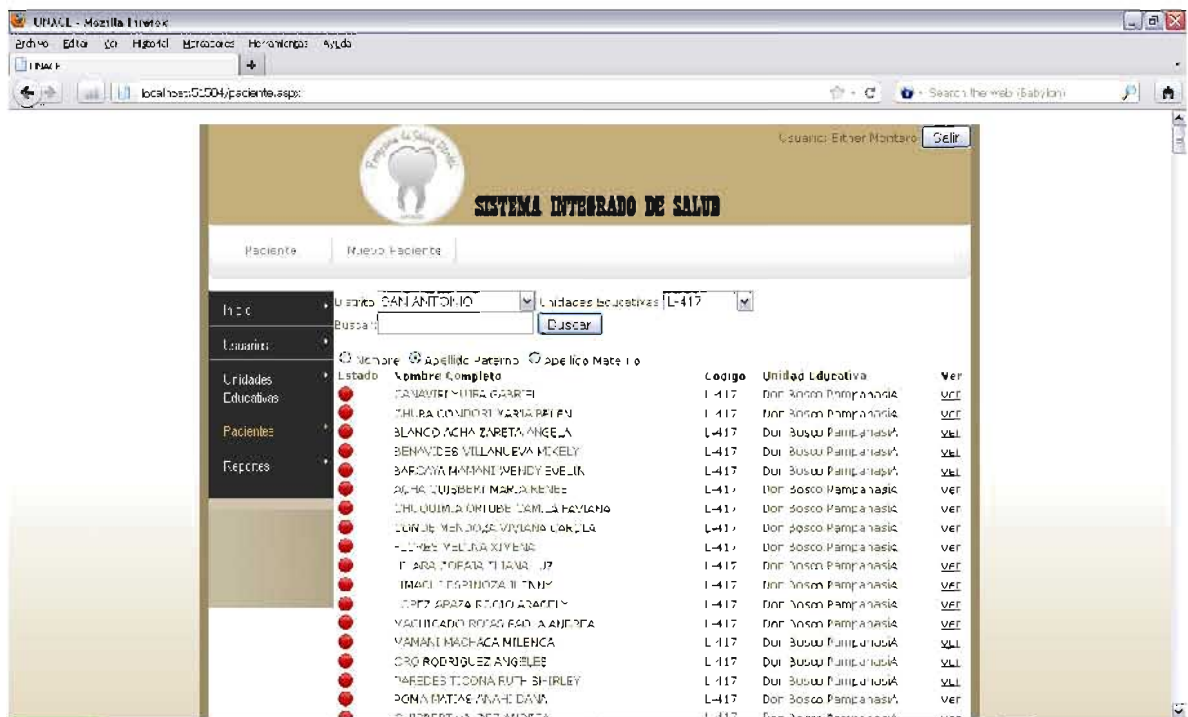


Figura 3.25 Búsqueda de Paciente

Fuente: Elaboración: Propia



Figura 3.26 Gestión de Paciente

Fuente: Elaboración: Propia

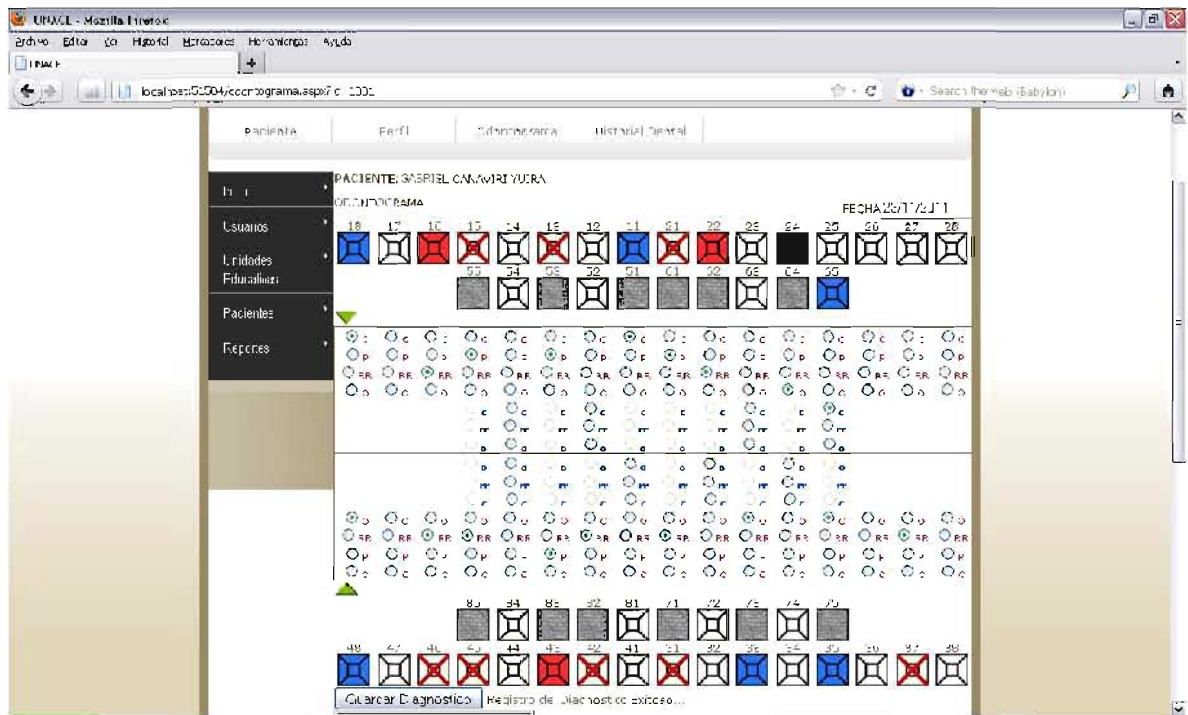


Figura 3.27 Odontograma de Paciente 1

Fuente: Elaboración: Propia

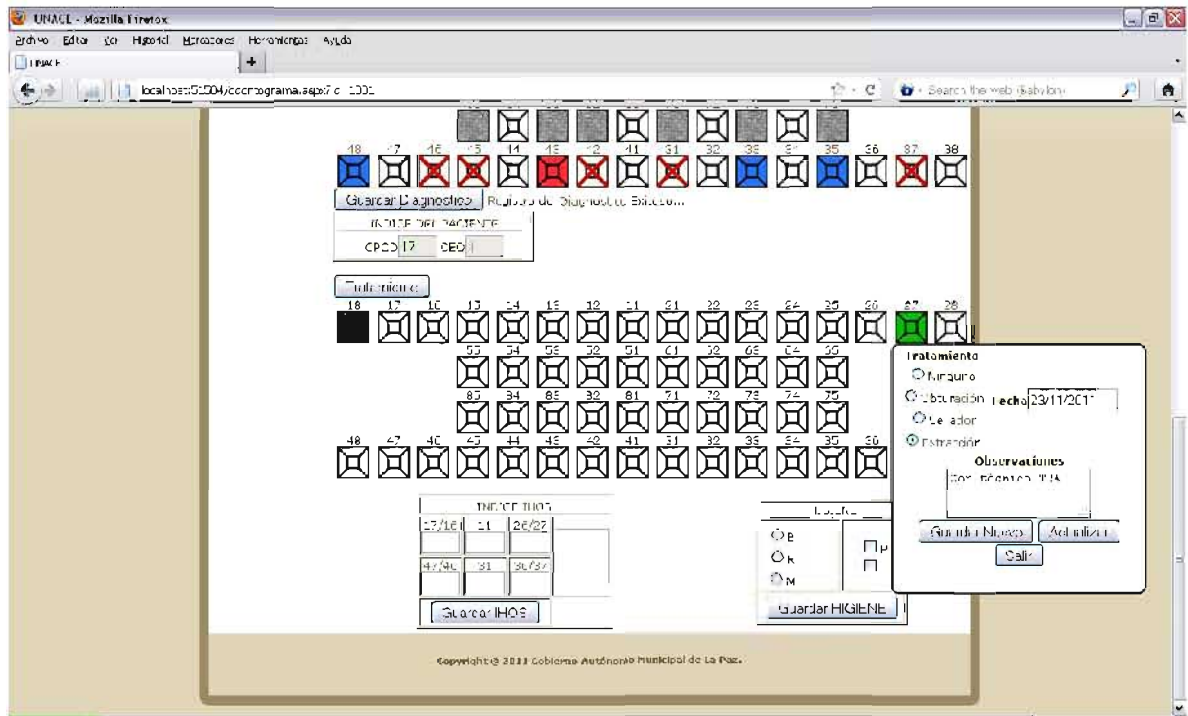


Figura 3.28 Odontograma de Paciente 2

Fuente: Elaboración: Propia

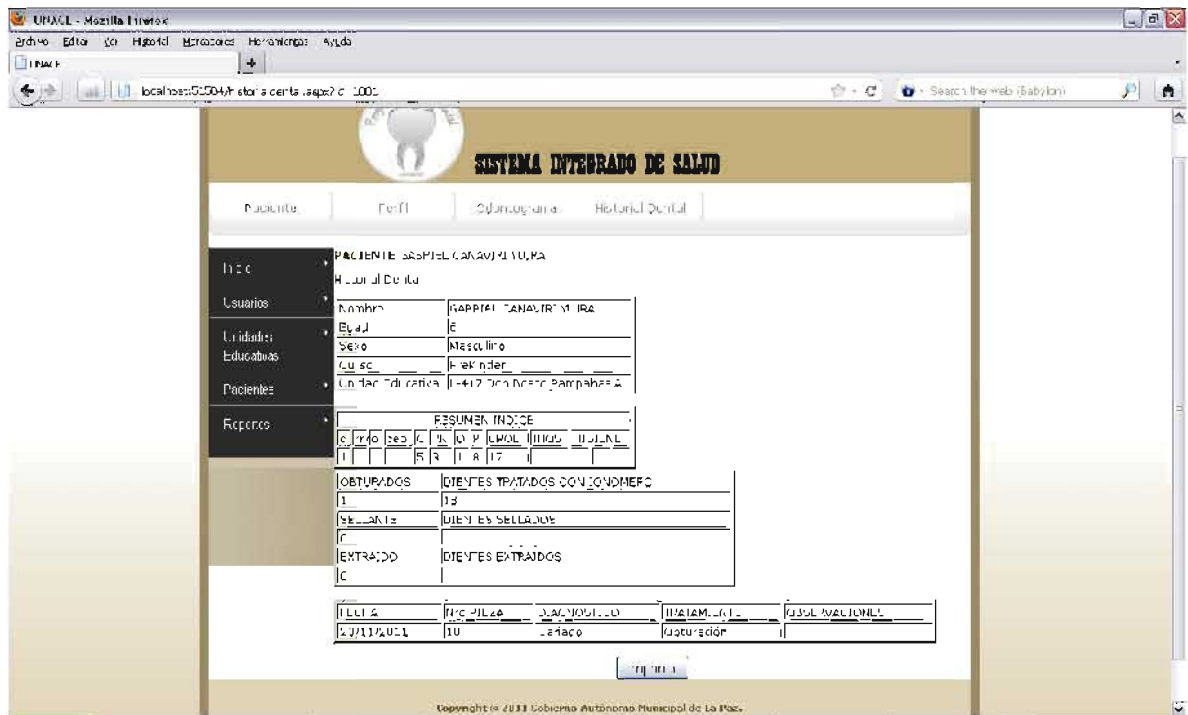


Figura 3.29 Reporte de Índices Odontológico de Paciente

Fuente: Elaboración: Propia

3.11 SEGURIDAD

Se debe tomar en cuenta que la seguridad en la informática abarca los conceptos de seguridad física, seguridad lógica y seguridad de base de datos.

3.11.1 Seguridad Física

La Seguridad Física, que se refiere a la protección del hardware y de los soportes de datos, así como a la de los edificios e instalaciones que los albergan. Contempla las situaciones de incendios, sabotajes, robos, catástrofes naturales, etc. Por tanto se debe tener en cuenta los ambientes de instalación de los sistemas, tener un respaldo para los cortes de energía eléctrica, colocar extinguidores contra incendios, etc.

3.11.2 Seguridad Lógica

La Seguridad Lógica, que se refiere a la seguridad del uso del Software, a la protección de los datos, procesos y programas, así como la del ordenador y autorizado acceso a los usuarios a la información.

3.11.3 Cifrado de la base de datos

Una de las grandes puntos de mejora de SQL Server 2008, es la seguridad y nivel de protección de las bases de datos utilizando la tecnología Transparent Data Encryption (TDE) que permite realizar el encriptado de una base de datos completa frente a la alternativa de encriptado a nivel de columna que tenía SQL Server 2005.

El GAMLP tiene una política de realización de backups que se depositan en una carpeta compartida en red con un cierto grado de seguridad, de manera que dicha copia de seguridad es accesible y puede ser restaurado de una forma sencilla. En la Figura 3.30 se muestra el código de encriptación para la base de datos SIS (Sistema Integrado de Salud), ya

que al estar expuesta en la red sin encriptación, existe una alta probabilidad de obtener la copia de seguridad de la base de datos y utilizarla con fines fraudulentos.

```
USE master;
GO

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'pa$$@word1';
GO

CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'Mi Certificado DEK';
GO

USE SIS
GO

CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
GO

ALTER DATABASE SIS
SET ENCRYPTION ON
GO
```

Figura 3.30 Implementación del cifrado DTE a la base de datos

Fuente: Elaboración: Propia

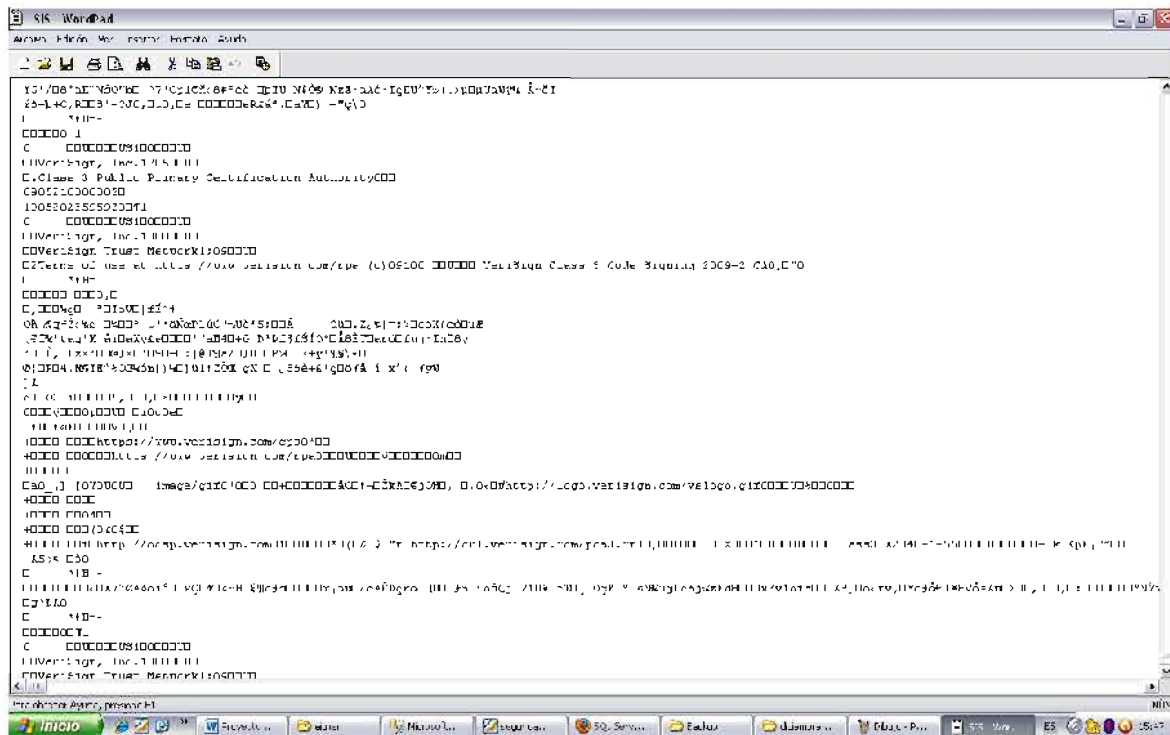


Figura 3.31 Copia de seguridad encriptado

Fuente: Elaboración: Propia

3.11.4 Políticas de seguridad

Como parte de la etapa de cierre del proyecto se propusieron las siguientes políticas de seguridad para el sistema, estas se tomaron en base a los requerimientos de la institución y a la bibliografía revisada. Las siguientes recomendaciones corresponden a la seguridad física del sistema:

- Se recomienda mantener al servidor Web en un lugar físico seguro, para que los usuarios no autorizados no puedan tener acceso a él.
- Se debe recomendar a los usuarios mantener su contraseña y nombre de usuario como un secreto y guardarlos de forma segura. En caso de olvido, vía correo electrónico puede solicitar al administrador del sistema que le recuerde sus datos de usuario. En el caso de que se sospeche una suplantación el usuario puede solicitar la eliminación de su cuenta y registrarse nuevamente con un nombre de usuario y contraseña diferentes.

También se ponen a consideración del administrador del Servidor Web las siguientes recomendaciones que son parte de la seguridad lógica del sistema:

- Realizar copias de seguridad encriptados de la base de datos semanalmente.
- Se debe proteger el servidor Web y todos los demás equipos de la misma red con contraseñas rigurosas.
- Se recomienda al administrador del sistema que la carpeta de administrativos funcione fuera del alcance de otros usuarios, y que los administrativos puedan acceder al sistema mediante intranet.
- Proteger la base de datos, ejecutando el servidor de base de datos únicamente con los privilegios necesarios.
- Cerrar los puertos que no se utilizan y desactivar los servicios no usados.
- Ejecutar en el Servidor un programa Firewall que controle el flujo de información y el envío de archivos hacia el Servidor.

CAPÍTULO IV

CALIDAD DE SOFTWARE

4 CALIDAD DE SOFTWARE

El control de calidad del software se refiere a una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del ciclo de desarrollo para asegurar que cada producto cumpla con los requisitos que han sido asignados.

4.1 METRICAS DE EVALUACIÓN DE CALIDAD WEB (WEB-SITE QEM)

4.1.1 Criterios de Usabilidad

➤ Tiempos de respuesta

Ante la mayoría de solicitudes que realice el usuario a través del sistema, el mismo responderá en un tiempo no mayor a 3 segundos (se verificó este valor en la implementación del sistema en la intranet del Gobierno Autónomo Municipal de La Paz, servidores de aplicación, bases de datos y una máquina cliente que accedía a través de un navegador Web); existe un pequeño incremento en el tiempo (hasta 3 segundos adicionales), cuando se consideran transacciones que involucren generación de reportes. El tamaño promedio de las páginas es de 35 Kbytes, el cual se encuentra por debajo del valor referencial (Kappel et al., 2006). Los tiempos de respuestas también se ven beneficiados con la aplicación de la paginación en los formularios de consultas.

➤ Eficiencia de interacción

Esta característica se consigue en el sistema, mediante 1) la presentación en todo momento, del menú principal, que disminuye la distancia para regresar a un lugar específico del sistema, 2) disminución de la necesidad de cambio entre el uso del mouse y del teclado, mediante la configuración adecuada del orden de tabulación de los controles y 3) necesidad de hasta 4 clicks para localizar un elemento en el sistema.

➤ **Colores**

El sistema presenta armonía adecuada de colores, tiene 5 colores básicos que se presentan a lo largo de sus páginas: café oscuro, café claro, negro (estos tres colores utilizados para fondos) amarillo, blanco y negro (estos colores utilizados para las fuentes). Estos colores identifican claramente las principales secciones de la mayoría de las páginas, fondo global, fondo de secciones y fondo de barras de menú de herramientas.

➤ **Disposición del texto**

Las fuentes empleadas son del tipo Verdana y Arial (aquellas que no contienen terminaciones curvas), para facilitar la lectura del texto mostrado. El color de las fuentes es negro la mayoría de las veces, el cual es legible perfectamente sobre el color café claro del fondo. El tamaño de las fuentes no presenta grandes variaciones en las diferentes secciones, su tamaño está entre 11 pt y 12 pt.

➤ **Estructura de navegación**

Los sitios a los cuales puede acceder el usuario, están dados por la estructura del Menú Principal, dado que este menú siempre es visible, el usuario no tiene dificultades en recordar los sitios que visita, tarea que además se ve facilitada por la “Ruta de Navegación” (también conocida como Breadcrumbs) que se muestra en la parte superior de las páginas y que indica al usuario la ruta que ha seguido desde la página de inicio llamada “Inicio”.

➤ **Multiculturalidad**

El principal aspecto en este punto tiene que ver con el idioma que utiliza el sistema. El idioma español es el que se empleará; sin embargo, se considera la opción de ofrecerlo también en idioma inglés en un trabajo que corresponderá a un proyecto futuro.

4.1.2 Eficiencia

Describe la relación entre el nivel de rendimiento de un producto de software y los recursos que utiliza bajo condiciones específicas. En este punto, debido a que el sistema es nuevo, no se tienen mediciones reales de análisis de carga en un ambiente de producción,

sino solo aquellas de un ambiente de prueba, sin embargo, éstas últimas nos pueden dar una percepción de cómo operaría el sistema en la Intranet. Los datos de los recursos que utiliza el sistema, se revisarán más adelante, cuando se traten las Pruebas.

4.1.3 Mantenimiento

Describe el esfuerzo requerido para implementar cambios predeterminados en un producto de software. El mantenimiento del sistema Web, se ve facilitado por los siguientes aspectos: 1) El sistema se encuentra desarrollado en capas, lo cual permite realizar cambios en un solo sector, sin afectar al resto del sistema, 2) las clases de las librerías tienen nombres totalmente descriptivos, lo que facilita la localización de métodos o propiedades dentro de las clases, 3) Existe un Manual de Usuario que permite comprender el funcionamiento del sistema y, 4) Los diferentes modelos de la Ingeniería del sistema están disponibles y pueden ser entendidos sin problema, ya que existe una trazabilidad adecuada entre uno y otro (no son modelos aislados).

4.1.4 Portabilidad

Describe la facilidad de un producto de software para ser trasladado de un ambiente a otro. Con respecto a este punto, el sistema es portable dentro de los siguientes escenarios: por un lado, el hardware de los equipos utilizados como servidores debe ser de al menos 512 MB en memoria RAM, procesador de 1.30 GHz, y 10 GB de espacio libre en disco para la instalación de los sistemas operativos; en cuanto al software, el sistema es portable dentro de los siguientes productos del fabricante Microsoft:

- Servidor Web: Internet Information Services (IIS) 7.0 sobre Windows Server 2008 o IIS 6.0 instalado sobre Windows XP Professional (en el último caso, las conexiones concurrentes no deberán superar las 10).
- Servidor de desarrollo: ASP NET de Visual Studio 2010. El sistema no ha sido probado en Visual Studio 2008.
- Base de datos: SQL Server 2008 R2. El sistema no ha sido probado sobre SQL Server 2005, ni sobre la versión SQL Express incluida en Visual Studio 2008.

- Crystal Reports para el IDE de Visual Studio 2010, para la generación de reportes.
- Navegador Web: el desarrollo se lo hizo tomando como referencia, una visualización correcta obtenida con Internet Explorer 8.0 (habilitada la opción de vista de compatibilidad). El sistema también puede ser mostrado en las versiones 6.0 y 7.0 de IE y Sobre Mozilla Firefox 3.0 o superior; sin embargo, la visualización de las páginas puede variar.

A continuación se presenta la selección y representación de las características y atributos para este trabajo. En la tabla 4.1 se muestra el árbol de requerimientos de calidad para el dominio de la aplicación Web.

Tabla 4.1 Árbol de requerimientos de calidad para el dominio de la aplicación Web para el Gobierno Autónomo Municipal de El Alto

1. Usabilidad	2. Funcionalidad
1.1 Comprensibilidad Global del Sitio	2.1 Aspectos de Búsqueda y Recuperación
1.1.1 Esquema de Organización Global	2.1.1 Mecanismo de Búsqueda en el Sitio
1.1.1.1 Mapa del Sitio	2.1.1.1 Búsqueda Restringida
1.1.1.2 Tabla de Contenido	2.1.1.2 Búsqueda Global
1.2 Mecanismos de ayuda y retroalimentación en línea	2.2 Aspectos de Navegación y Exploración
1.2.3 Directorio de Direcciones	2.2.1 Navegabilidad
1.2.3.1 Directorio E-mail	2.2.1.1 Orientación
1.2.3.2 Directorio TE-Fax	2.2.2 Objetos de Control Navegacional
1.2.3.3 Directorio Correo Postal	2.2.2.1 Nivel de Desplazamiento
1.3 Aspectos de Interfaces y Estéticos	2.2.2.2.1 Desplazamiento Vertical
1.3.1 Cohesividad al Agrupar los Objetos de Control Principales	2.2.2.2.2 Desplazamiento Horizontal
1.3.2 Permanencia y Estabilidad en la Presentación de los Controles Principales	2.2.3 Predicción Navegacional
1.3.2.1 Permanencia de Controles Directos	2.2.3.1 Enlace con Título (enlace con texto explicatorio)
1.3.2.2 Permanencia de Controles Indirectos	2.2.3.2 Calidad de la Frase del Enlace
	2.3 Aspectos del Dominio orientados al Usuario o Internauta

1.3.2.3 Estabilidad 1.3.3 Aspectos de Estilo 1.3.3.1 Uniformidad en el Color de Enlaces 1.3.3.2 Uniformidad en el Estilo Global 1.3.4 Uniformidad en el Estilo del sitio	2.3.1 Relevancia de Contenido 2.3.1.1 Información de la Institución 2.3.1.2 Información de programas de Salud 2.3.1.3 Noticias 2.3.1.4 Información de Índices 2.3.1.5 Información de actividades sociales
3. Confiabilidad	4. Eficiencia
3.1 No Deficiencia 3.1.1 Errores de Enlaces 3.1.1.1 Enlaces Rotos 3.1.1.2 Enlaces Inválidos 3.1.1.3 Enlaces no Implementados 3.1.2 Errores o Deficiencias Varias 3.1.2.1 Deficiencias o cualidades ausentes debido a diferentes navegadores (browsers) 3.1.2.2 Deficiencias o resultados inesperados independientes de browsers (p.ej. errores de búsqueda imprevistos, deficiencias con marcos (frames), etc.) 3.1.2.3 Nodos Destinos (inesperadamente) en Construcción 3.1.2.4 Nodos Web Muertos (sin enlaces de retorno)	4.1 Accesibilidad 4.1.1 Accesibilidad de Información 4.1.1.1 Soporte a Versión sólo Texto 4.1.1.2 Legibilidad al desactivar la Propiedad Imagen del Browser 4.1.1.2.1 Imagen con Título 4.1.1.2.2 Legibilidad Global 4.1.2 Accesibilidad de Ventanas 4.1.2.1 Número de Vistas considerando Marcos (frames) 4.1.2.2 Versión sin marcos 4.2 Performancia 4.2.1 Páginas de Acceso Rápido

Fuente: Elaboración Propia

❖ Tipo de Criterio Elemental

CVN: $IE = (X/Y) * 100$ con $X = \Sigma$ Puntaje Máximo $Y = \Sigma$ Puntaje Obtenido

CB: $IE = 0$ si no existe $IE = 1$ Si Existe

CPD: Sujeto a la Objetividad del Observador

CMN: $IE = 0 \approx 0$ Ausente $IE = 1 \approx 60$ Presencia Parcial $IE = 2 \approx 100$ Presente

Dónde:

CVN=Criterio de variable Normalizada

CB=Criterio Binario

CPD=Criterio de Preferencia Directa

CMN=Criterio de Multi-Nivel

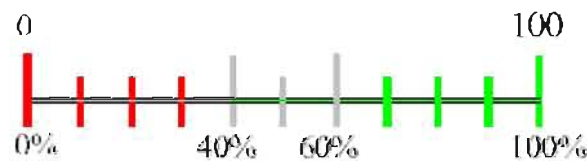
❖ Escala de Medición de Aceptabilidad

Insatisfactorio (0 – 40) %

Aceptabilidad Marginal (40-60) %

Satisfactorio (60 - 100) %

Figura 4.1 Escala de Preferencia



Fuente: [OLS06]

Tabla 4.2 Desarrollo de la Metodología con los Pesos de Agregación

Nombre	Criterio	Pref.
1. Usabilidad	CVN	76,53
1.1 Compresibilidad Global del sistema	CVN	54,17
1.1.1 Esquema de Organización Global	CVN	33,33
1.1.1.1 Mapa del Sitio	CB	1≈100
1.1.1.2 Tabla de Contenidos	CB	0≈0
1.1.1.3 Índice Alfabético	CB	0≈0
1.1.2 Calidad en el Sistema de Etiquetado	CPD	75

1.2 Mecanismos de ayuda y retroalimentación en línea	CVN	91,67
1.2.1 Calidad de la ayuda	CVN	75
1.2.1.1 Ayuda Explicatorio Orientada al Visitante	CPD	75
1.2.1.2 Ayuda de la Búsqueda	CPD	75
1.2.3 Directorio	CVN	100
1.2.3.1 Directorio de Unidades	CB	1≈100
1.2.3.2 Directorio de Responsables	CB	1≈100
1.2.3.3 Directorio misiones	CB	1≈100
1.2.4 Facilidad FAQ	CMN	1≈100
1.3 Aspectos de Interfaces y Estéticos.	CVN	83,75
1.3.1 Cohesividad al agrupar los objetos de control principal	CPD	65
1.3.2 Permanencia y estabilidad en la presentación de los controles Principales	CVN	80
1.3.2.1 Permanencia de controles directos	CPD	80
1.3.2.2 Permanencia	CPD	80
1.3.2.3 Estabilidad	CPD	80
1.3.3 Aspectos de Estilos	CVN	100
1.3.3.1 Uniformidad en el color de enlaces	CMN	2≈100
1.3.3.2 Uniformidad	CMN	2≈100
1.3.4 Preferencia Estética.	CPD	90
2. Funcionalidad	CVN	75,74
2.1 Aspectos de Recuperación y Búsquedas	CVN	80
2.1.1 Mecanismo de Búsqueda en el sistema	CVN	80
2.1.1.1 Búsqueda Restringida	CVN	100
2.1.1.1.1 De activos Fijos	CB	1≈100
2.1.1.1.2 De Asignados, Revaluados, Bajas, etc	CB	1≈100
2.1.1.2 Búsqueda Global	CMN	1≈60
2.1.2 Mecanismos de Recuperación	CVN	80
2.1.2.1 Nivel de Personalización	CMN	2≈100

2.1.2.2 Nivel de retroalimentación en la recuperación	CMN	1≈60
2.2 Aspectos de Navegación y Exploración	CVN	61,67
2.2.1 Navegabilidad	CVN	50
2.2.1.1 Orientación	CVN	100
2.2.1.1.1 Indicador del camino	CB	1≈100
2.2.1.1.2 Etiqueta de la posición actual	CB	1≈100
2.2.1.2 Promedio de enlaces por pagina	CMN	0≈0
2.2.2 Objetos de control navegacional	CVN	55
2.2.2.1 Permanencia y estabilidad en la presentación de los controles (Subsitio)	CVN	60
2.2.2.2.1.1 Permanencia de controles contextuales	CMN	1≈60
2.2.2.2.1.2 Estabilidad	CMN	1≈60
2.2.2.2 Nivel de desplazamiento	CVN	50
2.2.2.2.1 Desplazamiento vertical	CB	1≈100
2.2.3 Predicción de navegacional	CVN	80
2.2.3.1 Enlace con título (enlace con texto explicatorio)	CMN	2≈100
2.2.3.2 Calidad de la frase explicatorio	CMN	1≈60
2.3 Aspectos de Dominio Orientado al Visitante	CVN	85,55
2.3.1 Relevancia de Contenido	CVN	85,56
2.3.1.1 Información de Unidades, direcciones, etc	CVN	66,67
2.3.1.1.1 Índice de las unidades	CB	1≈100
2.3.1.1.3 Informaciones de Representaciones en el Exterior	CB	1≈100
2.3.1.2 Información de Activos Fijos	CVN	100
2.3.1.2.1 Formulario de asignación, revaluós, bajas	CMN	2≈100
2.3.1.3 Información de Activos Fijos en Reportes	CVN	90
2.3.1.3.1 Actualización y Depreciación de Activos Fijos	CMN	1≈60
2.3.1.3.2 Información de Asignaciones	CMN	2≈100
2.3.1.3.3 Información de Revaluós	CMN	2≈100
2.3.1.3.4 Información Bajas	CMN	2≈100

3. Confiabilidad	CVN	80
3.1 No Deficiencia	CVN	80
3.1.1 Errores de enlaces	CVN	100
3.1.1.1 Enlaces rotos	CMN	2≈100
3.1.1.2 Enlaces inválidos	CMN	2≈100
3.1.1.3 Enlaces no implementados	CMN	2≈100
3.1.2 Errores o Deficiencias Varias	CVN	60
3.1.2.1 Deficiencias o cualidades ausentes debido a diferentes navegadores(Browser)	CMN	1≈60
3.1.2.3 Nodos destinos (inesperadamente) en construcción	CMN	1≈60
3.1.2.4 Nodos Web muertos (sin enlaces de retorno)	CMN	1≈60
4. Eficiencia	CVN	72,5
4.1 Performancia	CVN	95
4.1.1 Paginas de Acceso Rápido	CPD	95
4.2 Accesibilidad	CVN	50
4.2.1 Accesibilidad de la Información	CVN	50
4.2.1.1 Soporte a Versión solo texto	CB	0≈0
4.2.1.2 Legibilidad al desactivar propiedad de imagen browser	CVN	100
4.2.1.2.1 Imagen con titulo	CB	1≈100
4.2.1.2.2 Legibilidad Global	CB	1≈100

Fuente: [OLS06]

La tabla 4.3 muestra un resumen de los datos obtenidos a partir de la tabla anterior (tabla 4.2)

Tabla 4.3 Resumen de los Resultados Obtenidos

NOMBRE	PREFERENCIA ELEMENTAL DE ACTIVOS FIJOS
Usabilidad	76.53 %
Funcionalidad	75.74 %
Confiabilidad	80.00 %
Eficiencia	72.50 %
Calidad Global	76.20 %

Fuente: Elaboración Propia

Por medio de la evaluación de la calidad de artefactos Web, podemos comprender el grado de cumplimiento de un conjunto de características, sub características y atributos con respecto a los requerimientos de calidad establecidos.

En este contexto observamos y concluimos que el Sistema Integrado de Salud en entorno Web, los valores (Usabilidad, Funcionalidad, Confiabilidad, Eficiencia y la Calidad Global) obtenidos se encuentran en el rango de 60 a 100 %. La cual nos indica que tiene un nivel de aceptabilidad satisfactorio, de los requerimientos de los diferentes perfiles de usuario.

4.2 PRUEBAS

4.2.1 Revisión del Contenido

La primera actividad que se realizó como parte de las pruebas de la aplicación Web fue la revisión del contenido de las páginas. Para esto se utilizó el corrector ortográfico de Adobe Dreamweaver CS5. Se encontraron errores ortográficos y gramaticales que fueron corregidos inmediatamente.

4.2.2 Pruebas de Navegación

Como segunda actividad se verificaron los enlaces y su correspondencia en busca de errores de navegación en la Aplicación Web. Obteniendo resultados satisfactorios pues todos los enlaces correspondían correctamente.

4.2.3 Pruebas de enlaces

Estas pruebas se realizaron en el sistema para verificar que: 1) no existan enlaces en las páginas del sistema que no conduzcan a ningún lugar y, 2) no existan páginas de las cuales no se pueda regresar.

4.2.4 Pruebas de usabilidad

Estas pruebas permitieron identificar las características de usabilidad del sistema ya presentadas anteriormente. Un aspecto aún pendiente por implementar en el sistema, es el referente al acceso al mismo por parte de personas con problemas visuales, auditivos o cognitivos, cuyos aspectos son cubiertos por WAI (Web Accesibility Initiative o Iniciativa de Accesibilidad para Web) de W3C (World Wide Web Consortium o Consorcio World Wide Web).

4.2.5 Pruebas de seguridad

Estas pruebas están destinadas a verificar los niveles de seguridad con que cuenta el sistema. Cabe indicar que en este aspecto, queda aún mucho por hacer dentro del sistema; sin embargo, dado que el objetivo de este proyecto no está centrado en seguridades, se mencionará únicamente aquellas seguridades que tienen que ver con la aplicación como tal y no con las comunicaciones a través de la red o con permisos de acceso a los servidores. Las seguridades con las que contará el sistema, son: 1) validación de los datos ingresados por los usuarios en los formularios de ingreso de datos, 2) asignación de perfiles a los usuarios, para

restringir el acceso a los componentes del sistema, 3) formularios de ingreso de credenciales de usuario (nombre de usuario y contraseña).

4.2.6 Pruebas Unitarias (Caja Blanca)

Esta etapa consistió en realizar las pruebas de unidad a cada uno de los módulos de la aplicación Web que contenían código dinámico. Para esto se utilizó la técnica del camino básico. Los resultados de las pruebas fueron los siguientes:

Tabla 4.4 Pruebas Unitarias

Archivo	Complejidad ciclomática	Camino básico	Resultados
Eliminar_Usuario.aspx.cs	eliminar_usuario() $V(G) = 2$	1. La consulta se realiza correctamente. 2. Ocurre un error en la ejecución de la	1. El usuario se da de baja, en la base de datos se pone en un estado inactivo. 2. Se envía un
Eliminar_UnidadE.aspx.cs	eliminar_UnidadE() $V(G) = 2$	1. La consulta se realiza correctamente. 2. Ocurre un error en la ejecución de la consulta.	1. La Unidad Educativa se da de baja, en la base de datos se pone en un estado inactivo.
Eliminar_odontograma.aspx.cs	eliminar_odontograma() $V(G) = 4$	1. La consulta se realiza correctamente sí, el odontograma es un registro existente. 2. La consulta se realiza correctamente, pero no se actualizó el estado de inactivo. 3. La consulta se realiza correctamente y el estado se actualiza correctamente. 4. ocurrió un error al realizar la consulta.	1. Se elimina el odontograma de la base de datos y se actualiza el estado del paciente. 2. Se elimina el odontograma de la base de datos pero el estado del paciente no se actualiza. 4. Se envía un mensaje de error.

Fuente: Elaboración Propia

4.3 ANÁLISIS DE COSTO

Como su nombre lo sugiere, el método de análisis Beneficio/Costo está basado en la razón de los beneficios a los costos asociada con un proyecto en particular. Se considera que un proyecto es atractivo, cuando los beneficios derivados de su implementación exceden sus costos asociados. Por tanto, el primer paso en un análisis Beneficio/Costo es determinar cuáles de los elementos son costos y cuales beneficios.

Análisis de costos

Para determinar el costo total del proyecto se tomará en cuenta los siguientes costos:

- Costo del Software desarrollado.
- Costo de implementación del Sistema. conformado por:
 - Costo de software
 - Costo de hardware.
- Costo de elaboración del proyecto.

Costo del software desarrollado

Para la determinación del costo del Software desarrollado, se hará uso del Modelo Constructivo de Costo COCOMO II, orientado a los Puntos de Función.

a) Cálculo de Puntos de Función no ajustados:

Tabla 4.5 Puntos de función

Tipo	Descripción	
Entradas de Usuario	Actualización de Usuario Actualización de Historial Clínico Actualización de Odontograma Pantalla de autenticación	Pantalla de registro de Usuario Pantalla de registro de Historial Clínico Pantalla de registro de Odontograma

Salidas de Usuario	Resultados de Actualizaciones Usuario Resultados de Actualizaciones Historial Clínico Resultados de Actualizaciones Odontograma	Reportes
Peticiones de Usuario	Menú de usuario Menu de administrador	Ventana Paciente Ventana Selección de Paciente
Archivos	Tabla de Usuario Tabla de Paciente	Tabla de Respuesta Tabla de Mensaje
Interfaces externas		Ninguna

Fuente: Elaboracion Propia

Ahora se procede a contabilizar los puntos de función para hallar el total:

Tabla 4.6 Puntos de función no ajustados

PARAMETROS DE MEDICION	CUENTA	FACTOR DE PONDERACION	TOTAL
Numero de Entradas de Usuario	17	4	65
Numero de Salidas de Usuario	9	5	45
Numero de Peticiones de Usuario	4	4	16
Numero de archivos	5	10	50
Numero de Interfaces externos	0	7	0
Cuenta Total			176

Fuente: Elaboracion Propia

b) Cálculo de valores de ajuste de la complejidad.

Los valores son respondidos usando una escala desde 0 (no importante o aplicable) hasta 5 (absolutamente esencial).

Tabla 4.7 Determinación de Complejidad.

FACTOR	VALOR
Copia de seguridad y recuperación	5
Comunicación de datos	5
Proceso distribuido	3
Rendimiento crítico	3
Entorno operativo existente	5
Complejidad del procesamiento interno	5
Código diseñado para la reutilización	5
Conversión/instalación en diseño	2
Instalaciones múltiples	5
Aplicación diseñada para el cambio	5
$\Sigma (Fi)$	43

Fuente: Elaboracion Propia

$$\text{Factor de Ajuste} = 0.65 + 0.01 * \Sigma (Fi)$$

$$\text{Factor de Ajuste} = 0.65 + 0.01 * 43$$

$$\text{Factor de Ajuste} = 1.08$$

c) Cálculo de Puntos de Función

El cálculo de Puntos de Función se basa en la fórmula:

$$PF = \text{Cuenta Total} * \text{Factor de Ajuste}$$

$$PF = 176 * 1.08$$

$$PF = 190,08 = 191$$

d) Conversión de los Puntos de Función a KLDC

Ahora se debe convertir los Puntos de Función a miles de líneas de código. Tomar en cuenta la siguiente tabla.

Tabla 4.8 Conversión de Puntos de Función a KLDC

LENGUAJE	NIVEL	Factor LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL/I	4	80
Ansi Cobol 74	3	107
Visual Basic	7.00	46
ASP	9.00	36
PHP	11.00	29
Visual C++	9.50	34

Fuente: [PRE06]

$$\text{LDC} = \text{PF} * \text{Factor LDC/PF}$$

$$\text{LDC} = 191 * 46$$

$$\text{LDC} = 8786$$

$$\text{KLDC} = (8700/1000) = 8.8$$

e) Aplicación de las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido.

Las ecuaciones del COCOMO básico tienen la siguiente forma:

$$E = a_b (KLDC)^{b_b} \dots\dots\dots (\text{Ecuación } 1)$$

$$D = c_b (E)^{d_b} \dots\dots\dots (\text{Ecuación } 2)$$

Donde:

E: Esfuerzo aplicado en personas por mes

D: Tiempo de desarrollo en meses cronológicos

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

Tabla 4.9 Coeficientes a_b y c_b y los exponentes b_b y d_b

Proyecto de Software	a_b	c_b	b_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semi – acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Fuente: [PRE06]

Se toma el caso del software Empotrado, por su complejidad en programación, robustez en la base de datos e Interfaz de Usuario (UI).

$$E = 3,0 * 8,8^{1,20}$$

$$E = 36.38$$

$$D = 1,20 * 36.38^{0,32}$$

$$D = 3.79 \approx 4$$

El personal requerido, en este caso el número de programadores se obtiene con la siguiente fórmula:

$$\text{Número de programadores} = E / D$$

$$\text{Número de programadores} = 36.38 / 3.79$$

$$\text{Número de programadores} = 9.59 \approx 10$$

Según consultoras desarrolladores de software, el salario de un programador puede oscilar entre los \$us 200 a 300 inicialmente, cifra que será tomada en cuenta para la estimación siguiente:

$$\text{Costo del software desarrollado} = \text{Número de programadores} * \text{salario de un programador}$$

$$\text{Costo del software desarrollado por persona} = 10 * 200$$

$$\text{Costo del software desarrollado por persona} = 2000$$

$$\text{Costo total del software desarrollado} = 2000 * 4$$

Costo total del software desarrollado = 8000 \$us

Costo de implementación del sistema

Se observa que las herramientas del software son de GPL o de uso libre y también el uso del internet para la difusión del sistema.

Tabla 4.10 Costo Software Express

Descripción	Costo total (\$us)
Hosting	27.9070 = 27 (anual)

Fuente: Hastle Hosting

Costo de elaboración del proyecto

Los costos de elaboración del proyecto se refieren a los costos del estudio del sistema, en la etapa de recopilación y análisis principalmente, estos costos se presentan en la siguiente tabla.

Tabla 4.11 Costo Elaboración del proyecto

DETALLE	IMPORTE (\$us)
Análisis y diseño del proyecto	300
Material de Escritorio	30
Internet	40
Otros	20
TOTAL	440

Fuente: Elaboración propia

Costo total

El costo total es la sumatoria del costo del software desarrollado, costo de implementación del Sistema y costo de elaboración del proyecto, detallados en la siguiente tabla 4.12.

Tabla 4.12 Costo total

DETALLE	IMPORTE (\$US)
Costo del Software desarrollado	8000
Costo de implementación del Sistema	27
Costo de elaboración del proyecto	440
Total	8467

Fuente: Elaboracion propia

Por todo esto se observa que el costo más alto es el “costo del software desarrollado”, y no tanto así los demás costos.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

La metodología UWE optimizó el tiempo de desarrollo del Sistema Integrado de Salud vía web, ya que esta metodología permite realizar cambios en el sistema sin tener ningún problema, organizando de una mejor manera los flujos de trabajo de: requerimientos, análisis, diseño, implementación y pruebas dando un resultado óptimo en el desarrollo del sistema. Por tanto se logró alcanzar con los objetivos propuestos llegando a las siguientes conclusiones:

- ❖ El sistema desarrollado es capaz de reemplazar al sistema manual en la mayoría de los procesos.
- ❖ Se implementó políticas de seguridad para el acceso de la información.
- ❖ La ventaja de aplicar la metodología UWE es la utilización de un lenguaje de modelamiento ampliamente extendido como lo es UML.
- ❖ La interfaz de usuario se acogió sin inconvenientes por el usuario final, de manera que tiene fácil acceso a la información, realizando consultas y reportes
- ❖ Desarrollar sistemas para Web conlleva mayores consideraciones en cuanto a la seguridad, ya que el sistema puede eventualmente estar expuesto en Internet, lo cual ya implica un riesgo mayor al de los sistemas tradicionales de software.
- ❖ El sistema desarrollado se puso a disposición de personas dentro del sector, para recoger su opinión. Los principales resultados obtenidos, muestran que:
 - 1) el 80% de los usuarios piensa que el sistema apoya en sus actividades laborales.
 - 2) el 80% de los usuarios considera fácil la utilización del sistema.
 - 3) el 90% de los usuarios piensa que es adecuada la apariencia visual del sistema (colores, fuentes, utilización de espacios en la pantalla, controles, etc.).

En conclusión se logró el desarrollo del “SISTEMA INTEGRADO DE SALUD” a partir del Sistema de Información Odontológico el cual permite agilizar los procesos de información, trabajo operativo facilitando el manejo de información y reducción de tiempo.

5.2 RECOMENDACIONES

En base al trabajo realizado, podemos establecer las siguientes recomendaciones:

- ❖ Ampliar la implementación del sistema con el programa de Alimentación Complementaria Escolar, programa encargado de la repartición del desayuno escolar en todo el Municipio de La Paz.
- ❖ Se recomienda definir desde un inicio los requerimientos de la manera más exacta posible, utilizando para ello: diagramas de casos de uso y sobre todo, una descripción clara de dichos casos.
- ❖ Se recomienda la aplicación de la Metodología UWE, en conjunto con otra metodología ágil, para así superar las deficiencias de cada una con las fortalezas de la otra, pues como indicamos, UWE está orientada hacia aspectos de navegación y presentación y no de estructura.
- ❖ Se recomienda capacitar al personal nuevo que ingrese a la Unidad, con el tutor interactivo realizado para el manejo del sistema.
- ❖ La seguridad lógica debe estar a cargo del administrador del servidor Web. Debe incluir el uso de contraseñas, la administración de puertos y servicios del servidor, el control de los archivos y contenidos, la ejecución de un Firewall y la administración de la base de datos.

Bibliografía

FUENTES DE INFORMACIÓN

Referencias Bibliográficas

[CAR00] Carrera de Nutrición y Dietética, 2000: PROYECTO DE EVALUACIÓN NUTRICIONAL, Facultad de Medicina – UMSA.

[UNI02] Universitat Jaume I, 2002: Especificación de Requisitos Software según el estándar de IEEE830.

[WEI04] Weitzenfeld A., 2004: Ingeniería de Software Orientada a Objetos con UML, Java e Internet, Thomson Learning.

[Pressman03] Pressman R, 2003: INGENIERÍA DEL SOFTWARE: Un enfoque práctico. 5ta. Edición, Editorial: McGraw Hill/Interamericana, España.

[IEEE 729-1983] Glossary of Software Engineering Terminology - Redesignated as IEEE 610.12, Institute of Electrical and Electronics Engineers.

[Tapper09] Tapper, J., Labriola, M. Boles, M. & Talbot, J. Adobe Flex 3. Edición en Español por Ediciones Anaya Multimedia. 2009.

[Davis08] Davis, M. & Phillips, J. Flex 3: A Beginner's Guide. McGraw- Hill Companies. 2008.

[Farré05] Farré, X. & Messeguer, R. Rich Internet Applications. Universidad Politécnica de Cataluña. Julio, 2005.

[Martinez-Ruiz06] Martinez-Ruiz, F., Muñoz, J., Vanderdonckt, J., González-Calleros, J & Mendoza, R. "A first draft of a Model-driven Method for Designing Graphical User Interfaces of Rich Internet Applications" Proc. of the Fourth Latin American Web Congress (LA WEB'06), IEEE Computer Society, pp. 32-38, 2006.

[Pérez08] Pérez, M. & Messeguer, R. "Evaluación y prueba de aplicaciones RIA con AJAX". Universidad Politécnica de Cataluña. Abril, 2008.

[Brown08] Brown, C. The Essential Guide to Flex 3. Ed. Friends of ED. 2008.

[Van00] Van Bemel, J.H., y Musen, M.A (2000). Handbook of Medical Informatics. Bohn Stafleu Van Loghum-Springer.

[IEEE 729-1983] Glossary of Software Engineering Terminology Redesignated as IEEE 610.12, Institute of Electrical and Electronics Engineers.

- [SOM05] Sommerville, I. Software Engineering. Seventh Ed. Addison Wesley. 2005.
- [ADA04] Adams, R., Eslinger, S., Owens, K. & Rich, M. A. "Software Acquisition Best Practices 2004 edition". Proc. of the 3° OSD Conference on the Acquisition of Software-Intensive Systems, The Aerospace Corporation, January, 2004.
- [WAL03] Walker, E: "Implementing Best Practices in the Joint Battlespace Infosphere (JBI) program at AFRL". Conference on the Acquisition of Software-Intensive System, Carnegie Mellon University, January 28-30, 2003.
- [BOO00] G. Booch, I. Jacobson, J. Rumbaugh , "El Proceso Unificado de Desarrollo", Addison Wesley, 2000.
- [KOC07] Koch, N., Knapp, A., Zhang, G. & Baumeister, H. "UML-Based Web Engineering: An Approach Based on Standards". Web Engineering: Modeling and Implementing Web Applications. HCI Series, Springer-Verlag, 2007.
- [KOC02] Koch, N., Kraus, A., & Hennicker, R. "The Authoring Process of the UML-based Web Engineering Approach". Institute of Computer Science Ludwig-Maximilians University of Munich. 2002.
- [OCA04] Ocaña, J. & Rossainz, M. "Introducción a la Ingeniería Web Basada en UML". Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación. 2004.
- [PRE05] Preciado, J. C, Linaje, M., Sanchez, F. & Comai, S. "Necessity of methodologies to model Rich Internet Applications". Proc. of the Seventh IEEE International Symposium on Web Site Evolution (WSE'05), IEEE Computer Society, pp. 7-13. 2005.
- [BEC1b] Beck, K. y M. Fowler, Planning Extreme Programming, Addison-Wesley, 2001.

Artículos y Revistas

- [DAV05] Davalo Álvaro, 2005: Importancia de los sistemas de información en las áreas administrativas de contabilidad y finanzas, recursos humanos, mercadotecnia y almacén, México.
- [GOB09] Gobierno Autónomo Municipal de La Paz, 2009: Paquete Estudiantil del Bicentenario, Revista Informativa, La Paz - Bolivia.

[GOB06] Gobierno Autónomo Municipal de La Paz, 2006: Perfil Epidemiológico del Escolar, Oficialía Mayor de Desarrollo Humano, La Paz - Bolivia.

[MIN09] Ministerio de Salud y Deportes, 2009: Salud Integral Oral, La Paz – Bolivia.

[ASS86] Association of American Medical Colleges. Evaluation of medical information science in medical education. Journal of Medical Education, 61 (1986).

[HAM97] Michael Hammer y James Champy, Reingeniería, 226 paginas, año 1994.

[ASS86] Association of American Medical Colleges. Evaluation of medical information science in medical education. Journal of Medical Education. 61 (1986)

[HER01] Hernández Orallo Enrique, El Lenguaje Unificado de Modelado, 2001.

[ROS03] Gustavo Rossi, Oscar Pastor, Daniel Schwabe, Luis Olsina; Libro sobre UWE: “Web Engineering: Modelling and Implementing Web Applications”, 2003.

[TSJ11] The SolidQ™ Journal, Mayo 2011– www.solidq.com/sq.

Sitios Web

[COM02] Comité Permanente de Nutrición del Sistema de Naciones Unidas (SCN), 2002: “Nutrición: la Base para el Desarrollo”, Ginebra.

<http://www.unscn.org/layout/modules/resources/files/Bolivia2005.pdf>

[INT09] INTECO, Laboratorio Nacional de Calidad de Software, 2009: Curso de Desarrollo Ágil.

<http://empredecaminos.com/docs/%5BAgil%5DCurso%20de%20Desarrollo%20%C3%81gil.pdf>

[USE11] useit.com, 1995-2011: Jakob Nielsen's Website

www.useit.com

[CEN04] CEN/TC 251 (2005). Health Informatics-Electronic Healthcare Record Communication. 13 de Enero del 2005.

[STA08] The Standish Group International. “My Life is Failure & CHAOS Summary 2008”. Copyright 2006.

http://www.centc251.org/TCMeet/doclist/TCdoc04/N04-012prEN13606-1_2WD.doc

[URL-1] http://www.adobe.com/es/resources/business/rich_internet_apps/

Adobe: Rich Internet Applications

[URL-2] <http://www.ajax.org/>

Asynchronous Javascript and XML

[URL-3] <http://www.adobe.com/es/products/flash/>

Adobe Flash

[URL-4] <http://www.adobe.com/es/products/flex/>

Adobe Flex

[URL-5] <http://windowsclient.net/>

Windows Presentation Foundation

[URL-6] <http://silverlight.net/>

Microsoft Silverlight

[URL-FIG-1] Página oficial de Programming Extreme:

<http://www.programmingextreme.com/>

[URL-FIG-2] Página oficial de UWE: <http://uwe.pst.ifi.lmu.de/>

[CEN04] CEN/TC 251 (2005). Health Informatics-Electronic Healthcare Record Communication. 13 de Enero del 2005,

http://www.cen251.org/TCMeet/doclist/TCdoc04/N04-012prEN13606-1_2WD.doc.

[Man04] Mandirola, H.F. Weis, F. Franco, F. Nuñez, U. Ferraro, E (2004). Diferencias comparativas entre historia clínica tradicional (HCT) y la historia clínica computarizada (HCC). Marzo del 2004. Informéica. http://www.informaticamedica.org/I04/papers/mandirolabrieux_37.pdf

ANEXOS
DOCUMENTOS