

**SISTEMA DE INFORMACIÓN BASADO EN WEB PARA APOYAR LA ORGANIZACIÓN
Y EXHIBICIÓN DE LOS ESPECÍMENES ANATOMOPATOLÓGICOS
PERTENECIENTES AL MUSEO DE PATOLOGÍA UIS**

**OMAR SAÚL DUARTE BALLÉN
ÁLVARO MARTÍNEZ DUARTE**

**ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
BUCARAMANGA**

2006

**SISTEMA DE INFORMACIÓN BASADO EN WEB PARA APOYAR LA ORGANIZACIÓN
Y EXHIBICIÓN DE LOS ESPECÍMENES ANATOMOPATOLÓGICOS
PERTENECIENTES AL MUSEO DE PATOLOGÍA UIS**

OMAR SAÚL DUARTE BALLÉN

ÁLVARO MARTÍNEZ DUARTE

**Trabajo de grado para optar al título de
Ingeniero de Sistemas.**

Director

ALFONSO MENDOZA CASTELLANOS

Bachelor of Science, DEA en Automática, Informática y Robótica

Codirector

OLGA MERCEDES ÁLVAREZ OJEDA

Medica Cirujana Especialista en Patología

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS

BUCARAMANGA

2006

A Dios, por permitirme conocer que dentro del caos todo guarda perfecto orden y coherencia. Que somos parte de todo y que absolutamente todo influye en nuestro devenir.

A mis padres, por su amor y apoyo incondicional e infinito.

A Natalí, por que todo lo que aprecio pierde la mitad de su valor si no estás allí para compartirlo.

A Álvaro, por ser un amigo fiel y sincero.

Omar Saúl Duarte Ballén.

Al Creador, quien pone orden en medio del Caos.

A mis padres, de cuyos plateados cabellos, soy una de las principales causas.

A Marie, quien con sus ojos élficos y voz de ruiseñor, dio sentido a mi existencia.

A Omar, por ser un gran compañero de proyecto.

Álvaro Martínez Duarte.

AGRADECIMIENTOS

Son muchas las personas que de una u otra forma han contribuido a la realización de este proyecto y no bastaría una simple página para nombrarlas a todas, ni para demostrar a las nombradas toda nuestra gratitud. En el largo camino desde la idea hasta el resultado, desde la siembra a la cosecha, los amigos, compañeros y familiares son quienes nos acompañan en la lucha, en las buenas y en las malas, en las duras y en las maduras.

Los autores agradecen a la Dra. Olga Mercedes Álvarez, nuestra codirectora, por todo el apoyo brindado en la realización de este proyecto y al profesor Alfonso Mendoza Castellanos, nuestro director, quien nos guió durante su realización. También agradecemos a los residentes del departamento de Patología quienes tomaron y describieron las fotografías exhibidas en SIMUPAT, los doctores Pedro Vicente Duarte y Mariela Liliana Mejía; así como a todos los directivos y personal del departamento, encabezados por el doctor Jorge Humberto Echeverri Perico; y a los estudiantes de Medicina que pertenecieron al comité de sistemas de información, en especial a Adriana Marcela Barajas Carreño y Andrea Milena González Gómez.

Finalmente agradecemos al Creador, a nuestras familias y a todos los colombianos que con sus aportes, hacen posible la existencia de la universidad pública y la educación superior de calidad como la impartida en nuestra querida UIS.

Omar Saúl Duarte Ballén y Álvaro Martínez Duarte.

CONTENIDO

	Pág.
INTRODUCCIÓN.....	1
1 PRESENTACIÓN DEL PROYECTO	4
1.1 OBJETIVO GENERAL	4
1.2 OBJETIVOS ESPECÍFICOS	4
1.2.1 Primer objetivo	4
1.2.1.1 Descripción textual.....	4
1.2.1.2 Referencias	5
1.2.2 Segundo objetivo	5
1.2.2.1 Descripción textual.....	5
1.2.2.2 Referencias	6
1.2.3 Tercer objetivo	6
1.2.3.1 Descripción textual.....	6
1.2.3.2 Referencias	8
1.2.4 Cuarto objetivo	8
1.2.4.1 Descripción textual.....	8
1.2.4.2 Referencias	9
1.2.5 Quinto objetivo	9
1.2.5.1 Descripción textual.....	9
1.2.5.2 Referencias	9
1.2.6 Sexto objetivo.....	9
1.2.6.1 Descripción textual.....	9
1.2.6.2 Referencias	10

1.3 IMPACTO.....	10
1.4 VIABILIDAD.....	11
2 MARCO TEÓRICO RELACIONADO CON PATOLOGÍA.....	12
2.1 CONCEPTOS BÁSICOS DE PATOLOGÍA	12
2.1.1 Enfermedad.....	13
2.1.2 Lesión.....	13
2.1.3 Patología.....	13
2.1.4 Anatomía patológica	14
2.1.5 Fisiopatología.....	14
2.1.6 Etiología y patogenia	15
2.1.7 Génesis causal.....	15
2.1.8 Génesis formal	15
2.1.9 Signos y síntomas.....	16
2.1.10 Diagnóstico	16
2.1.11 Tipos de diagnóstico	16
2.1.12 Historia clínica.....	17
2.1.13 Técnicas diagnósticas en histopatología	17
2.1.13.1 Biopsia	18
2.1.13.2 Autopsia	18
2.2 MUSEO DE PIEZAS ANATOMOPATOLÓGICAS.....	19
2.2.1 Definición de museo	19
2.2.2 Especímenes anatomopatológicos	19
2.2.3 Descripción macroscópica de una pieza	20
2.2.4 Clasificación de enfermedades.....	20
2.2.5 Cie-10.....	20
2.2.6 Código CIE-10.....	21
2.2.7 Tecnología del museo de patología.....	21
2.3 OTROS MUSEOS Y GALERÍAS DIGITALES PUBLICADAS EN INTERNET	22

3 EL PROBLEMA	24
3.1 ESCENARIO DEL PROBLEMA	24
3.2 PROPUESTA DE INNOVACIÓN PEDAGÓGICA PARA LA ASIGNATURA DE PATOLOGÍA	26
3.2.1 Captación de los estudiantes y metodología de la actividad.....	29
3.3 DEFINICIÓN DEL PROBLEMA	31
4 MARCO TECNOLÓGICO	36
4.1 LOS SISTEMAS DE INFORMACIÓN	36
4.1.1 Los sistemas de información basados en Web.....	38
4.2 INTERNET	38
4.2.1 WWW.....	39
4.2.1.1 El localizador uniforme de recursos (URL).....	39
4.2.1.2 El lenguaje HTML.....	39
4.2.1.3 Los navegadores Web.....	40
4.2.1.4 El HTTP.....	40
4.2.1.5 El servicio de transferencia de archivos o FTP.....	41
4.2.2 Los servicios de correo electrónico.....	41
4.3 EL MODELO OSI Y LAS REDES DE COMPUTADORES	42
4.3.1 Capa de aplicación.....	43
4.3.2 Capa de presentación.....	43
4.3.3 Capa de sesión.....	44
4.3.4 Capa de transporte.....	44
4.3.5 Capa de red.....	44
4.3.6 Capa de enlace.....	45
4.3.7 Capa física.....	45
4.4 EL PROTOCOLO TCP/IP	46
4.4.1 Capa física.....	47
4.4.1.1 Conexión guiada.....	47
4.4.1.2 Conexión no guiada.....	48

4.4.1.3	Protocolos de última milla	48
4.4.2	Capa de enlace	48
4.4.3	Capa ip (interred)	49
4.4.4	La capa de transporte TCP/UDP	50
4.4.4.1	Tcp	50
4.4.4.2	Udp.....	50
4.4.5	La capa de aplicación	50
4.5	LA ARQUITECTURA DE UN SISTEMA	51
4.5.1	Arquitectura cliente-servidor	52
4.5.1.1	Arquitectura basada en tres capas	52
4.5.2	Arquitectura P2P (par a par)	54
4.6	EL SOFTWARE	54
4.6.1.1	Las licencias de software.....	55
4.6.1.2	Licencias de código cerrado	55
4.6.1.3	Licencias de código abierto	56
4.6.1.3.1	<i>El software libre y la licencia pública GNU</i>	57
4.6.1.3.2	<i>La licencia pública LGPL</i>	58
4.6.1.3.3	<i>La licencia BSD</i>	58
4.6.1.3.4	<i>La licencia MPL</i>	59
4.6.2	El sistema operativo.....	59
4.6.3	El servidor Web.....	59
4.6.4	La base de datos.....	59
4.6.5	Los lenguajes del lado del servidor.....	63
4.6.6	La API	64
4.6.7	Los framework.....	64
4.7	CÓDIGO DE BARRAS	65
4.7.1	Simbologías	65
4.7.2	Código 93.....	65

4.7.3	Cálculo de los dígitos de chequeo	65
4.7.4	Codificación.....	66
4.7.5	Estructura del código 93	68
5	DESCRIPCIÓN DEL SISTEMA	70
5.1	LA DISTRIBUCIÓN DE SIMUPAT	70
5.2	MARCO LEGAL.....	71
5.2.1	Licencia de uso de SIMUPAT	71
5.2.2	Licencia de uso del framework PRADO	72
5.2.2.1	Licencia BSD de PRADO original, en inglés	72
5.2.2.2	Licencia BSD de PRADO en español.....	73
5.2.3	Licencia de uso de ADODB	74
5.2.3.1	Licencia BSD/LGPL de ADODB original en inglés	75
5.2.3.2	Licencia BSD/LGPL de ADODB en español.....	75
5.2.4	Licencia de uso de PHP	75
5.2.5	Nota sobre el uso de nombres y abreviaturas registradas, en este libro	76
5.3	TECNOLOGÍA UTILIZADA EN EL PROCESO DE DESARROLLO DE SIMUPAT.....	76
5.3.1	¿Por qué usar la Web?	76
5.3.2	Herramientas de la fase de desarrollo	77
5.3.2.1	Php.....	77
5.3.2.2	MySQL	78
5.3.2.3	Adodb.....	79
5.3.2.4	Prado.....	79
5.3.3	Herramientas para la fase de prueba	80
5.3.3.1	El servidor Apache y Postfix	80
5.3.3.2	Lector de código de barras Metrologic	80
5.4	DESCRIPCIÓN DEL SISTEMA POR MÓDULOS.....	81
5.4.1	Módulo de administración de piezas.....	82
5.4.1.1	Descripción	82

5.4.1.2	Requerimientos asociados.....	82
5.4.1.3	Funciones.....	82
5.4.1.3.1	<i>Administrar la información anatomopatológica de los especímenes del museo de patología</i>	82
5.4.1.3.1.1	<i>Crear, modificar y eliminar piezas de la colección</i>	82
5.4.1.3.1.2	<i>Crear, modificar, eliminar y administrar la información sobre los diagnósticos presentes en la colección de especímenes del museo</i>	84
5.4.1.3.1.3	<i>Crear, modificar, eliminar y administrar las palabras clave que sirven como descriptores para las piezas presentes en SIMUPAT</i>	86
5.4.2	Módulo de consulta y visualización de material.....	86
5.4.2.1	Descripción	86
5.4.2.2	Requerimientos asociados.....	87
5.4.2.3	Funciones.....	87
5.4.2.3.1	<i>Visualización de la información anatomopatológica</i>	87
5.4.2.3.2	<i>Protección del material fotográfico</i>	89
5.4.2.3.3	<i>Consulta de material mediante búsqueda temática ramificada</i>	89
5.4.2.3.4	<i>Consulta de material mediante palabras clave</i>	89
5.4.3	Módulo de mantenimiento de inventario.....	92
5.4.3.1	Descripción	92
5.4.3.2	Requerimientos asociados.....	93
5.4.3.3	Funciones.....	93
5.4.3.3.1	<i>Permitir la creación, desecho y reporte del cambio de estado de los bloques consignados bajo el inventario del museo</i>	93
5.4.3.3.2	<i>Generación de reportes de inventario</i>	94
5.4.4	Módulo de administración de prácticas	96
5.4.4.1	Descripción	96
5.4.4.2	Requerimientos asociados.....	97
5.4.4.3	Funciones.....	97

5.4.4.3.1	<i>Administración y despacho de prácticas</i>	97
5.4.5	Módulo de administración de usuarios, envío de sugerencias y ayuda	99
5.4.5.1	Descripción	99
5.4.5.2	Requerimientos asociados	100
5.4.5.3	Funciones	100
5.4.5.3.1	<i>Administración de usuarios</i>	100
5.4.5.3.2	<i>Envío de sugerencias y el sistema de ayuda</i>	102
5.4.6	El módulo de administración de software	103
5.4.6.1	Descripción	103
5.4.6.2	Requerimientos asociados	104
5.4.6.3	Funciones	104
5.4.6.3.1	<i>Realización de copias de seguridad</i>	104
5.4.6.3.2	<i>Instalación del sistema</i>	104
5.4.7	Módulo de seguridad	105
5.5	REQUERIMIENTOS DE INSTALACIÓN	105
5.5.1	Requerimientos de software	105
5.5.2	Requerimientos de hardware	106
6	METODOLOGÍA	107
6.1	PROCESO UNIFICADO DE RATIONAL	107
6.1.1	Fases del RUP	108
6.1.2	Casos de uso	112
6.1.3	Diagrama de casos de uso	112
6.2	EL PLAN DE TRABAJO	114
6.3	CONCEPTO INICIAL DE LA ARQUITECTURA DE LA APLICACIÓN	116
6.4	ITERACIONES DE LA FASE DE CONSTRUCCIÓN	118
6.5	CRONOGRAMA DE ACTIVIDADES PARA UNA ITERACIÓN	118
7	ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE PARA SIMUPAT	125
7.1	INTRODUCCIÓN	125

7.1.1	Propósito de la ERS.....	125
7.1.2	Participantes	125
7.1.3	Definición del problema	125
7.1.4	Alcance del sistema	126
7.2	DESCRIPCIÓN GLOBAL DEL PRODUCTO A DESARROLLAR.....	127
7.2.1	Perspectiva del producto	127
7.2.2	Funciones del producto.....	127
7.2.3	Características generales del usuario.....	128
7.2.4	Suposiciones y dependencias	128
7.3	OBJETIVOS DEL SISTEMA, SEGÚN ESTA ESPECIFICACIÓN DE REQUERIMIENTOS ...	129
7.4	REQUERIMIENTOS DE ALMACENAMIENTO DE INFORMACIÓN.....	131
7.5	REQUERIMIENTOS FUNCIONALES	136
7.5.1	Diagramas de casos de uso	136
7.5.2	Definición de actores	140
7.5.3	Facultades de usuarios.....	141
7.5.3.1	Tipos de facultades.....	141
7.5.4	Tablas asociadas a los casos de uso	144
7.6	REQUERIMIENTOS NO FUNCIONALES.....	203
7.6.1	Requisitos de la Interfaz de usuario.....	203
7.6.2	Requisitos de la interfaz de hardware	205
7.6.2.1	Puesto de trabajo del usuario	206
7.6.3	Requisitos de la interfaz de software	206
7.6.4	Requisitos de la interfaz de comunicaciones.....	207
7.6.5	Requisitos de rendimiento	207
7.6.6	Requisitos de diseño y desarrollo.....	207
8	ESPECIFICACIONES DE DISEÑO Y ARQUITECTURA DEL SISTEMA	208
8.1	VISTA DE DESPLIEGUE	209
8.2	VISTA DE PROCESOS	212

8.3 VISTA DE IMPLEMENTACIÓN.....	221
8.4 VISTA DE DISEÑO.....	226
8.4.1 Capa de presentación.....	230
8.4.2 Capa de interfaz con presentación.....	230
8.4.3 El kernel.....	234
8.4.3.1 Capa de reglas del negocio.....	234
8.4.3.2 Capa de interfaz con la capa de datos.....	235
8.4.3.3 Las excepciones.....	239
8.4.3.4 Las utilidades.....	243
8.4.4 La capa de datos.....	243
8.4.5 La capa de configuración.....	246
8.4.6 La seguridad.....	247
8.4.6.1 El proceso de autenticación y autorización en SIMUPAT.....	249
8.4.6.2 Las copias de seguridad.....	253
8.4.6.3 SIMUPAT y la protección contra la inyección del código malicioso y las técnicas de “ingeniería social”.....	256
8.5 CONSIDERACIONES FINALES.....	259
9 DOCUMENTACIÓN Y MANUALES.....	260
9.1 EL PROCESO DE INSTALACIÓN DE SIMUPAT.....	260
9.1.1 Proceso de instalación de forma manual.....	260
9.1.2 Proceso de instalación usando el programa de instalación.....	265
9.1.3 Gestión de configuraciones.....	267
9.2 MANUAL DE USUARIO.....	267
9.3 MANUAL DEL PROGRAMADOR.....	268
10 CONCLUSIONES.....	275
11 RECOMENDACIONES.....	277
BIBLIOGRAFÍA.....	284
ANEXOS.....	285

LISTA DE TABLAS

	Pág.
Tabla 4.1. Cálculo de los pesos para los caracteres de la cadena "TEST93".....	66
Tabla 4.2. Cálculo de los pesos para los caracteres de la cadena "TEST93+".....	66
Tabla 4.3. Tabla de codificación para la simbología código 93.	67
Tabla 6.1. Tabla de productos e hitos generados en cada una de las fases del RUP.	115
Tabla 7.1. Resumen de los permisos de usuario para los requerimientos del RF-01 al RF-20.	142
Tabla 7.2. Resumen de los permisos de usuario para los requerimientos del RF-21 al RF-40.	143
Tabla 7.3. Resumen de los permisos de usuario para los requerimientos del RF-41 al RF-43.	144
Tabla 9.1. Lista de controladores para ADODB y su nombre en el objeto conexión.....	263

LISTA DE FIGURAS

	Pág.
Figura 2.1. Espécimen anatomopatológico que presenta como diagnóstico principal “Atrofia cerebral” y como diagnóstico secundario “Hidrocefalia”.....	20
Figura 2.2. Museo de patología perteneciente a la Escuela de Medicina de la Universidad de Queensland en Brisbane Australia.....	21
Figura 3.1. Estanterías del museo de patología antes del proceso de organización. Se evidencia el problema de desconocimiento de la ubicación.	26
Figura 3.2. Planos de la reforma proyectada para el laboratorio de patología. (Arriba) Vista superior. (Abajo) Vista lateral.	28
Figura 3.3. (a) Fotografía de un recipiente que contiene una pieza en estado de putrefacción. (b) Fotografía donde se puede observar que la pieza se adhirió al fondo del recipiente.	33
Figura 3.4. Espécimen anatomopatológico incluido en un bloque de resina perteneciente a la colección de la Universidad Autónoma de Bucaramanga.	35
Figura 4.1. Ejemplo típico de un código de barras utilizando la simbología código 93.	65
Figura 4.2. Codificación grafica del número 1101.....	68
Figura 4.2. Ejemplo de la estructura de un código de barras que utiliza la simbología código 93... ..	69
Figura 5.1. Captura de pantalla de la interfaz gráfica de usuario de SIMUPAT correspondiente a la página de bienvenida.	81
Figura 5.2. Fragmento de la captura de pantalla del control administrador de fotos de SIMUPAT.	83

Figura 5.3. Fragmento de la captura de pantalla del administrador de diagnósticos de la pieza en SIMUPAT. Se observan el órgano y los diagnósticos de una pieza en particular para su edición. .	83
Figura 5.4. Fragmento de la captura de pantalla del paso numero 2 del proceso de modificación de una pieza, en donde se muestra al usuario la descripción macroscópica actual para que haga los cambios pertinentes.	84
Figura 5.5. Fragmento de la captura de pantalla de la página de administración de diagnósticos..	85
Figura 5.6. Captura de pantalla luego de activar la función de modificación de diagnósticos.	85
Figura 5.7. Captura de pantalla de la página de administración de palabras clave. A diferencia de la figura 5.5, esta captura no se ha fragmentado para que se observe como encaja la interfaz gráfica de usuario con la presentación general de SIMUPAT.	86
Figura 5.8. Fragmento de la captura de pantalla de la interfaz gráfica de usuario de visualización de detalles de la pieza. Se pueden observar los diagnósticos asociados, el diagnóstico principal, el órgano, las miniaturas de las otras fotografías de la pieza (si el usuario hace clic, se amplían a tamaño real) y la fotografía principal con un comentario añadido.	88
Figura 5.9. Captura de pantalla del primer paso de la búsqueda ramificada por órganos, en donde el usuario selecciona el órgano de su interés.	90
Figura 5.10. Captura de pantalla del segundo paso en la búsqueda ramificada (por diagnóstico)..	90
Figura 5.11. Fragmento de captura de pantalla de los resultados de la búsqueda ramificada. Si el usuario da clic sobre alguno de los resultados, una pantalla como la mostrada en la figura 5.8 con la información de la pieza que desea observar.....	91
Figura 5.12. Fragmento de captura de pantalla que muestra los resultados de búsqueda directa..	91
Figura 5.13. Fragmento de la captura de pantalla de la página de administración de bloques de SIMUPAT para la pieza cuyo código es A01.2+01.	93

Figura 5.14. Captura de pantalla del programa de cambio de estado del bloque, en donde el usuario puede reportar un bloque como averiado, en mantenimiento, etc. La imagen muestra este caso para el bloque rotulado A01.2+01W02.....	94
Figura 5.15. Rótulo del bloque A01.2+01W04 en código de barras codificación 93, generado por SIMUPAT.....	94
Figura 5.16. Fragmento de la captura de pantalla de la interfaz gráfica de generación de reportes de inventario en SIMUPAT.....	95
Figura 5.17. Fragmento de captura de pantalla del reporte de inventario generado por SIMUPAT.	96
Figura 5.18. Fragmento de captura de pantalla de la información brindada al estudiante sobre la práctica “Lesión celular” consistente en la lista de piezas a estudiar, la semana de realización y el tipo de práctica. Si se da clic sobre alguna de estas piezas se verán los detalles de la misma, como en la figura 5.8.....	97
Figura 5.19. Fragmento de captura de pantalla de los detalles de un cerebro en donde se puede apreciar el botón “agregar a una práctica”.....	98
Figura 5.20. Fragmento de captura de pantalla de la página de modificación de prácticas.	99
Figura 5.21. Fragmento de la captura de pantalla de la página de “despachar práctica” en SIMUPAT.....	100
Figura 5.22. Captura de pantalla de la página de inicio de sesión.	101
Figura 5.23. Captura de pantalla de la página de administración de usuarios.....	101
Figura 5.24. Captura de pantalla del formulario de envío de inquietudes y sugerencias.....	102
Figura 5.25. Captura de pantalla de la película Flash del tutorial de administración de usuarios..	103
Figura 5.26. Captura de la película Flash del tutorial de administración de usuarios, en donde se muestra un globo de ayuda.....	103
Figura 5.27. Fragmento de captura de pantalla de la página de creación de copia de seguridad de la base de datos. El botón “Comenzar volcado” inicia este proceso.	104

Figura 5.28. Fragmento de captura de pantalla del proceso de configuración de la base de datos.	105
Figura 6.1. El ciclo de vida del desarrollo de software.....	109
Figura 6. 2. Modelado de la arquitectura de un sistema.....	115
Figura 6.3. Arquitectura global de la aplicación.	117
Figura 7.1. Diagrama de Subsistemas que muestra la interacción entre los diferentes actores de SIMUPAT y los subsistemas (módulos) que lo componen.....	136
Figura 7.2. Diagrama de casos de uso del módulo (subsistema) de administración de piezas.....	137
Figura 7.3. Diagrama de casos de uso para el módulo de consulta.....	138
Figura 7.4. Diagrama de casos de uso para el módulo de mantenimiento de inventario.....	138
Figura 7.5. Diagrama de casos de uso del módulo (subsistema) de administración de prácticas.	139
Figura 7.6. Diagrama de casos de uso del módulo (subsistema) de administración de usuarios..	139
Figura 7.7. Diagrama de casos de uso para el subsistema de administración de software.....	140
Figura 8.2. Diagrama de secuencia que muestra los eventos en PRADO.....	214
Figura 8.3. Diagrama que actividades que muestra el proceso de respuesta de petición del cliente, desde que el usuario interactúa con el formulario hasta que se utiliza la base de datos.....	216
Figura 8.4. Diagrama de actividades que muestra el proceso de salida de resultados hacia el cliente, efectuado por SIMUPAT, desde la consulta en la base de datos hasta la presentación final.	218
Figura 8.5. Diagrama de actividad del ciclo de vida de una aplicación en PRADO.	220
Figura 8.6. Organización de las carpetas de SIMUPAT dentro de la carpeta de documentos Web en el servidor.....	226
Figura 8.7. Diagrama de paquetes que refleja la arquitectura de SIMUPAT.....	229

Figura 8.8. Ejemplo de una plantilla de página para usar con PRADO, cuyo nombre de archivo es quiz.page.....	231
Figura 8.9. Ejemplo del código PHP de SIMUPAT que responde a los eventos de PRADO para la página de ejemplo quiz.page y cuyo nombre de archivo es quiz.php	232
Figura 8.10. Fragmento de la captura de pantalla para la página generada por el código de ejemplo quiz.php y la plantilla quiz.page.	233
Figura 8.11. Fragmento de la página de salida una vez que el usuario ha escrito el nombre “SIMUPAT” y pulsado el botón “Enviar datos”.	233
Figura 8.12. Diagrama de clases correspondiente a la Capa de Reglas del Negocio en SIMUPAT. Los métodos necesarios para establecer y obtener los atributos han sido omitidos por comodidad... ..	235
Figura 8.13. Diagrama de clases con acceso a datos para la capa de Interfaz con la capa de datos en SIMUPAT.....	238
Figura 8.14. Diagrama de las clases de colección para la capa de Interfaz con la capa de datos en SIMUPAT.....	239
Figura 8.15. Fragmento de una captura de pantalla del módulo de instalación, en donde se ha mostrado un error al usuario.	240
Figura 8.16. Diagrama de clases para el paquete de excepciones en SIMUPAT.....	241
Figura 8.17. Implementación del método cargar, perteneciente a la clase Corgano_db, en donde se muestra el punto en donde una excepción es disparada.	242
Figura 8.18. Ejemplo de una pantalla con trazado inverso generada por una excepción de la clase Cusuario.	243
Figura 8.19. Diagrama entidad-relación para la base de datos de SIMUPAT.....	244
Figura 8.20. Fragmento de la captura de pantalla de la sección “Configuración de la base de datos” parte del proceso de instalación de SIMUPAT.	245

Figura 8.21. Fragmento del archivo de configuración config.inc en donde pueden verse algunas de las definiciones de parámetros de SIMUPAT.	247
Figura 8.22. Estructura del archivo de autorización para las páginas del módulo de administración de piezas en SIMUPAT.	251
Figura 8.23. Estructura de ejemplo para el archivo de autenticación de SIMUPAT. No se puede ver la clave del usuario, sólo su resumen HASH.	253
Figura 8.24. Ejemplo de volcado de instrucciones SQL en un archivo.....	254
Figura 8.25. Fragmento del archivo de volcado generado por SIMUPAT del contenido de la tabla órganos.....	255
Figura 9.1. Captura de pantalla que muestra la organización de carpetas que tiene una copia instalada de SIMUPAT y que ilustra como copiar SIMUPAT junto con los paquetes ADODB y PRADO.....	261
Figura 9.2. Figura que muestra la página de instalación de SIMUPAT.....	261
Figura 9.3. Ejemplo de un error “Página no encontrada”.....	262
Figura 9.4. Captura de pantalla del código del archivo conexion.inc.....	263
Figura 9.5. Captura de pantalla que muestra el contenido inicial del archivo lista_usuarios.xml...	264
Figura 9.6. Fragmento de la captura de pantalla del inicio de sesión.....	265
Figura 9.7. Formulario de configuración de la base de datos en el programa de instalación.	265
Figura 9.8. Pantalla que informa que las tablas han sido creadas con éxito.....	266
Figura 9.9. Formulario de creación del primer administrador del sistema.....	266
Figura 9.10. Captura de pantalla de una animación del Manual de Usuario.....	268
Figura 9.11. Captura de pantalla de la página principal del Manual del Programador.....	269
Figura 9.12. Árbol de navegación del Manual del Programador, expandido para mostrar las clases de la capa “Reglas del negocio”.....	269

Figura 9.13. Árbol de clases que muestra las clases que heredan de la clase CObjeto.....	270
Figura 9.14. Paquetes disponibles en la documentación contenida en el Manual del Programador.	270
Figura 9.15. Descripción de la clase Cpieza en el Manual del Programador.	271
Figura 9.16. Relación de constantes en la clase listador_mini_piezas_administracion.	271
Figura 9.16 Descripción de las variables de la clase de Cpieza.....	272
Figura 9.17. Relación de las variables heredadas de Cmini_pieza por la clase Cpieza en el Manual del Programador.....	272
Figura 9.18. Relación de los métodos de Corgano en el Manual del Programador.	273
Figura 9.19. Resumen de los métodos heredados por Cpieza de Cmini_pieza.....	273
Figura 9.20. Detalle de la documentación del método establecer_ruta() en la clase Cfoto.	274
Figura 9.21. Resumen de los archivos incluidos (y clases) incluidas en el archivo Corgano.inc...	274
Figura B.1. Prototipo en papel de la interfaz gráfica de usuario para el caso de uso crear pieza.	288
Figura B.2. Prototipo en papel de la interfaz gráfica de usuario para el caso de uso buscar piezas.	289

LISTA DE ANEXOS

	Pág.
A. ACTA DE ENTREGA DEL SISTEMA	286
B. PROTOTIPOS EN PAPEL DE LA INTERFAZ GRÁFICA DE USUARIO	288

RESUMEN

TITULO:

SISTEMA DE INFORMACIÓN BASADO EN WEB PARA APOYAR LA ORGANIZACIÓN Y EXHIBICIÓN DE LOS ESPECÍMENES ANATOMOPATOLÓGICOS PERTENECIENTES AL MUSEO DE PATOLOGÍA UIS.*

AUTORES:

DUARTE BALLÉN, Omar Saúl.
MARTÍNEZ DUARTE, Álvaro. **

PALABRAS CLAVE:

Museo, patología, software, código de barras, UML, MySQL, PHP, Web.

DESCRIPCIÓN:

El museo de piezas macroscópicas del Departamento de Patología, es una colección de especímenes anatomopatológicos, provenientes de cirugías y autopsias, cuya función es suministrar material didáctico para la realización de actividades académicas de práctica en las modalidades de pregrado y postgrado en la Facultad de Medicina de la Universidad Industrial de Santander. Las actividades administrativas y de mantenimiento de este museo demandan gran cantidad de tiempo y recursos por parte del personal administrativo y técnico del departamento. Además es voluntad de las directivas, mejorar continuamente las estrategias pedagógicas, de tal forma que se favorezca el aprendizaje significativo y se logre una disminución de la mortalidad académica. Para dar solución a esta situación se realizó este proyecto cuyo resultado es el Sistema de Información para el Museo de Patología (SIMUPAT 1.0).

SIMUPAT es un sistema de información basado en Web, desarrollado siguiendo la metodología del Proceso Unificado de Rational (RUP), la cual soporta las técnicas de modelado orientado a objetos y utiliza UML como lenguaje para la elaboración de especificaciones de requerimientos y diseño.

El sistema desarrollado ha sido concebido como una herramienta para la publicación en Internet de la información anatomopatológica de los especímenes pertenecientes al museo y para la organización de dichos especímenes dentro del inventario físico. La información anatomopatológica está contenida, en forma textual y fotográfica, dentro de una base de datos que facilita su consulta por parte del público interesado. La organización del inventario físico es apoyada por medidas como la adopción de código de barras para identificar los especímenes y la generación de reportes para la administración y mantenimiento.

* Trabajo de grado, modalidad investigación.

** Facultad de Ingenierías Físico Mecánicas.
Escuela de Ingeniería de Sistemas e Informática.

Director de proyecto: Bachelor of Science DEA Alfonso Mendoza Castellanos.

ABSTRACT

TITLE:

WEB BASED INFORMATION SYSTEM FOR PUBLISHING AND MANAGING THE ANATOMOPATHOLOGIC SPECIMENS STORED IN THE PATHOLOGY MUSEUM OF THE UIS*.

AUTHORS:

DUARTE BALLÉN, Omar Saúl.
MARTÍNEZ DUARTE, Álvaro. **

KEYWORDS:

Museum, pathology, software, barcode, UML, MySQL, PHP, Web.

DESCRIPTION:

The museum of the Department of Pathology is an anatomopathologic collection, whose specimens are the result of autopsies and surgeries, and its function is to supply didactic material for the accomplishment of different academic activities, such as practices, for medical students and pathology residents, inside the Universidad Industrial de Santander. The management and maintenance tasks of this museum take a lot of time and resources from the managers and technicians of the Department. In addition, one of the management policies is to improve continuously the current pedagogic strategies, promoting the significant learning and improving the academic results. For solving this situation this project was made, resulting in the Information System for the Pathology Museum (SIMUPAT 1.0).

SIMUPAT is a Web based information system. It was developed using the Rational Unified Process (RUP), which supports the object oriented modeling techniques and uses UML as a language for the construction of requirements specifications and software design.

The system constructed has been designed as a tool for publishing, in Internet, the anatomopathologic information related to the specimens of the museum and for organizing those specimens inside the inventory facilities. The anatomopathologic information of the specimens is stored, as pictures and text, in a database which makes easier to browse and search that information by the interested community. The museum management is supported with policies and features, such as the use of barcodes for identifying the specimens, and the report generation for helping the management and maintenance tasks.

* This is a research project for getting the college degree.

** Facultad de Ingenierías Físico Mecánicas.

Escuela de Ingeniería de Sistemas e Informática.

Project Director: Bachelor of Science DEA Alfonso Mendoza Castellanos.

INTRODUCCIÓN

El presente documento es la recopilación textual de varios meses de trabajo invertidos en el desarrollo del Sistema de Información basado en Web para apoyar la organización y exhibición de los especímenes anatomopatológicos pertenecientes al Museo de Patología UIS (SIMUPAT) el cual ha sido construido por los estudiantes Omar Saúl Duarte Ballén y Álvaro Martínez como trabajo de grado para optar por el título de Ingeniero de Sistemas otorgado por la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander. El objetivo de este documento es presentar un informe textual del proceso de desarrollo de SIMUPAT y adjuntar la documentación necesaria para referencia futura por parte de los lectores interesados.

En este libro los autores del proyecto serán referidos como “los autores” y el sistema construido será referido como SIMUPAT. El presente libro comienza con un capítulo denominado “Presentación del Proyecto” en el cual se describen el objetivo general, los diferentes objetivos específicos del mismo, de forma que el lector tenga una referencia global del alcance y objetivos alcanzados en este proyecto.

El segundo capítulo, denominado “Marco teórico relacionado con Patología”, es una síntesis de los conceptos básicos referentes a la Patología como disciplina en el ámbito de la Medicina y de los términos usados frecuentemente en la descripción de los museos de especímenes anatomopatológicos, y por ende a lo largo de este libro.

El tercer capítulo tiene como objetivo contextualizar al lector dentro de la situación, denominada “El Problema”, que dio origen a la planeación y realización de este proyecto.

El cuarto capítulo es una síntesis de los conceptos y términos usados para describir las diferentes tecnologías informáticas que se relacionan, de una u otra forma, con los recursos empleados por

los autores para el desarrollo de este proyecto, por ejemplo la forma de construir y leer códigos de barras, y que por consiguiente son utilizados de forma exhaustiva en este libro.

En cuanto al quinto capítulo, denominado “Descripción del sistema”, es una sección que describe los diferentes módulos que componen SIMUPAT, junto con las funciones desempeñadas por dichos módulos. Esta sección cuenta con un número considerable de capturas de pantalla tomadas de la Interfaz Gráfica de Usuario de SIMUPAT, de forma que el lector que no tenga a su disposición el sistema para interactuar directamente con él, pueda al menos formarse una idea de su aspecto y sus características. Adicionalmente cada módulo referencia los requerimientos funcionales (consignados en el capítulo siete, la Especificación de Requerimientos de Software) que corresponden a su fase de análisis, de modo que pueda observar la relación entre tales requerimientos y su implementación final. Este quinto capítulo está relacionado directamente con el tercer objetivo específico de este proyecto.

El sexto capítulo versa sobre la metodología utilizada por los autores en el proceso de desarrollo de SIMUPAT y muestra la manera en que la realización de este proyecto fue planeada y ejecutada. Esta sección tiene como objetivo que el lector que disponga del Plan de Proyecto de SIMUPAT tenga una visión de cómo se aplicó dicho plan a la realización de este trabajo de grado.

El séptimo capítulo da respuesta al primer objetivo específico de este proyecto y corresponde a la Especificación de Requerimientos de Software (ERS) que fue aprobada por el Comité de Sistemas de Información, conformado por los autores y personal del Departamento de Patología de la Universidad Industrial de Santander. Corresponde este capítulo al resultado de la fase de análisis y ha sido construido usando tablas de requerimientos funcionales, lista de requerimientos no funcionales, suposiciones, dependencias y diagramas de casos de uso (Usando UML 2.0)

Las especificaciones de diseño y arquitectura del sistema, nombre dado al capítulo octavo, dan respuesta al segundo objetivo específico de este proyecto y contienen los diferentes diagramas UML que muestran la arquitectura del sistema desde diferentes puntos de vista, en concreto: el modelo de datos (diagrama entidad-relación), los diagramas de clases y componentes, el diagrama de despliegue y la vista de implementación. Este capítulo no se limita a mostrar pautas y

especificaciones de diseño, sino que muestra como estas especificaciones han sido implementadas en SIMUPAT. Se discuten aspectos importantes del diseño y construcción del sistema como la seguridad (lo que da respuesta del sexto objetivo específico), la forma en que SIMUPAT tiene organizados sus archivos, la creación de copias de seguridad del contenido de la base de datos, la gestión de parámetros y configuraciones, entre otros.

El capítulo noveno, denominado “Manuales de usuario”, muestra al lector en donde encontrar y como utilizar la ayuda en el uso de SIMUPAT, la localización de las animaciones que ilustran el uso de los diferentes módulos, la localización de la documentación de cada clase y cada método que conforma SIMUPAT, (Esta última documentación en formato de página Web para su mejor comprensión y lectura) junto con las instrucciones de instalación y configuración. Este capítulo da respuesta a los objetivos específicos cuatro y cinco.

Los dos últimos capítulos, las conclusiones y recomendaciones, son para los autores los más importantes. Estos capítulos muestran lo que los autores han aprendido en la ejecución de este proyecto, los errores que cometieron, la solución a los problemas que se iban presentando y otros temas relativos a la realización de este trabajo de grado.

Para concluir esta introducción, este libro fue estructurado en tres partes principales que dan respuesta a preguntas importantes que los autores, en calidad de lectores, se harían si leyesen un informe de trabajo de grado: ¿Cuáles fueron los objetivos del proyecto? ¿Qué problema originó la idea de realizar el proyecto? ¿Cómo se cumplieron los objetivos? ¿Cómo se desarrolló el proyecto y que se aprendió? Los autores esperan que al finalizar la lectura de este libro el lector conozca la respuesta a estas preguntas.

1 PRESENTACIÓN DEL PROYECTO

Este capítulo, como se dijo en la Introducción, tiene como propósito que el lector conozca el objetivo general y los objetivos específicos de este trabajo, referenciados de forma cruzada con las diferentes secciones de este libro que muestran cómo se cumplieron estos objetivos y la documentación asociada a ellos.

1.1 OBJETIVO GENERAL

Contribuir a la organización del museo de patología por medio de la construcción de una herramienta software basada en Web que permita administrar la colección y sirva como material didáctico de consulta.

1.2 OBJETIVOS ESPECÍFICOS

Antes de comenzar a enunciar los objetivos específicos de este proyecto, los autores pensando en la comodidad del lector, han colocado bajo cada objetivo una sección llamada “referencias”, la cual contiene las referencias cruzadas a las secciones de este libro y ubicación de los productos entregables dentro del software SIMUPAT, en donde se muestra el cumplimiento de dicho objetivo, es decir, el resultado concreto del mismo. La descripción textual de los objetivos se ha tomado tal y como está consignada en el Plan de Proyecto del presente trabajo de grado.

1.2.1 Primer objetivo

1.2.1.1 Descripción textual

Realizar un análisis con el fin de construir una Especificación de Requerimientos de Software que recopile las necesidades de información del personal administrativo del Museo y en general de la comunidad académica beneficiaria de sus servicios. Los requerimientos que serán documentados son:

- Requerimientos funcionales del Sistema, basados en casos de uso¹.
- Requerimientos de Almacenamiento de Información.
- Requerimientos no funcionales del sistema²:
- Requerimientos de Interfaz de Usuario.
- Requerimientos de Interfaz de Hardware y Software.
- Requerimientos de comunicaciones.
- Requerimientos de Rendimiento.
- Suposiciones y Dependencias.
- Restricciones.

1.2.1.2 Referencias

- Especificación de requerimientos de software, capítulo séptimo.
- Diagramas UML de casos de uso, correspondientes a las figuras: 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7.

1.2.2 Segundo objetivo

1.2.2.1 Descripción textual

Construir, con base en la Especificación de Requerimientos de Software y en la metodología especificada para este proyecto (ver sección metodología), Especificaciones de Diseño desde los siguientes puntos de vista:

- Componentes del Sistema: Módulos, Objetos, Clases y Métodos.

¹ DURAN TORO, Amador y BERNÁRDEZ JIMÉNEZ, Beatriz. Metodología para elicitación de requisitos software. Versión 2.1. Sevilla: 2000. 67 p. Universidad de Sevilla. Facultad de Informática y Estadística. Departamento de Lenguajes y Sistemas Informáticos.

² IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830-1998, Institute of Electrical and Electronics Engineers, 1998.

- Despliegue del Sistema: Distribución del Sistema en los diferentes componentes hardware en los que se constituye.
- Interfaz Gráfica de Usuario: Alternativas y pautas para el diseño de esta interfaz.
- Modelo de Datos (Entidad-Relación).

1.2.2.2 Referencias

- Especificaciones de diseño y arquitectura del sistema, capítulo octavo.
- Diagramas UML correspondientes a las figuras 8.1, 8.3, 8.4, 8.7, 8.12, 8.13, 8.14 y 8.16
- Diagrama entidad-relación correspondiente a la figura 8.19.

1.2.3 Tercer objetivo

1.2.3.1 Descripción textual

Desarrollar con base en la Especificación de Requerimientos de Software y las Especificaciones de Diseño, un Sistema de Información de tipo modular consistente en:

- **Módulo de Administración de Piezas:** Encargado de manejar la información relacionada con cada uno de los especímenes anatomopatológicos del museo, específicamente: fotografías macroscópicas y microscópicas de la pieza, diagnósticos, historia clínica del donante, número de clasificación en el estándar CIE-10, etc. Las funciones que tendrá este módulo son:
 - Registrar el ingreso de piezas nuevas a la colección del museo.
 - Actualizar la información relevante a cada pieza agregando, eliminando o modificando la información asociada a dicha pieza.
- **Módulo de Mantenimiento de Inventario:** Provee funciones para apoyar la realización del mantenimiento de la colección del museo:
 - Impresión de rótulos provistos de código de barras para la identificación de las piezas.
 - Impresión de catálogos e informes de inventario.

- **Modulo de Consulta y Visualización de Material:** Como su nombre lo indica, este módulo es el encargado de llevar a cabo las operaciones de consulta que necesiten los usuarios de la aplicación y de mostrar los resultados de dicha consulta. Este módulo tendrá las siguientes características:
 - Búsqueda ramificada por sistemas, órganos y diagnósticos.
 - Búsqueda por palabras clave.
 - Búsqueda directa por código de clasificación CIE-10.
 - Búsqueda por número de Inventario.
 - Resultados de las búsquedas ordenados por relevancia (ranking).
 - Visualización de la colección de fotografías de la pieza mediante vistas en miniatura.
 - Protección de la colección fotográfica mediante una marca de agua digital con el logotipo de la Universidad Industrial de Santander.
- **Módulo de Registro de Solicitudes y Sugerencias:** Encargado de recopilar y registrar las sugerencias que los usuarios tengan en relación con los servicios que presta el museo, de registrar las solicitudes y sugerencias relacionadas con la adquisición de nuevas piezas, y registrar las solicitudes de revisión de los criterios diagnósticos con los que ha sido evaluada una pieza determinada.
- **Módulo de Administración de Prácticas:** Este módulo se encargará, como su nombre lo indica, de agilizar y facilitar la realización de las prácticas de clase que ofrece el Departamento de Patología haciendo uso de los servicios del Museo, para lo cual desempeñará las siguientes funciones:
 - Almacenar y publicar, para cada práctica: el nombre, la fecha, marco teórico, lista de piezas a observar, y demás información pertinente para los estudiantes y profesores.
 - Publicar los cambios en las fechas asignadas a las prácticas.

- Mostrar al técnico encargado las piezas que debe tener preparadas de antemano para la realización de cada práctica.
- **Módulo de Administración de Usuarios:** Este módulo se encarga de administrar los diferentes perfiles de usuario, y los diferentes permisos y facultades que tienen cada uno de ellos dentro del sistema. Sus funciones son:
 - Realizar la autenticación de usuario.
 - Permitir al administrador agregar o eliminar usuarios.
 - Permitir al administrador cambiar los permisos de cada usuario.
 - Vigilar que cada usuario sólo tenga acceso a las funciones del sistema que sean de su competencia.
- **Módulo de Administración y Mantenimiento del Sistema:** Proporciona funciones relativas a la administración y mantenimiento del Sistema de Información tales como:
 - Realización de copias de seguridad.
 - Instalación del Sistema.
 - Configuración del Sistema.

1.2.3.2 Referencias

- Software entregado junto con este libro como resultado de la realización de este proyecto de grado.
- Capítulo 5, Descripción del Sistema.

1.2.4 Cuarto objetivo

1.2.4.1 Descripción textual

Documentar la construcción del sistema descrito del tal forma que sirva como referencia para futuras modificaciones y actualizaciones, mediante la elaboración de un Manual del Programador.

1.2.4.2 Referencias

- Manual de programador, disponible en formato de páginas Web, ubicadas en la carpeta: *manual_del_programador* dentro de la carpeta principal que contiene el software SIMUPAT (llamada SIMUPAT en el CD entregado por los autores).
- Sección 9.3 de este libro, perteneciente al capítulo noveno “Manuales de usuario”.
- Capítulo séptimo, denominado, Especificaciones de diseño y arquitectura del sistema.

1.2.5 Quinto objetivo

1.2.5.1 Descripción textual

Construir documentación que sirva como referencia al usuario sobre el uso del Sistema, mediante un módulo de Ayuda en Línea respaldado por documentos impresos, específicamente:

- Manual de Operación.
- Guía de Administración y Mantenimiento del Sistema.

1.2.5.2 Referencias

- Películas, en formato *SWF*¹, ubicadas en el subdirectorio /ayuda, dentro de la carpeta principal que contiene el software SIMUPAT (llamada SIMUPAT en el CD entregado por los autores).
- Secciones 9.2 de este libro, pertenecientes al capítulo noveno “Manuales de Usuario”.
- Sección 5.4.5.3.2 relativa a la explicación del módulo de ayuda en el capítulo quinto.

1.2.6 Sexto objetivo

1.2.6.1 Descripción textual

Implementar mecanismos de seguridad informática para reducir el riesgo e impacto de un eventual ataque por parte de código malicioso o piratas informáticos, conservando en lo posible la integridad de los datos manejados por el Sistema.

¹ SWF hace referencia al formato *Shockwave Flash*, popular en animaciones para incluir dentro de páginas Web.

1.2.6.2 Referencias

- Software entregado junto con este libro como resultado de la realización de este proyecto de grado.
- Sección 8.4.6 perteneciente al capítulo octavo “Especificaciones de diseño y arquitectura del sistema”.

1.3 IMPACTO

Aunque el presente proyecto se encuentra aún en las primeras fases de su implantación y las siguientes afirmaciones no pueden ser probadas en este momento, los autores piensan que la posterior implantación del Sistema de Información desarrollado impactará positivamente al Departamento de Patología, a la Facultad de Salud, y a la UIS en general por las siguientes razones:

- Las existencias del museo de patología estarán disponibles para la consulta en línea por Internet, apoyada por fotografías en distintos planos de las mismas; lo que convierte al Sistema en una fuente primaria de consulta por parte de los estudiantes de medicina, contribuyendo de esta forma con la misión institucional de la Universidad Industrial de Santander, en lo referente a las premisas de formación de personas de alta calidad profesional y su propósito de generación y adecuación de conocimiento.
- Este sistema reducirá considerablemente los costos de manejo del museo, manteniendo un control real sobre el mismo, lo cual hará que el personal (tanto estudiantes como profesores y técnicos) que lo utilice pueda administrarlo de una forma más efectiva.
- Al saber exactamente de que se dispone en el museo, los profesores podrán diseñar nuevas prácticas de clase que aprovechen mejor los recursos disponibles.
- El personal del museo adoptará mejores procedimientos para la administración de los especímenes, que pueden servir como base para futuros proyectos y la creación del manual operativo, o de procedimientos, del museo.

1.4 VIABILIDAD

La realización de este proyecto fue posible debido a los siguientes aspectos:

En primera instancia, la disponibilidad del recurso humano, representado por el cuerpo docente y administrativo del departamento de Patología, el personal técnico del museo, el profesor Alfonso Mendoza Castellanos director del grupo GIB y los autores del presente proyecto. El Departamento de Patología demostró su apoyo incondicional para supervisar el desarrollo del proyecto, al punto de haber creado, junto con los autores, el comité de sistemas de información del museo que se reunió semanalmente y que permitió mantener un contacto directo y una supervisión constante por parte de los usuarios.

En cuanto al recurso tecnológico, dos computadores marca Dell de última tecnología¹, adquiridos recientemente por el departamento de patología, dedicados exclusivamente al desarrollo de este proyecto; el servidor Power Edge, propiedad del grupo GIB; y el software de libre distribución: PHP, Apache y MySQL. El uso de estas herramientas de software libre tuvo grandes ventajas, como lo es el respaldo suministrado por el enorme grupo de personas que trabajan en todo el mundo para mejorar su calidad y seguridad, y el apoyo y orientación suministrado por parte de los grupos de discusión en línea que además propician el intercambio de componentes software que pueden incluirse en nuevas aplicaciones.

En el aspecto económico se encontró un escenario favorable debido a que los costos causados por concepto de recurso humano, los cuales constituyen una gran proporción del costo total del proyecto, son cubiertos por la universidad en calidad de inversión social. Los costos correspondientes a la adquisición del lector de código de barras y a los suministros para la impresión de documentos fueron solventados con el dinero recaudado por el comité, provenientes de las ganancias que conllevó la realización de una fiesta. Además, el uso de software libre ahorró el costo de adquirir licencias para desarrollar este proyecto.

¹ A la fecha de escritura de esta sección, 17 de Octubre de 2006.

2 MARCO TEÓRICO RELACIONADO CON PATOLOGÍA

En esta sección se introducen algunos conceptos básicos de patología y museos de patología con el objetivo de familiarizar al lector con los términos médicos utilizados en este libro. Dichos conceptos serán necesarios para comprender el significado de los datos manejados por el sistema así como la lógica y arquitectura descritas en capítulos posteriores. También se mencionan y describen brevemente algunos museos y galerías digitales que se encuentran actualmente publicadas en Internet. La mayoría de los conceptos que aparecen en la sección 2.1 fueron tomados de la Wikipedia¹.

2.1 CONCEPTOS BÁSICOS DE PATOLOGÍA

Clásicamente se ha definido la Patología como la disciplina que estudia el origen, curso, diagnóstico y pronóstico de la enfermedad. Además de estos objetivos que cubren la totalidad de las especialidades médicas y quirúrgicas, la Patología actualmente es imprescindible en la investigación médica, en la detección y manejo de los tumores, y en general, en la interpretación y control de todos los trastornos orgánicos. Como especialidad médica moderna la Patología constituye el pilar científico fundamental que permite la integración de los conocimientos biológicos básicos con la práctica clínica, basada en la observación objetiva de las alteraciones morfológicas provocadas por la enfermedad a nivel de órganos, tejidos y células.

Debido a lo anterior, el estudiante de Medicina y de Bacteriología y Laboratorio Clínico cursa dentro de su plan de estudios la asignatura de Patología. Esta asignatura, se compara con el eslabón que une las materias básicas que estudian el aspecto normal del ser humano, con las materias clínicas que abordan al paciente desde la enfermedad mental y física. La asignatura de patología es el primer contacto que tiene el estudiante con la enfermedad, desde sus causas, cambios en órganos,

¹ WIKIMEDIA FOUNDATION, INC. Wikipedia, la enciclopedia libre. Disponible en: < <http://es.wikipedia.org/>>

tejidos y células, mecanismo de enfermedad, evolución de la misma y desenlace. Por todo esto se hace imprescindible aunar esfuerzos para lograr que el aprendizaje que se adquiriera sea significativo.

2.1.1 Enfermedad

Una enfermedad es cualquier trastorno anormal del cuerpo o la mente que provoca malestar y alteración de las funciones normales. Las enfermedades son entidades creadas por el hombre, en su afán de clasificar el conocimiento, en la que se agrupan una serie de componentes que las definen.

2.1.2 Lesión

Una lesión es un cambio anormal en la morfología o estructura de una parte del cuerpo producida por un daño externo o interno. Las heridas en la piel pueden considerarse lesiones producidas por un daño externo como los traumatismos. Las lesiones producen una alteración de la función o fisiología de órganos, sistemas y aparatos, trastornando la salud y produciendo enfermedad.

Podría decirse que todas las enfermedades están producidas por una lesión, pero en muchas enfermedades no es posible identificar una clara lesión morfológica, como en las enfermedades mentales. Además no todas las lesiones provocan una enfermedad, ni requieren tratamiento.

La especialidad médica encargada de identificar las lesiones de las enfermedades, generalmente mediante biopsias es la anatomía patológica.

2.1.3 Patología

La patología es el estudio de las enfermedades en su amplio sentido, es decir, como procesos o estados anormales de causas conocidas o desconocidas. La palabra deriva del griego, estudio (logos) del sufrimiento o daño (pathos).

Las pruebas que mejor demuestran la existencia de una enfermedad se basan principalmente en el examen de una lesión en todos sus niveles estructurales, la evidencia de la presencia de un microorganismo (bacteria, parásito, hongo o virus) cuando se trata de una enfermedad infecciosa o

la alteración de algún o algunos componentes del organismo (por ejemplo la glucosa en la diabetes mellitus, o la hemoglobina, en la anemia).

En los países anglosajones los patólogos pueden ser anatomopatólogos o patólogos clínicos. Los anatomopatólogos se dedican al diagnóstico basado en la observación morfológica de lesiones, principalmente a través de la microscopía de luz, utilizando diversos tipos de tinciones. Los patólogos clínicos se dedican al diagnóstico a través de los análisis propios del laboratorio clínico, incluyendo los métodos utilizados en diversas disciplinas como la Hematología, Inmunología, Microbiología y Química clínica. En los países francófonos e hispanos los estudios de células aisladas en fluidos (citología) los realizan también los anatomopatólogos; los análisis de sustancias los realizan los bioquímicos clínicos.

2.1.4 Anatomía patológica

La anatomía patológica se encarga del estudio de las lesiones celulares, tejidos, órganos, de sus consecuencias estructurales y funcionales y por tanto de las repercusiones en el organismo.

La anatomía patológica es uno de los pilares fundamentales de la medicina y una disciplina básica imprescindible para médicos, veterinarios y otros profesionales de la salud.

La interpretación de los síntomas de las distintas enfermedades o alteraciones que se encuentran en la exploración de los pacientes exige el conocimiento de todo el espectro de lesiones que se presentan en cada uno de los tejidos u órganos.

La anatomía patológica comprende todos los aspectos de la enfermedad, fundamentalmente a nivel morfológico. Estas alteraciones son estudiadas con diversos métodos, que abarcan desde la patología molecular hasta la patología macroscópica, pasando por todos los niveles y técnicas de observación y estudio, desde la histoquímica e inmunohistoquímica hasta la ultraestructura, la microscopía óptica (**microscópica**) o la simple visión ocular directa (**macroscópica**).

2.1.5 Fisiopatología

Es el estudio de los procesos patológicos (enfermedades), físicos y químicos que tienen lugar en los organismos vivos durante la realización de sus funciones vitales. Estudia los mecanismos de

producción de las enfermedades en relación a los niveles molecular, subcelular, celular, tisular, orgánico y sistémico o funcional.

2.1.6 Etiología y patogenia

La etiología es el estudio de las causas de enfermedad. El proceso patológico mismo, esto es la serie de cambios patológicos excluidas las causas que la originan, se llama patogenia.

La patogenia puede estudiarse desde distintos puntos de vista, básicamente, del funcional, como lo hace la fisiopatología, o del morfológico, como lo hace la patología general. Ambos se complementan en la comprensión de la patogenia.

La patología general es una morfopatología que consiste fundamentalmente en el estudio de los aspectos morfológicos de la patogenia. Sólo ocasionalmente las causas de la enfermedad son reconocibles por los métodos de la morfopatología.

2.1.7 Génesis causal

La génesis causal representa la explicación de la enfermedad en términos de la noción causa-efecto. Aquí interesa conocer por qué se producen los cambios patológicos y, en particular, por qué se origina la enfermedad. La génesis causal, por lo tanto, es inherente a lo que trata la etiología, pero el análisis causal puede extenderse también a la patogenia. Así, la patogenia aparece comprendida en términos de mecanismos patogenéticos cada uno con una causa y un efecto. Si se trata de mecanismos en cadena, el efecto de uno pasa a ser causa de otro.

2.1.8 Génesis formal

La génesis formal se llama también morfogénesis. En ella interesa saber cómo se producen las alteraciones morfológicas que se suceden en una enfermedad y cuáles pudieran caracterizar el proceso patológico. El estudio comparativo de la génesis formal permite conocer mejor las formas de reacción del organismo, hace posible caracterizar muchas enfermedades y facilita descubrir la génesis causal.

La patología morfológica y morfología en general, se sustentan en el concepto de forma. La forma es expresión de un alto grado de orden de la materia y en los organismos vivientes ella aparece determinada dentro de un plan genético.

En el estudio de la forma en patología hay que distinguir la forma visible y su significado. La forma visible alterada es objeto de la descripción, el significado se expresa en una interpretación de las alteraciones.

2.1.9 Signos y síntomas

Un signo clínico es la manifestación objetiva de una enfermedad o alteración de la salud. Es lo que el médico puede percibir en un examen físico, en contraposición a los síntomas que son los elementos subjetivos, percibidos sólo por el paciente. Por ejemplo: la fiebre, el edema, el enrojecimiento de una zona del cuerpo, son signos clínicos. En cambio, el dolor, la astenia, los mareos, son síntomas.

2.1.10 Diagnóstico

Se refiere al procedimiento con el que se identifica una enfermedad. El diagnóstico clínico requiere tener en cuenta los dos aspectos de la lógica, es decir, el análisis y la síntesis, utilizando diversas herramientas como la anamnesis, exploración física y exploración complementarias.

El diagnóstico médico establece a partir de unos síntomas, unos signos y los hallazgos de exploraciones complementarias, qué enfermedad padece una persona. Generalmente una enfermedad no está relacionada de una forma biunívoca con un síntoma, es decir, un síntoma no es exclusivo de una enfermedad. Cada síntoma o hallazgo en una exploración presenta una probabilidad de aparición en cada enfermedad.

2.1.11 Tipos de diagnóstico

Diagnóstico diferencial: Es el conjunto de enfermedades que pueden ocasionar un síndrome, descartándose una a una según la hipótesis planteada y las exploraciones complementarias, hasta encontrar la enfermedad que ocasiona los síntomas del paciente.

Diagnóstico clínico: Es el que se establece mediante la anamnesis, exploración física y exploraciones complementarias que no sean de anatomía patológica. Tiene menos certeza que el diagnóstico patológico.

Diagnóstico patológico: También llamado anatomopatológico o histológico, que es obtenido mediante una biopsia y es el que ofrece el diagnóstico definitivo en un cáncer.

2.1.12 Historia clínica

La historia clínica es un documento, el cual se registran de los datos clínicos que tengan relación con la situación del paciente, su proceso evolutivo, tratamiento y recuperación, no se limita a ser una narración o exposición de hechos simplemente, sino que incluye juicios, documentos, procedimientos, informaciones y consentimiento informado, consentimiento del paciente; es un documento que se va haciendo en el tiempo, documentando fundamentalmente la relación médico-paciente.

La información contenida en la historia clínica puede obtenerse por diferentes vías que son:

- La anamnesis: Es la información surgida de la entrevista clínica proporcionada por el propio paciente.
- Exploración física o clínica.
- Pruebas o exámenes complementarios realizados por el médico.
- Juicios de valor que el propio médico extrae o de documentos que él elabora para fundar un diagnóstico, prescribir el tratamiento y, finalmente, dejar constancia del curso de la enfermedad.
- Tratamiento instaurado.

2.1.13 Técnicas diagnósticas en histopatología

Si bien el objeto central del estudio teórico de la Patología es la enfermedad como ente, en la práctica el objeto más importante es el diagnóstico de enfermedades y lesiones, lo que tiene mucha trascendencia para el tratamiento y pronóstico en pacientes. Los patólogos usan diferentes

técnicas para generar diagnósticos acertados entre ellas las más utilizadas son la biopsia y la autopsia.

2.1.13.1 Biopsia

Puede definirse como el procedimiento en el que se remueve tejido de un organismo vivo para examen microscópico y así establecer un diagnóstico. La muestra obtenida también se llama biopsia. La principal aplicación de la biopsia es la detección precoz del cáncer. En los casos avanzados, la biopsia se emplea para determinar la naturaleza del tumor maligno y para conocer los efectos del tratamiento

2.1.13.2 Autopsia

La palabra autopsia significa ver por sí mismo y se usa como sinónimo de necropsia o examen post-mortem. Quizás si el mejor término sea examen post-mortem, porque representa en verdad un examen médico después de la vida, cuyos objetivos son la búsqueda de las causas de la muerte, el análisis de la enfermedad básica y de sus efectos y complicaciones en sus aspectos anatómicos y de las consecuencias de la intervención médica. La autopsia permite formular un diagnóstico médico final o definitivo, dar una explicación de las observaciones clínicas dudosas y evaluar un tratamiento dado. Para el cirujano la autopsia proporciona información acerca de las causas de muerte en el postoperatorio, del estado de las suturas y de la presencia de complicaciones quirúrgicas.

Un médico especialista en patología es quien realiza la autopsia. Después de analizar el exterior del cadáver se practica una incisión para observar los órganos internos. Se anota su posición en el cuerpo y se extraen para su análisis visual y con ayuda del microscopio. La autopsia se suele practicar inmediatamente después del fallecimiento, y para realizarla es necesaria la autorización de algún familiar.

El valor de la autopsia puede resumirse en los siguientes puntos:

- Cientos de enfermedades descubiertas y descritas.
- Clasificaciones de innumerables lesiones.

- Control de efectividad de los tratamientos médicos.
- Origen de ideas para tratamientos médicos y quirúrgicos.
- Comprobación del diagnóstico médico.
- Fuente de enseñanza de estudiantes y médicos.
- Fuente de información epidemiológica.

2.2 MUSEO DE PIEZAS ANATOMOPATOLÓGICAS

Los museos de piezas anatomopatológicas constituyen una de las principales fortalezas de las escuelas de medicina que cuentan con la fortuna de poseer uno. Los museos han servido como material didáctico para generaciones completas de estudiantes de medicina alrededor de todo el mundo. Estos tienen como objetivo demostrar a los alumnos las características de las lesiones en distintos órganos permitiendo relacionar la teoría con la práctica.

2.2.1 Definición de museo

Un museo es un lugar donde se conserva todo tipo de colecciones de arte, científicas, etc., siempre con un valor cultural y expuestas adecuadamente para que se puedan contemplar o estudiar. El verdadero objetivo de los museos es la divulgación de la cultura, la investigación, las publicaciones al respecto y las actividades educativas.

2.2.2 Especímenes anatomopatológicos

Los especímenes o piezas anatomopatológicas son órganos, tejidos y partes del cuerpo humano que presentan lesiones. Se obtienen mediante procedimientos médicos, quirúrgicos y autopsia. Son utilizados para la enseñanza de la patología a nivel macroscópico en cursos de Medicina, Bacteriología y Laboratorio Clínico. Cada espécimen puede ser identificado con uno o más diagnósticos de los cuales uno de ellos puede definirse como el más representativo o **diagnostico principal**, los demás diagnósticos son denominados entonces **diagnósticos secundarios**.



Figura 2.1. Especimen anatomopatológico que presenta como diagnóstico principal "Atrofia cerebral" y como diagnóstico secundario "Hidrocefalia".

2.2.3 Descripción macroscópica de una pieza

Es una descripción de la forma visible alterada por la enfermedad y su significado o interpretación. Describe cómo se producen las alteraciones morfológicas que se suceden en una enfermedad y cuáles caracterizan el proceso patológico.

2.2.4 Clasificación de enfermedades

Una clasificación de enfermedades puede definirse como un sistema de categorías a las que se asignan entidades morbosas de conformidad con criterios establecidos. La clasificación puede girar en torno a muchos ejes posibles, y la elección de uno en particular estará determinada por el uso que se hará de las estadísticas recopiladas. Una clasificación estadística de enfermedades debe abarcar toda la gama de estados morbosos dentro de un número manejable de categorías.

2.2.5 CIE-10

La décima revisión de la Clasificación estadística internacional de enfermedades y problemas de salud (CIE-10), también conocida como Lista de códigos ICD-10, llevada a cabo por la Organización Mundial de la Salud. Es la más reciente de una serie que se formalizó en 1893 como Clasificación de Bertillon o Lista Internacional de Causas de Defunción. En la clasificación

actualizada, las afecciones se han agrupado de la manera que se creyó más apropiada para los fines epidemiológicos generales y para la evaluación de la atención de la salud.

2.2.6 Código CIE-10

La CIE-10 propone un sistema de codificación alfanumérico consistente en una letra seguida de tres dígitos. De las 26 letras del alfabeto, se utilizaron 25 y la U se dejó vacante para adiciones y enmiendas futuras, y para posibles clasificaciones provisionales que resuelvan las dificultades surgidas en los ámbitos nacional e internacional entre una revisión y la siguiente¹. Por ejemplo el código CIE-10 para el diagnóstico Atrofia cerebral (Ver figura 2.1) es “G31.9”.

2.2.7 Tecnología del museo de patología

La tecnología de los museos de patología ha sido influenciada por la ciencia de los plásticos y su desarrollo en las últimas décadas, lo anterior junto con las técnicas mejoradas de preservación ha conducido a la formación de un nuevo campo que permite obtener piezas de excelente calidad. En la siguiente fotografía (figura 2.2) se muestran algunas piezas del museo perteneciente a la Escuela de Medicina de la Universidad de Queensland en Brisbane Australia.



Figura 2.2. Museo de patología perteneciente a la Escuela de Medicina de la Universidad de Queensland en Brisbane Australia.

¹ ORGANIZACIÓN MUNDIAL DE LA SALUD. International Statistical Classification of Diseases and Related Health Problems, 10th Revision, 1992.

2.3 OTROS MUSEOS Y GALERÍAS DIGITALES PUBLICADAS EN INTERNET

Como resultado de una labor exhaustiva de búsqueda en Internet, se citarán los museos de piezas anatomopatológicas más destacados y los servicios que ofrecen:

- **Pathology Museum (<http://gsm.herston.uq.edu.au/pathology/museum/home.html>):** Este museo pertenece a la Escuela de Medicina de la Universidad de Queensland en Brisbane Australia consta de una infraestructura física excelente, con piezas conservadas en resina. En el sitio Web, se está considerando sistematizar el museo de la misma manera que se va a hacer en el presente proyecto, pero hasta ahora solo existe una gran colección de fotografías que no se han publicado en su totalidad y una versión de demostración del museo virtual que se puede observar en la siguiente dirección:

http://gsm.herston.uq.edu.au/pathology/museum/Demo_Museum/Condition_index.html

Esta versión ofrece una colección digital de fotografías con información sobre la ubicación física de las piezas dentro del museo y poca información anatomopatológica. La consulta se hace mediante un sistema enlaces estáticos.

- **Pathweb, the virtual Pathology Museum (<http://pathweb.uchc.edu>):** Este sitio ofrece al público una colección digital de fotografías tanto microscópicas como macroscópicas que pueden consultarse por órganos, sistemas fisiológicos y procesos patológicos. Además ofrece información anatomopatológica de las fotografías que publica. Posee una gran cantidad de fotografías y está respaldado por la Escuela de Medicina de la Universidad de Connecticut en Estados Unidos.
- **Museum of Pathology (<http://museum.med.monash.edu.au/>):** En este sitio se ofrece al público una colección digital de fotografías a nivel macroscópico que pueden consultarse por el sistema de palabras clave (como en los motores de búsqueda Web actuales), también ofrece información anatomopatológica que puede accederse mediante consulta ramificada por clasificación patológica. Pertenece a la Facultad de medicina de la Monash University en Melbourne Australia.

- **The Museum of Human Disease Page (<http://web.med.unsw.edu.au/pathmus/>):** En este sitio se ofrece al público una colección digital de fotografías con información anatomopatológica, que se consultan mediante un sistema de búsqueda lineal ramificada basada en enlaces estáticos. Pertenece a la Facultad de medicina de la UNSW (University of New South of Wales) en Sydney Australia.
- **Northern Clinical School Pathology Museum (http://www.med.usyd.edu.au/palms/pathology_museum/):** En este sitio se ofrece al público una colección digital de fotografías con información anatomopatológica, que se consultan mediante un sistema de búsqueda lineal ramificada basada en enlaces estáticos. Pertenece a la Escuela de Patología de la Universidad de Sydney en Australia.
- **Gosden Museum of Pathology (<http://www.fsm.ac.fj/pws/VPM/untitled.html>):** Este sitio ofrece una pequeña galería de fotografías, en forma de miniaturas a escala (thumbnails), pero no ofrece ningún tipo de interfaz de consulta. Contiene muy poca información anatomopatológica. Pertenece a la Escuela de Medicina de las Islas Fiji.

3 EL PROBLEMA

3.1 ESCENARIO DEL PROBLEMA

El Departamento de Patología de la Universidad Industrial de Santander es una unidad académica y administrativa dependiente de la Facultad de Salud, cuyo objetivo es prestar servicios académicos a las escuelas y desarrollar programas de investigación y extensión de conformidad con las políticas y directrices de la Universidad. Para garantizar esto, el Departamento cuenta con instalaciones como las del Museo de Piezas Macroscópicas, que permiten realizar actividades académicas tanto en pregrado como en postgrado.

El museo es una colección de especímenes anatomopatológicos, provenientes de cirugías y autopsias, almacenados en recipientes con formol, cuya función es suministrar material didáctico para la realización de actividades académicas de práctica en pregrado y postgrado, así:

- **Prácticas de patología macroscópica en cursos de pregrado:** son clases de tipo demostrativo con una intensidad de ocho horas semanales, durante las cuales se muestra a los estudiantes los órganos con las enfermedades vistas en clases teóricas anteriores. Los estudiantes pueden manipular y observar detalladamente las piezas. En cada práctica se observan entre 10 y 30 piezas. La actividad se realiza en la morgue del Departamento de Patología de la UIS, para la cual, el técnico lava los especímenes antes de comenzar la práctica y los exhibe en un mesón (mesa de Morgagni, en la cual se realizan las autopsias). El profesor, realiza una exposición demostrando en cada órgano las características morfológicas de las lesiones, haciendo un breve repaso de la fisiopatología de la enfermedad. Posteriormente, los estudiantes que están alrededor de la mesa (35 por grupo), se acercan a observarlos, examinarlos, palparlos y a aplicar lo visto en la clase magistral días antes. Al cumplirse el tiempo de la práctica, el estudiante cambia de salón y se dirige al salón de práctica microscópica en el cual verá la representación microscópica de las lesiones que vio microscópicamente en la morgue. La metodología de la práctica y el lugar de práctica, hacen

un ambiente poco propicio para el aprendizaje, ya que por la forma de ubicar las piezas macroscópicas, hay hacinamiento de estudiantes, que sumado a los vapores irritantes del formol hacen que éste no pueda realizar la elaboración de conceptos, proceso mental esencial para el aprendizaje significativo.

- **Examen de patología macroscópica:** se realiza en cursos de pregrado cuatro veces por semestre durante cuatro horas, es un examen escrito con preguntas sobre las piezas. Se utilizan de 10 a 15 piezas.
- **Clases de patología macroscópica para el estudiante de postgrado:** son clases similares a las realizadas en pregrado, con la diferencia de que se dicta a uno o dos estudiantes, su periodicidad es de dos horas semanales. Se utilizan máximo dos piezas.
- **Reuniones Ínter-departamentales para cirugía y ginecología:** son demostraciones realizadas ante el grupo de profesores y estudiantes de postgrado de cirugía o ginecología, no hay manipulación de las piezas, se realizan semanalmente y su duración es de una hora. Se utilizan una o dos piezas.

Antes de cada práctica el técnico debe extraer las piezas de los recipientes con formol e introducirlas en recipientes con agua, esto con el fin de reducir la cantidad de formol impregnado en las piezas.

El formol o formaldehído (HCHO) es un gas volátil, ligeramente más pesado que el aire, incoloro y muy soluble en agua. Sus vapores irritan todas las partes del sistema respiratorio superior y también afectan a los ojos¹.

Los recipientes cuya capacidad oscila entre 0.5 y 5 litros, son de vidrio, poseen tapa metálica de rosca, pueden contener uno o más especímenes y cuentan con una identificación donde se especifica el nombre del órgano y su diagnóstico.

¹ MORET DE ARCIA, Olga J. Contribución al Estudio de los Efectos Tóxicos del Formaldehído. Universidad de los Andes Facultad de Medicina Departamento de Ciencias Morfológicas Unidad Académica de Anatomía Humana. Mérida, Venezuela 1990.

La colección se encuentra organizada por órganos en estantes metálicos de varios niveles. Cuenta aproximadamente con 450 piezas de las cuales se estima que el 30% ya no son aptas para las demostraciones por su mal estado de preservación; además, cada semana se procesan, como máximo, dos piezas quirúrgicas que pasan a formar parte de la colección.



Figura 3.1. Estanterías del museo de patología antes del proceso de organización. Se evidencia el problema de desconocimiento de la ubicación.

El manejo del museo está a cargo de un técnico, que saca y guarda las piezas por solicitud de los profesores, quienes las exhiben a los estudiantes de pregrado y postgrado en las clases practicas programadas para cada semestre.

Actualmente el departamento de patología ha iniciado el proceso de organización del museo a través de la propuesta de innovación pedagógica para la asignatura de Patología, cuyo objetivo es mejorar las condiciones en las que se encuentra el museo y de esta manera favorecer la ejecución de estrategias pedagógicas para disminuir la mortalidad académica en esta asignatura.

3.2 PROPUESTA DE INNOVACIÓN PEDAGÓGICA PARA LA ASIGNATURA DE PATOLOGÍA

El problema que motiva la realización de ésta propuesta, se plantea desde dos perspectivas: una, la académica y otra, la de salud ocupacional y bienestar estudiantil.

Con respecto a la primera, vale mencionar que la asignatura de Patología, ocupa el cuarto lugar dentro de las materias de mayor mortalidad académica de la Escuela de Medicina de la UIS, motivo por el cual, los profesores de esta dependencia han tomado la decisión de realizar estrategias que favorezcan no solo el aprendizaje significativo, sino la disminución de número de estudiantes repitentes. Una de las estrategias es la aplicación de los conocimientos adquiridos en las sesiones magistrales, en los laboratorios de anatomía patológica tanto macroscópica como microscópica, mediante la cual los estudiantes logran correlacionar éstos conocimientos con los cambios morfológicos de los órganos, producidos por las enfermedades basados en la comprensión del mecanismo fisiopatológico que condujo a este cambio anatómico, de manera que los estudiantes logren inferir los síntomas y signos que el paciente con dicha lesión pueda presentar.

En cuanto al problema de salud ocupacional se puede decir que no sólo se debe a la exposición de estudiantes y profesores al formaldehído, sino que durante el tiempo de la práctica, están entrando cadáveres provenientes del Hospital Universitario, de modo que en una hora de práctica, se puede estar acompañado de uno a cinco cadáveres. En numerosas ocasiones, la practica se debe realizar al mismo tiempo que en el mesón conjunto uno de los docentes del departamento está realizando una autopsia, lo cual hace que el grupo de estudiantes esté expuesta a los vapores emanados por el cadáver y a posibles agentes infecciosos causantes de la muerte del paciente, que podrían contagiar a algún integrante del grupo.

Por todas las razones anteriormente expuestas, y con el fin de favorecer la implantación de nuevas estrategias pedagógicas, se pretende adaptar el actual aula de micropatología para integrar los dos tipos de práctica (la macroscópica que actualmente se realiza en la morgue y la microscópica), organizar el museo existente mediante códigos por la Clasificación Internacional de Enfermedades¹, y cambiar el modo de conservación de los órganos, de modo que sean éstos

¹ ORGANIZACIÓN MUNDIAL DE LA SALUD. International Statistical Classification of Diseases and Related Health Problems, 10th Revision, 1992.

incluidos en resinas o en recipientes herméticos de acrílico con formol o glicerina. Con lo anterior, se favorece el tiempo que los estudiantes dedicarían a estudiar cada especimenes al no estar expuestos a los vapores irritantes del formol y al permitírseles acceder a éstos pues casi desaparecería el riesgo de deterioro de las piezas por la protección que le conferirían los nuevos recipientes.

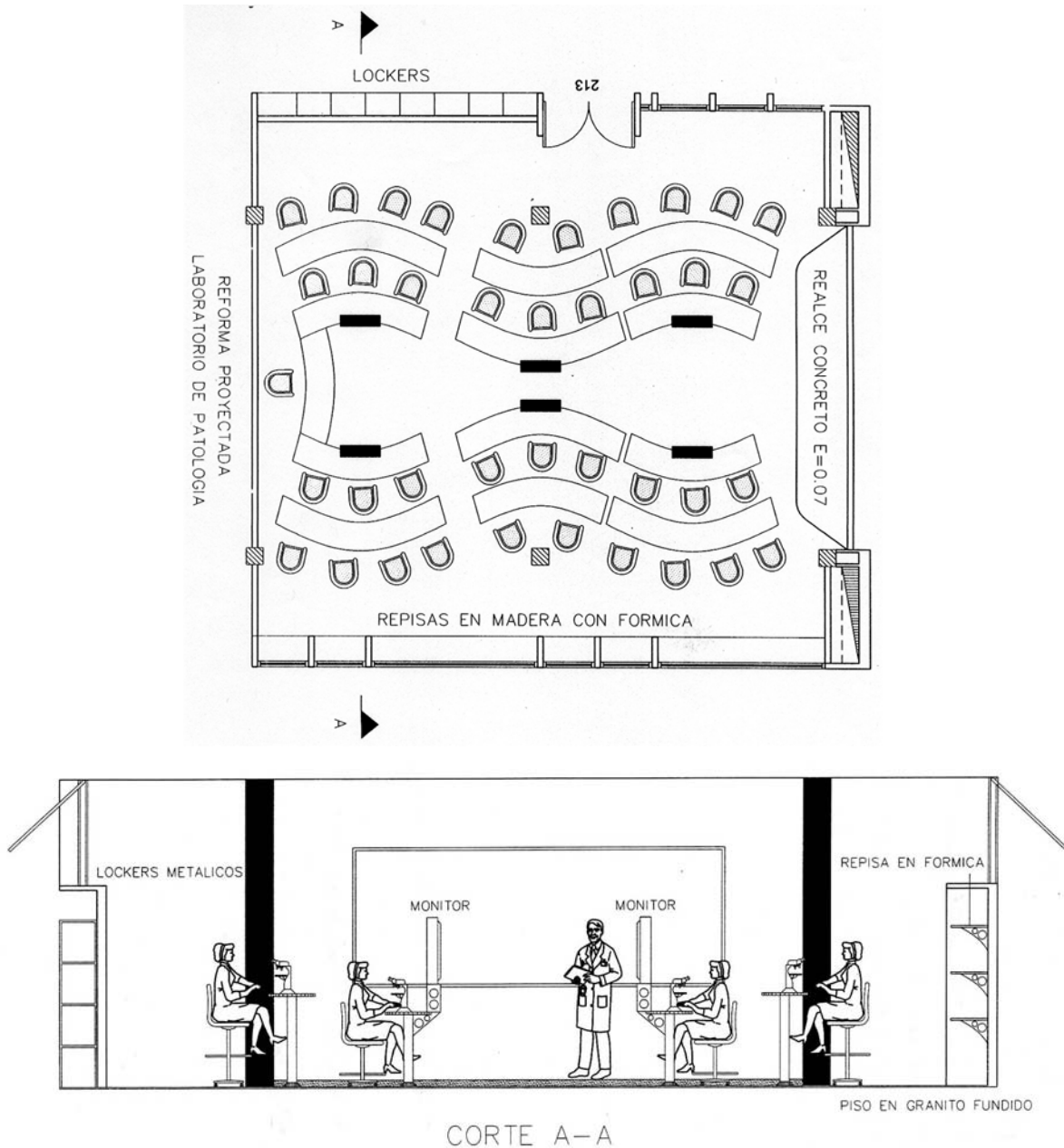


Figura 3.2. Planos de la reforma proyectada para el laboratorio de patología. (Arriba) Vista superior. (Abajo) Vista lateral.

Dado que la propuesta de innovación pretende fortalecer y mejorar los servicios que presta el museo a la comunidad académica de la Facultad de salud se han propuesto dos proyectos de grado simultáneos y complementarios; uno de ellos a cargo de dos estudiantes de postgrado de patología que consiste en organizar, clasificar, seleccionar, describir y fotografiar cada espécimen anatomopatológico, con el fin de mejorar las condiciones del museo, conocer las existencias y el estado de preservación de las piezas que conforman la colección. El otro proyecto a cargo de dos estudiantes de ingeniería de sistemas para apoyar las labores de organización a través de la construcción de una herramienta software que permita dar el soporte informático al nuevo museo.

La ejecución de la propuesta de innovación permitirá resolver la pregunta de investigación ¿Cómo es el rendimiento académico de los estudiantes de la asignatura de patología que utilizan la herramienta de trabajo, comparado con el de los estudiantes que no tuvieron la oportunidad de utilizarla? Para esto se diseñó un proyecto de tipo prospectivo con un análisis univariado y bivariado, que pretende comparar si hay diferencias significativas entre estos dos grupos de estudiantes.

3.2.1 Captación de los estudiantes y metodología de la actividad

Para la realización de la investigación, se requiere la adecuación del aula y la inclusión de los órganos en resinas, así como de la sistematización de las existencias y creación de la herramienta software.

Debido a que la implementación de la estrategia toma seis meses y a que las carreras antes mencionadas son anuales, se contará con un grupo de estudiantes de Bacteriología y Laboratorio Clínico durante el primer semestre del año 2007 y un grupo de Medicina el segundo semestre del 2007.

Al iniciar el semestre, se explicará al grupo el proyecto de investigación utilizando la innovación pedagógica. Posteriormente se realizará una entrevista personal, en la cual se llenará el formulario con los datos sociodemográficos de cada uno.

La metodología de la actividad académica será la siguiente, el estudiante recibirá las clases teóricas en auditorio (clase magistral). Al terminar la unidad, el estudiante deberá resolver en

tiempo dirigido la guía que corresponde a la práctica del tema visto en la clase teórica. El desarrollo de ésta se hará basándose en la información fotográfica macroscópica y microscópica de los órganos enfermos contenida en la página Web del Museo Virtual de Patología de la UIS. Esta guía, deberá entregarse resuelta antes de entrar a la clase práctica al docente responsable de la actividad.

Para comprender el desarrollo de la actividad de clase, es necesario describir la distribución del aula, la cual está compuesta por cuatro grupos o módulos de siete puestos y dos grupos o módulos de cinco puestos, Cada puesto tiene su respectivo microscopio. El módulo se conforma de dos hileras semicirculares, que se ubican una delante de la otra; siendo anterior la de tres puestos, ambas orientadas hacia un monitor de computador. Los estudiantes de la hilera anterior, tiene la posibilidad de girar 180 grados para interactuar con los compañeros y hacer grupos de trabajo.

El profesor, se ubica en un extremo del aula, en donde tiene acceso a un monitor de computador, a través del cual puede explicar en detalle las características de los órganos.

La clase consiste en desarrollar la guía que previamente ha sido entregada por los estudiantes, de manera que el profesor oriente no solo la observación de las características macroscópicas de los órganos y microscópicas de los tejidos pertenecientes al órgano que se está estudiando, sino se realice un análisis fisiopatológico de la enfermedad haciendo la correlación entre los hallazgos morfológicos y los síntomas y signos que pudo haber presentado ese paciente.

En cada clase, se estudiarán máximo cinco enfermedades, por lo que al llegar al aula, el estudiante encontrará en la mesa como mínimo un ejemplar de cada enfermedad que se va a analizar durante la clase. De manera que en el aula se tendrá como mínimo, cinco órganos, cada uno de distinta enfermedad.

El profesor, orientará la descripción de cada órgano desde su monitor, utilizando el mouse, para demostrar las características físicas de las lesiones, como contornos, formas, color, profundidad y extensión (entre otras) de la lesión. Éste, inicia con el ejemplo que tienen los estudiantes del módulo número 1, éstos estudiantes y todos los demás estarán observando en detalle en su respectivo monitor lo que el profesor está detallando, además de esto, los estudiantes del módulo 1

tendrán la oportunidad de ubicarse en mesa redonda y observar en grupo aquello que el profesor está explicando desde su monitor para todo el grupo. De ésta forma, todos los estudiantes pueden observar en detalle no solo su ejemplo, sino disfrutar de la explicación de los ejemplos de los compañeros de otros módulos.

Posteriormente, después de analizar la imagen macroscópica y comprender la fisiopatología de la enfermedad, cada estudiante verá la lesión del tejido desde el punto de vista microscópico, la cual será también explicada desde su monitor para todo el grupo por el profesor. Una vez terminadas las demostraciones, el profesor podrá resolver las dudas particulares visitando cada grupo de estudiantes. A su vez, los estudiantes tendrán la oportunidad de visitar otros módulos para observar en vivo los órganos lesionados de los demás.

De éste modo, los estudiantes lograrán la interacción teoría práctica, y la interrelación entre lesión macroscópica con microscópica, de manera que se logran integrar aún más el conocimiento.

Se seleccionará una muestra aleatoria de las guías de cada clase para calificarlas, de manera que éstas se convierten en un instrumento objetivo de evaluación diagnóstica que podrá compararse con la evaluación sumativa de los previos prácticos.

A la mitad del período, se aplicará una encuesta a los estudiantes para que evalúen el grado de satisfacción con la estrategia implementada.

Los estudiantes tendrán la posibilidad de acceder en sus horas libres o trabajo dirigido al museo con el fin de repasar lo visto en clase y preparar las prácticas siguientes.

3.3 DEFINICIÓN DEL PROBLEMA

Después de conocer el planteamiento inicial del problema propuesto por el Departamento de Patología, se decidió hacer un estudio preliminar para comprender y definir la problemática del Museo y esbozar una solución. Éste estudio preliminar se abordó con la creación del Comité de Sistemas de Información del Museo, que se reúne todos los jueves en horas de la tarde, y cuyos integrantes son:

- Director del departamento de patología:

Jorge Humberto Echeverri Perico, médico cirujano especialista en patología.

- Profesores de planta del departamento de patología:

Olga Mercedes Álvarez Ojeda, médica cirujana especialista en patología, directora de la tesis de postgrado para la organización física del museo y codirectora del proyecto SIMUPAT.

Julio Cesar Mantilla Hernández, médico cirujano especialista en patología, director de la tesis de postgrado para la organización física del museo.

- Estudiantes de postgrado del departamento de patología, autores del proyecto para la organización física del museo:

Pedro Vicente Duarte Ballén, médico cirujano, residente de segundo año de la especialización en patología.

Mariela Liliana Mejía, médica cirujana, residente de segundo año de la especialización en patología.

- Personal Técnico del museo:

Genaro Barchelot Gutiérrez, técnico de procesamiento quirúrgico.

- Estudiantes de la escuela de medicina, representantes del grupo de colaboradores del proyecto SIMUPAT:

Adriana Marcela Barajas Carreño.

Andrea Milena González Gómez.

- Estudiantes de la escuela de ingeniería de sistemas e informática, autores del proyecto SIMUPAT y del presente documento:

Omar Saúl Duarte Ballén.

Álvaro Martínez Duarte.

Como resultado de las reuniones del comité y de la visita a las instalaciones del Museo, en donde se observó la operación cotidiana del Museo en circunstancias normales, se llegó a la descripción

del escenario problema y las funciones que viene desempeñando actualmente, la cuales fueron consignadas en el documento de Especificación de Requisitos de Software (ERS) versión 1.0 que fue revisado por el comité de Sistemas de Información durante las reuniones semanales. Al mismo tiempo se pudieron identificar los siguientes problemas:

- **Desconocimiento de la ubicación:** El técnico ubica las piezas fácilmente gracias a la experiencia conseguida a lo largo de varios años de trabajo en el museo, si por alguna circunstancia no puede asistir al trabajo los profesores deben remplazarlo temporalmente, lo que implica que para buscar una pieza sea necesario abrir varios recipientes antes de encontrarla, esto ocasiona retardos en la ejecución del procedimiento y ocupa tiempo valioso que los profesores pueden invertir en actividades propias a su rol (ver figura 3.1).
- **Desconocimiento de las existencias:** Al no tener datos exactos sobre el nivel de existencias, se aumenta el riesgo de pérdida de piezas y se dificulta la planificación de las actividades académicas, fomentando así que la selección de piezas para las demostraciones obedezca al criterio del técnico y no al del profesor como debería ser.
- **Desconocimiento del estado de preservación:** Se desconoce el estado de preservación de cada pieza, haciendo que en muchas oportunidades se escojan especímenes que ya no son óptimos para la enseñanza (ver figura 3.3).



(a)



(b)

Figura 3.3. (a) Fotografía de un recipiente que contiene una pieza en estado de putrefacción. (b) Fotografía donde se pudo observar que la pieza se adhirió al fondo del recipiente.

- **Disponibilidad limitada:** Se está desaprovechando el gran potencial educativo y didáctico del museo al no seleccionar especímenes idóneos para las prácticas, y no exhibir su colección en espacios diferentes al aula de clase.
- **Identificación incorrecta de especímenes:** Presencia de especímenes anatomopatológicos ubicados en recipientes que están rotulados con diagnósticos diferentes a los que corresponden. Lo anterior genera confusión en el momento de la práctica, haciendo que los profesores deban invertir tiempo para hacer una segunda selección de material didáctico.

El estudio preliminar mencionado plasmó sus resultados en la primera versión de la Especificación de Requerimientos de Software, en donde se tuvo una comprensión de la magnitud del sistema a desarrollar y las principales funciones que el mismo debe desempeñar para solucionar los problemas enunciados, de forma satisfactoria.

Sin embargo, durante las reuniones del Comité de Sistemas de Información se decidió por parte del Departamento de Patología hacer una reestructuración a fondo del Museo, de tal manera que el escenario descrito anteriormente presenta los siguientes cambios:

- **Cambios en la metodología de enseñanza y evaluación:** El cuerpo docente del departamento de Patología replanteará el horario y metodología de enseñanza y evaluación, para aprovechar de una mejor forma los recursos disponibles, especialmente la colección del Museo.
- **Integración del Museo con el aula de prácticas microscópicas:** El museo se integrará con el aula de prácticas microscópicas (ver figura 3.2), es decir, en el mismo salón se dispondrá de microscopios con sus correspondientes placas y otro material como: televisores, mesas, tableros, computadores, etc.
- **Cambio en el método de conservación de los especímenes (piezas) anatomopatológicos:** Las piezas ya no se conservarán en recipientes de vidrio llenos de formol, sino que se embeberán en una resina transparente (figura 3.4), que al estar libre de aire, conservará dichas piezas en perfecto estado y libre de efectos contaminantes nocivos

para sus usuarios y administradores. Además la colección de especímenes puede ser guardada en armarios similares a los de una biblioteca.

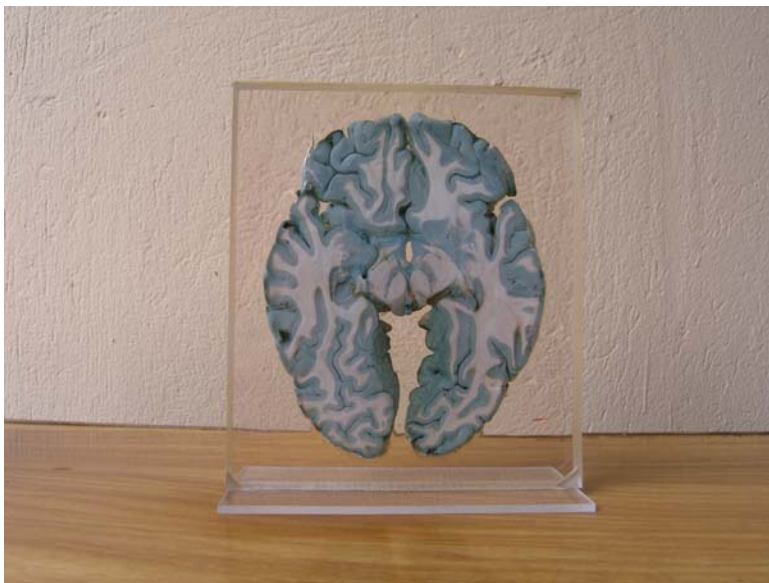


Figura 3.4. Especímen anatomopatológico incluido en un bloque de resina perteneciente a la colección de la Universidad Autónoma de Bucaramanga.

Mientras se realizaba la fase de análisis del proyecto SIMUPAT; los residentes de patología, supervisados por la Doctora Olga Álvarez y el Doctor Julio Cesar Mantilla; con la ayuda de el grupo de estudiantes colaboradores; reorganizaron y prepararon las piezas para el proceso de inclusión en resina del nuevo museo.

En conclusión, el presente proyecto informático se encargará de construir una herramienta software que permitirá solucionar los problemas de desconocimiento de la ubicación, desconocimiento de las existencias, identificación incorrecta de especímenes y disponibilidad limitada, que fueron inspirados por la propuesta inicial y aun prevalecen en el nuevo Museo de Patología, del cual el sistema será parte integral y contribuirá a su correcta administración, mantenimiento, y publicación del material fotográfico, e informativo. Pero, en vista de que el proceso de reestructuración del Museo supera la duración de este proyecto, no se contemplará la fase de implantación de la herramienta a desarrollar, aunque existe el compromiso explícito de los autores con el Departamento de Patología de llevarla a cabo.

4 MARCO TECNOLÓGICO

Esta sección tiene como objetivo familiarizar al usuario sobre los diferentes conceptos técnicos involucrados en el desarrollo del presente proyecto, de tal forma que esté mejor preparado para leer las diferentes secciones de este libro. Esta sección ha sido escrita por los autores, de tal forma que sea lo más explícita y sencilla para que los lectores sin conocimientos en el área de la ingeniería de sistemas pueda entenderla sin mayores dificultades y por lo tanto se limita a generalidades que un profesional de los sistemas encontrará insuficiente pero que serán de gran ayuda al neófito para entender los conceptos tratados en el presente libro.

4.1 LOS SISTEMAS DE INFORMACIÓN

SIMUPAT es, como su nombre lo indica, un sistema de información. ¿Pero qué es un sistema de información? Aunque muchas definiciones pueden encontrarse en la literatura disponible, que es realmente abundante, los autores se atreven a dar su propia definición: un sistema de información es un software cuya función es administrar la información que es importante para el funcionamiento de una organización. ¿Pero qué significa administrar la información? En este contexto administrar información implica:

- Compartir la información con cualquier miembro de la organización que la necesite y que esté autorizado para utilizarla.
- Reducir, en lo posible, el tiempo necesario para que un usuario tenga acceso a la información que necesita.
- Evitar en lo posible la redundancia en la información, es decir, que el costo de mantener la información almacenada sea lo menor posible.
- Mantener actualizada, en todo momento, la información sin importar que diferentes usuarios estén operando con ésta. El costo por mantener actualizada dicha información debe ser lo menor posible.

- Extender, en lo posible, la cobertura; es decir, que la información esté disponible al mayor número de usuarios ubicados en un área física mayor.
- Volver transparente al usuario las operaciones técnicas involucradas con la administración de la información.

Es posible que el lector encuentre que administrar la información conlleva mucho más que lo enunciado aquí, y de hecho tiene razón, pero para el objetivo de esta sección será suficiente. Los sistemas de información nacieron de la necesidad de mantener organizada la información de la organización y de agilizar los procesos de ésta con el objetivo de ganar tiempo, y por ende dinero.

A medida que los computadores y las tecnologías de la información y la comunicación han venido desarrollándose, las exigencias a los sistemas de información han aumentado. Ahora se hace necesario que los empleados tengan acceso a la información, no solo desde sus puestos de trabajo, sino desde sus hogares y sitios de descanso. Los ejecutivos por ejemplo necesitan la información en pleno vuelo, los ingenieros civiles en el campamento de obras, los médicos donde se encuentren sus pacientes, etc.

Antes del advenimiento de la Internet, o mejor dicho, de la popularización de Internet como medio masivo de comunicación, las empresas poseían sus sistemas de información implantados en redes particulares, con hardware y software específicos para cada una de ellas. Este fenómeno conllevaba a un inmenso margen de costo para hacer mantenimiento, soporte y mejoras a los sistemas de información, ya que la organización no solo debía invertir dinero en mejorar el software del sistema, sino también en mejorar el hardware de la red en la que estaba implementado; labor que únicamente podía realizar personal calificado por la marca propietaria de la tecnología de la red conllevando costos inmensos de recursos humanos.

Pero con la masificación de Internet el panorama cambió radicalmente: las redes comenzaron a estandarizarse en redes para Internet y el alcance prácticamente global de Internet comenzó a brindar grandes oportunidades de ampliar la cobertura de los sistemas de información de una forma mucho más económica que antes: nacieron los sistemas basados en Web.

4.1.1 Los sistemas de información basados en Web

Se dijo que con la masificación de Internet los desarrolladores de sistemas de información y las compañías vieron una gran oportunidad de expandir sus negocios y la cobertura de sus sistemas. Los sistemas de información que se ejecutan en Web son apreciados cuando se desea publicar contenido y brindar a clientes en todo el mundo los servicios de la organización. Este es el caso de SIMUPAT.

Un sistema de información basado en Web es, básicamente, un sistema de información que utiliza la Web como medio de comunicación entre sus partes y los usuarios. Tiene la gran ventaja que un usuario necesita únicamente de una conexión a Internet y un navegador para ser cliente del sistema desde cualquier parte del mundo y las veinticuatro horas al día. De hecho, a la fecha de escritura de este libro, no es necesario que sea un computador personal el que haga la conexión: los teléfonos celulares, las agendas electrónicas, cualquier dispositivo con conexión a Internet puede, en teoría, conectarse con un sistema Web.

4.2 INTERNET

Un sistema basado en Web es uno que use Internet, y más específicamente el HTTP y el HTML para la comunicación con sus usuarios. Antes de profundizar en estas siglas demos un vistazo breve a la historia de Internet y a los diferentes servicios que ofrece.

Internet, la red internacional, nació en la década de los 60 como Arpanet, una tecnología de telecomunicación financiada por las fuerzas militares norteamericanas (específicamente el DARPA) para sus sistemas, mediante líneas telefónicas conmutadas. Cuando esta tecnología fue liberada del secreto por el gobierno norteamericano, se continuaron los trabajos para difundirla y mejorarla, hasta convertirla en lo que hoy conocemos.

Internet es mucho más que las páginas que navegamos a diario desde los cafés y o nuestros hogares, lo que constituye el Web, y fue utilizado en un principio como medio de comunicación militar, transferencia de noticias entre agencias internacionales y medio de divulgación de trabajos científicos. Actualmente Internet es un gigantesco sistema distribuido, es decir que no tiene un núcleo o componente central, que presta, entre otros, los siguientes servicios:

4.2.1 WWW

El servicio de navegación Web o WWW (*World Wide Web*) es sin duda el servicio más utilizado de Internet. Consiste en la publicación de contenido interactivo, en unidades llamadas páginas Web, al cual el usuario puede acceder con tan solo ordenar a su navegador que vaya a la dirección URL del sitio respectivo.

La Web es un desarrollo que nació décadas después del nacimiento de Internet, en su forma primitiva de Arpanet, y que comenzó en 1989 como un proyecto del CERN para facilitar la divulgación y lectura de trabajos científicos, de tal forma que el lector pudiera revisar las referencias de un artículo determinado mientras lo estuviera leyendo. Tim Berners-Lee fue quien desarrolló el primer prototipo que se convertiría en el núcleo de la World Wide Web.

El concepto fundamental tras la WWW es de hipertexto, es decir, texto interactivo que conectara varios documentos entre sí de tal forma que alguien pudiera acceder a una determinada referencia con tan solo dar clic sobre ella.

Los conceptos claves en el funcionamiento de la WWW son: el localizador uniforme de recursos o URL, el lenguaje HTML, los navegadores Web y el protocolo HTTP.

4.2.1.1 El localizador uniforme de recursos (URL)

URL (*Uniform Resource Locator*) es una forma de caracteres, cuya sintaxis obedece a reglas establecidas, para la notación de los diferentes recursos publicados en Internet de tal forma que el explorador pueda localizarlos, visualizarlos e identificarlos de manera única. Esta secuencia de caracteres está compuesta por:

- El nombre del servicio Internet (HTTP, FTP, GOPHER, etc.)
- El nombre de dominio del sitio Web
- La ruta de acceso al documento dentro del servidor que aloja el recurso.

4.2.1.2 El lenguaje HTML

El lenguaje HTML, cuya sigla significa Lenguaje de Marcado de Hipertexto (*Hyper Text Markup Language*) es un lenguaje diseñado para la publicación de contenido interactivo en Internet.

Consiste en una serie de etiquetas, las cuales se pueden anidar, que proveen de ciertas propiedades al texto que está encerrado dentro de ellas. La masificación de Internet ha ocasionado que el HTML se haya convertido en uno de los formatos más populares para la elaboración de documentos y es uno de los más fáciles de aprender.

HTML es una aplicación SGML conforme al estándar ISO-8879. XHTML es una adaptación del HTML basada en el lenguaje XML.

4.2.1.3 Los navegadores Web

Los conocidos navegadores o *browsers* son programas cuyo objetivo es transformar el código HTML en lo que vemos en las pantallas de nuestros computadores cuando navegamos en Internet. Aunque la definición del HTML es estándar, hay que decir lamentablemente que a pesar de los esfuerzos de la W3C (la entidad encargada de regular el estándar en la Web) no han sido del todo fructíferos. Los diferentes navegadores, en su lucha por la conquista del mercado de Internet, han introducido miles de mejoras y código a su manera creando diferencias notables a la hora de mostrar un determinado sitio Web. Una página en un navegador A tiene muchas probabilidades de no verse igual en un navegador B y es posible que no pueda siquiera verse en un navegador C.

Los navegadores web más conocidos a la fecha de escritura de este documento son:

- Internet Explorer, de Microsoft Corp.
- Firefox, de Mozilla.
- Mozilla.
- Konqueror.
- Opera.
- Lynx.

4.2.1.4 El HTTP

Resulta evidente que es necesario establecer un idioma común de comunicación entre un computador que navega en Internet y los servidores en donde se alojan las páginas visitadas por

dicho computador. Este idioma común, llamado protocolo en la jerga técnica, es el HTTP (*Hyper Text Transfer Protocol*) que es el protocolo utilizado para la transmisión de archivos de hipertexto (HTML) para su visualización en los navegadores. El HTTP está en su versión 1.1 y se encuentra consignado en el documento RFC-2616.

Una de las desventajas principales del HTTP es que no soporta estados, es decir, la información de una determinada conexión se pierde cuando se termina la comunicación, por lo que no hay información sobre las conexiones anteriores. Esta desventaja ha sido solucionada parcialmente por las *cookies*, archivos de texto plano dejados en la máquina del cliente con la información que desea guardar.

El HTTP tiene una versión que soporta cifrado SSL (*Secure Sockets Layer*), o capa de zócalos seguros, llamada HTTPS.

4.2.1.5 El servicio de transferencia de archivos o FTP

Además del servicio de páginas Web, Internet ofrece un servicio diseñado para la transferencia de archivos llamado FTP (*File Transfer protocol*), o protocolo de transferencia de archivos por sus siglas en inglés. Como este protocolo no es de especial importancia para el entendimiento de SIMUPAT dejaremos que el lector interesado profundice en este tema consultando otras fuentes.

4.2.2 Los servicios de correo electrónico

Después de la navegación de páginas Web, el servicio de envío y recepción de correo vía Internet es el servicio que más se utiliza por parte de los usuarios de Internet en todo el mundo. Es prácticamente inconcebible que un usuario de Internet no haya enviado o recibido nunca un correo electrónico, que comparado con el correo ordinario, tiene un costo mucho menor y tiempos de entrega prácticamente instantáneos.

Aunque los clientes de correo electrónico basados en Web se han vuelto en extremo populares, el lector debe comprender que son servicios totalmente diferentes. Los sitios Web que ofrecen correo electrónico con simplemente una interfaz entre el usuario y los programas que se encargan de la emisión y recepción de e-mail.

4.3 EL MODELO OSI Y LAS REDES DE COMPUTADORES

Ya que hemos efectuado un vistazo muy rápido sobre Internet y los servicios más importantes que ofrece, es momento de retomar las consideraciones que habíamos dejado pendientes sobre el estándar en el diseño e implantación de redes de computadores.

El modelo OSI (*Open System Interconnection*) o modelo de interconexión de sistemas abiertos, nació en 1984 como un modelo de red descriptivo creado por la Organización Internacional de Estándares (ISO). El modelo proporciona, a los fabricantes de tecnologías de redes de computadores, estándares que buscan garantizar una mayor compatibilidad e interoperabilidad entre las diferentes tecnologías de red producidos por dichos fabricantes a nivel mundial.

Lo que llevó al desarrollo del modelo OSI fue el crecimiento, rápido y desordenado, en la cantidad y el tamaño de las redes de computadores. A medida que las empresas vieron las ventajas que conlleva la implantación de una red corporativa comenzaron a implantar tecnologías de red a tal ritmo que las redes se agregaban o expandían casi a la misma velocidad con que aparecían las tecnologías de red.

En la década de los 80, las empresas comenzaron a sentir los efectos de esta expansión desordenada, ya que no podían interconectar sus redes corporativas por estar construidas usando tecnología diferente. De igual forma los costos de mantenimiento de dichas redes requerían personal altamente calificado en una tecnología en particular, lo cual conllevaba un alto costo de mano de obra. Por último los fabricantes de las distintas redes protegían en patentes las características técnicas de su tecnología, lo cual hacía muy difícil la creación de interfaces de comunicación entre diferentes tecnologías de red.

Ante esta situación, la ISO investigó modelos de interconexión como la red de Digital Equipment Corp (DECNet), la arquitectura de sistemas de red (SNA) y el protocolo TCP/IP a fin de encontrar características comunes que llevaran a la formulación de un estándar. El resultado se llamo modelo de interconexión de sistemas abiertos, abreviado OSI.

El modelo OSI contempla siete capas, cada una de las cuales contiene programas para llevar a cabo tareas específicas. Es necesario aclarar que este modelo no es considerado una arquitectura,

pues no especifica los protocolos que deben usarse en cada capa, sino una referencia para el desarrollo de arquitecturas de red.

4.3.1 Capa de aplicación

Esta capa provee a las aplicaciones, sean de usuario final o no, la posibilidad de acceder a los servicios de las capas inferiores y de definir los protocolos utilizados en el intercambio de datos. Es muy raro en caso en que el usuario interactúa con el nivel de aplicación, por lo general los que interactúan con esta capa son interfaces que vuelven transparente para el usuario la complejidad de los servicios de esta capa.

Las tecnologías más comunes presentes en esta capa son:

- HTTP
- FTP
- SMTP
- Sendmail
- POP
- SSH
- TELNET
- SNMP
- DNS

4.3.2 Capa de presentación

Esta capa se encarga de representar la información a pesar del formato en el que haya sido construida, de tal forma que aunque dos equipos, por ejemplo, tengan diferente representación interna de los caracteres, sonido o imágenes, los datos lleguen de forma reconocible.

Además esta capa puede realizar labores de compresión y cifrado de los datos.

4.3.3 Capa de sesión

Esta capa se encarga de ofrecer servicios que se relacionan con la forma de administrar la comunicación, tales servicios son:

- Control de la comunicación entre el emisor y el receptor, es decir, quien transmite, quien escucha y como se mantiene la conexión.
- Controla que no haya concurrencia, es decir, que no colisione la comunicación con otra en la misma operación crítica.
- Mantiene puntos de verificación o *checkpoints* por si la comunicación se interrumpe, pueda ser retomada en donde se estableció el último punto de verificación y no desde el principio.

4.3.4 Capa de transporte

Esta capa se encarga de dividir los datos en unidades, más grandes que las tramas, llamadas paquetes y pasarlos a la capa de red para su transmisión. Además debe aislar completamente las tecnologías de las tres capas inferiores (física, enlace y red) de las tres superiores (sesión, presentación y aplicación) de tal forma que la comunicación no se vea afectada si en uno u otro nivel se ha implementado una tecnología diferente. Además presta diversos servicios a la capa de sesión según las exigencias de la comunicación, es decir, si se requiere que los paquetes lleguen estrictamente ordenados, es la capa de transporte la que debe acometer la responsabilidad que así sea.

4.3.5 Capa de red

Esta capa es la encargada de administrar el viaje de los datos por la red, es decir, de guiar los datos desde su origen hasta su destino, aún cuando no exista una conexión directa entre ellos. Esta capa se encarga de encontrar un camino desde el emisor al receptor, labor efectuada mediante tablas de enrutamiento, y de proporcionar un esquema que identifique a cada equipo conectado a la red (esquema de direcciones).

Además, esta capa debe solventar el problema de la congestión o atascamiento de la red, que es una situación que ocurre cuando un nodo de la red tiene una cola de espera muy alta y queda

saturado; esta capa debe evitar el envío de datos a dicho nodo congestionado y hacer que se tome una ruta no congestionada en lo posible.

Las tecnologías más comunes en esta capa son:

- IP
- IPX

4.3.6 Capa de enlace

Provee mecanismos para la detección y corrección de errores de tal forma que, sin importar el medio de transmisión, se establezca una conexión fiable a través de un enlace físico. Además de reconocer y establecer los límites de las tramas de bits, esta capa debe solventar los problemas derivados de la corrupción, duplicación o pérdida de las tramas de información a través del medio de transmisión, así como soluciones a los problemas de “cuello de botella”. Esta capa de enlace se encarga de aspectos como:

- Topología de la red.
- Direccionamiento a nivel físico.
- Notificación de errores a nivel físico.
- Control del flujo de transmisión.

Ejemplos de tecnologías de esta capa son:

- Ethernet
- Token Ring
- ATM

4.3.7 Capa física

La capa física del modelo OSI comprende la infraestructura necesaria para comunicar el computador con la red, lo cual incluye el medio de transmisión (cable, fibra óptica, microondas,

etc.), como a la forma en que se transmite la información (modulación, codificación, niveles de tensión eléctrica, tasa binaria, etc).

Esta capa se encarga de transmitir los bits que conforman el mensaje a través del medio de transmisión, además que las propiedades físicas y eléctricas del medio de transmisión, los nodos de la red y los terminales de envío y recepción; ya sea que la comunicación sea bidireccional o en un solo sentido.

Se pueden resumir las funciones de esta capa en:

- Definir el medio físico por el que va a viajar la información y sus especificaciones físicas como, niveles de tensión, propiedades eléctricas, conectores y hardware involucrado, etc.
- Define como va a establecerse, mantenerse y liberarse una conexión a nivel físico.
- Transmite el flujo de bits a través del medio de transmisión.
- Especifica los cables, conectores, y componentes de la interfaz entre el computador y el medio de transmisión.
- Garantiza que haya una conexión, aunque NO garantiza que sea fiable.
- Codificar las señales electromagnéticas en tramas de bits y viceversa.

4.4 EL PROTOCOLO TCP/IP

Más que un protocolo, TCP/IP es un amplio conjunto de protocolos sobre los cuales funciona Internet. Al lector puede sorprenderle, pero TCP/IP no fue diseñado estrictamente siguiendo el modelo OSI y por lo tanto es imposible intentar de “encajarlo” para que sea acorde con dicho modelo. TCP/IP debe su nombre a los dos protocolos más importantes de los que se compone, y que fueron los primeros en desarrollarse, hablamos del protocolo IP y TCP.

De hecho, son tantos los protocolos que componen el TCP/IP que fácilmente su número puede superar la centena. TCP/IP sirve para enlazar computadoras sin importar su sistema operativo y su cobertura abarca desde las redes de área local (LAN) hasta Internet, la red mundial.

TCP/IP fue desarrollado y demostrado por el Departamento de Defensa de los Estados Unidos, ejecutándose en el predecesor de Internet: Arpanet.

De forma similar al modelo OSI, el TCP/IP está conformado por una pila de capas, en donde los niveles superiores son los más cercanos al usuario, mientras los inferiores más cercanos a la implementación física de la transmisión: las señales electromagnéticas y eléctricas transmitidas.

Mientras que TCP/IP fue diseñado y desarrollado como respuesta a un problema de ingeniería, OSI fue concebido como un modelo de índole teórica y como un primer paso en la evolución de las redes de computadores. Del último hecho se deriva que OSI es más fácil de entender, pero es TCP/IP el que se usa.

4.4.1 Capa física

Como se enunció en la descripción de la capa física del modelo OSI, la capa física es la que está conformada por el cableado y diferente hardware que está involucrado con la transmisión de la información, en este caso, por Internet. Además en esta capa se definen las características físicas, eléctricas y materiales del medio de transmisión. Según dicho medio, una conexión se divide en:

4.4.1.1 Conexión guiada

Se dice que una conexión es guiada cuando las señales eléctricas viajan por un hilo conductor o medio de transmisión. Entre las más famosas tecnologías de conexión guiada están:

- El cable telefónico.
- La fibra óptica.
- El cable coaxial.
- El par trenzado o cable UTP.
- Las líneas de transmisión ATM
- De las anteriores tecnologías de conexión guiada es posible que el lector esté familiarizado con las primeras cuatro que son las maneras más comunes de conexión a una red de computadores. Las líneas de transmisión ATM son líneas que soportan grandes volúmenes de

información y por su altísimo costo y excelente desempeño son utilizadas por las gigantescas redes corporativas, las redes de acceso de los Proveedores de Servicio Internet (los famosos ISP o *Internet Service Provider*) y los *backbone* o redes nacionales y trasnacionales.

4.4.1.2 Conexión no guiada

Una conexión no es guiada cuando no se dispone de un hilo conductor que lleve las señales eléctricas desde un punto a otro, sino cuando el medio de transmisión es el aire mediante ondas electromagnéticas. Este es el caso de las microondas, las señales radiales y de televisión, pero en el contexto que nos ocupa las principales tecnologías de comunicación no guiada (o inalámbrica) son:

- Infrarrojos.
- Conexión por WAP.
- Conexión por Bluetooth.
- Conexión por WiFi
- Conexión por WiMax.
- No profundizaremos explicando cada una de estas tecnologías que están ahora disponibles en la mayoría de dispositivos móviles y que prometen ser el futuro del acceso a Internet.

4.4.1.3 Protocolos de última milla

Como colofón a esta descripción breve de la capa física, diremos que a nivel físico y de enlace hay protocolos que son necesarios para la transmisión de la señal desde el computador que accede a Internet hasta el proveedor de servicios de Internet. Estos protocolos son llamados de “última milla” pues se supone que es el último tramo que recorre la señal antes de ser enviada a la Internet. No se profundizará en estos protocolos, ya que escapan al objetivo de este marco teórico.

4.4.2 Capa de enlace

Como se dijo en la sección del mismo nombre para el modelo OSI, la capa de enlace es la que se encarga de llevar el mensaje desde el computador hasta la red. Esta capa define cómo serán

transmitidos los datos en el nivel físico, organizados en grupos o *tramas*, además de delimitar e interpretar dichas tramas. Este nivel es, con frecuencia subdividido en dos niveles que son:

- Control de acceso a medios MAC.
- Control de enlace lógico LLC.

Ejemplos de protocolos en este nivel son Ethernet, Wireless Ethernet, Token Ring y ATM.

4.4.3 Capa ip (interred)

- Como se dijo en la sección del modelo OSI, el nivel de red es el encargado de transportar paquetes desde el origen hasta el destino a través de la red.
- Sin embargo, TCP/IP amplía el concepto de nivel de red y crea el nivel de Interred, nivel con funciones extendidas como el encaminamiento de información entre redes de computadores en una red de redes o Internet. En el esquema TCP/IP, el protocolo IP (*Internet Protocol*) es el encargado de brindar el direccionamiento y el enrutamiento de los paquetes de datos a través de la compleja red de redes, aunque en ocasiones pasa datos a otros protocolos superiores como ICMP e IGMP. ICMP se encarga de realizar labores de control y diagnóstico de la red e IGMP es usado para dirigir tráfico en multidifusión.
- El direccionamiento en IP sea realiza con el uso de un código, llamado dirección IP, que consta de 32 bits distribuidos en cuatro octetos. Cada dirección IP dentro de una misma red es única para cada dispositivo conectado a la misma y por ello le identifica. Existen cuatro tipos de direcciones IP:
 - Direcciones clase A: Son direcciones en donde el primer octeto se designa para identificar la red y los otros para identificar los miembros de dicha red.
 - Direcciones clase B: Son direcciones en donde los dos primeros octetos identifican la red y los dos restantes identifican los miembros de dicha red.
 - Direcciones clase C: Son direcciones en donde tres octetos son utilizados para identificar la red y el último para los equipos miembros de la misma.

- Direcciones clase D y E: Direcciones de uso especial y experimental.
- El enrutamiento o encaminado de la información es implementado por IP con base en diferentes protocolos de enrutamiento como BGP, OSPF y RIP.

4.4.4 La capa de transporte TCP/UDP

Los protocolos de nivel de transporte tratan problemas como la fiabilidad de la comunicación, es decir, que todos los paquetes del mensaje lleguen a su destino y que lo hagan en el orden correcto. Además en el TCP/IP esta capa determina a que aplicación van dirigidos los datos.

Los protocolos en esta capa son:

4.4.4.1 Tcp

TCP (*Transmisión Control Protocol*) o protocolo de control de transmisión, creado en los años 1973 y 1974 por Vint Cerf y Robert Khan y es uno de los protocolos fundamentales de Internet. Este protocolo es orientado a la conexión y garantiza un flujo de bytes confiable, sin errores y en el orden correcto. TCP efectúa constantemente medidas sobre el estado de la red para evitar la congestión. Dado que TCP intenta que los paquetes lleguen en el orden correcto es un protocolo no adecuado en aplicaciones de tiempo real.

4.4.4.2 Udp

En contraste con TCP, UDP (*User datagram protocol*) o protocolo de datagramas de usuario, es un protocolo que no es orientado a la conexión. La anterior afirmación quiere decir que no realiza controles para ver si los paquetes llegaron sin pérdida, o en el orden correcto, a su destino. Lo último no implica que el protocolo sea malo, sino que no realiza tareas que son de poca relevancia en ciertas aplicaciones en donde la velocidad es fundamental. UDP es utilizado primariamente en aplicaciones de tiempo real como el *streaming*, o publicación de contenido multimedia en tiempo real.

4.4.5 La capa de aplicación

Tal y como sucede en el modelo OSI, la capa de aplicación es en donde los programas que hacen uso de Internet residen. Dichos programas son, básicamente:

- HTTP: Protocolo usado para la comunicación mediante páginas Web en la WWW.
- FTP: Protocolo de transferencia de archivos.
- SMTP: Protocolo para envío de correo electrónico.
- POP: Protocolo para recepción de correo electrónico.
- TELNET: Protocolo para acceso remoto a terminales vía Internet.
- SSH: Servicio de shell remoto cifrado.
- DNS Servicio de resolución de nombres de dominio.

Un concepto importante en esta capa es el de puerto, que es un número que indica a TCP y UDP a que aplicación van dirigidos los datos transmitidos. Cada aplicación posee uno o más puertos asociados por los que establecen comunicación.

4.5 LA ARQUITECTURA DE UN SISTEMA

Una vez que se ha examinado, de forma general, el funcionamiento básico de Internet, el lector está en capacidad de entrar un poco más “en materia”. Comenzamos por el concepto de “arquitectura de un sistema”. Llamaremos arquitectura a la vista conceptual que muestra los componentes que conforman un sistema y las relaciones entre ellos que hacen del sistema un todo (una entidad holística). La arquitectura es un modelo que sirve para ilustrar el diseño del sistema y puede ser vista bajo diferentes puntos de vista. Los empleados en este documento son:

- Despliegue: En esta vista se consideran los diferentes componentes *hardware* del sistema, las relaciones entre ellos y el papel que desempeñan dentro de la aplicación.
- Implementación: En esta vista se consideran los diferentes archivos y recursos *software* que componen el sistema, las relaciones entre ellos, su organización, y el papel que desempeñan dentro de la aplicación.
- Procesos: En esta vista se estudia la topología del sistema considerando los diferentes procesos que se ejecutan como resultado de la interacción entre el usuario y la aplicación.

- **Diseño:** En esta vista se estudian los diferentes componentes del sistema desde el punto de vista del diseño del software que lo compone, es decir, como están organizadas las diferentes clases, funciones y variables que constituyen la funcionalidad del sistema y que están implementadas en los archivos y componentes descritos en la vista de implementación.

Ahora que el lector ha sido introducido al concepto de arquitectura de un sistema enunciaremos las arquitecturas más utilizadas desde el punto de vista del despliegue y del diseño:

4.5.1 Arquitectura cliente-servidor

La arquitectura cliente-servidor es la más tradicional de las arquitecturas empleadas en el diseño de sistemas de información, la causa es que evita de forma significativa la redundancia en la información, al estar organizada y almacenada en un sistema llamado *Servidor*, el cual satisface las peticiones de los usuarios que utilizan otros sistemas llamados clientes y que se encargan de transmitir al servidor las peticiones y recibir de éste los servicios requeridos en tales peticiones. En este esquema un sistema es el servidor y muchos otros son los clientes. Los clientes se dividen en dos tipos principales:

- **Clientes delgados:** Si sólo se encargan de la presentación de los datos y no realizan labor alguna de procesamiento de la información.
- **Clientes gruesos:** Si se encargan de parte del procesamiento de la información.

Puede existir confusión al considerar los grandes servidores implementados en forma de *cluster*, en donde no hay una sola máquina que ejecuta el programa servidor sino que son varias. A pesar de ello, la arquitectura es la misma ya que un solo sistema es presentado al cliente como servidor, a pesar que tras bambalinas haya muchos sistemas realizando el trabajo.

Los sistemas en Web utilizan formas modificadas de esta arquitectura, en especial la de tres capas que será la estudiada a continuación.

4.5.1.1 Arquitectura basada en tres capas

Antes de comenzar hablando de la arquitectura de tres capas, es necesario enunciar en que consiste el diseño de aplicaciones por capas. El objetivo primordial consiste en separar la lógica del

negocio de las otras labores del sistema como la presentación, el almacenamiento, el control de errores, etc. Este esquema brinda a los desarrolladores las siguientes ventajas:

- Cada capa es independiente de las otras, luego es posible cambiar la implementación de una capa sin afectar el comportamiento de las otras.
- En caso de algún cambio o corrección hay que buscar en la clase perteneciente a la capa afectada sin tener que revisar en un código mezclado, lo que implica probablemente, revisar en todo el código de la aplicación.
- Cada capa puede ser desarrollada, con un buen esquema de diseño, en paralelo disminuyendo el tiempo de desarrollo.
- Cada capa puede ser depurada independiente de las otras.
- Es posible incrementar el rendimiento de la aplicación ejecutando cada capa en hardware separado.
- No es necesario conocer en detalle la implementación de una capa para programar otra, tan solo basta con tener una buena documentación de la API de la capa.
- El sistema es fácilmente escalable, es decir, pueden desarrollarse nuevas capas que amplíen las funciones y prestaciones de la aplicación

La comunicación entre las capas se lleva a cabo por capas especiales llamadas interfaces que se encargan de llevar los datos de una capa a otra.

En esta arquitectura no se considera un solo sistema como servidor, sino tres sistemas generales que interactúan entre sí para conformar el servidor. Es como si en la arquitectura cliente-servidor “abriéramos” la caja negra del servidor para encontrar que hay tres cajas negras más pequeñas que lo componen. A estas cajas se les llama capas y son las siguientes:

- Capa de presentación: Esta capa se encarga de mostrar la salida de resultados al usuario final y de proveer una interfaz gráfica de usuario para recoger las entradas suministradas por éste, como los formularios.

- Capa de reglas del negocio: Esta capa contiene las reglas que regulan la administración de la información dentro del sistema. El diseño de esta capa es altamente influenciado por el análisis que se ha hecho antes del desarrollo del sistema.
- Capa de datos: En esta capa se gestiona el almacenamiento de la información en forma organizada.
- El lector debe tener en cuenta que hablamos de sistemas, no de máquinas hardware, por tanto es posible que los tres sistemas se ejecuten en la misma máquina física teniendo de esta forma la arquitectura de las tres capas, o es posible también que cada uno de estos sistemas se ejecute cada uno en un computador por separado.
- Existe otra forma de ver las capas, desde un punto de vista más afín con el diseño de la aplicación, que consiste en considerar cada capa como un conjunto de clases que realizan el trabajo de dicha capa y que la comunicación entre capas se hace instanciando clases de las capas de interfaz. Este esquema es explicado con mayor detalle en la sección sobre la Arquitectura del Sistema bajo el punto de vista del diseño.

4.5.2 Arquitectura P2P (par a par)

Aunque la arquitectura P2P no está involucrada con el desarrollo de SIMUPAT, la enunciaremos como contraparte del clásico esquema cliente-servidor. En la arquitectura P2P no hay cliente ni servidor, sino que todos son clientes y todos son servidores al tiempo, es decir, cada sistema conectado posee recursos que otros desean (en este caso su rol es servidor) y desea recursos que otros tienen (en este momento es cliente). Esta arquitectura es la utilizada por las redes en donde se comparten archivos para la descarga de todos los que están conectados a la red, como el caso del programa Emule y la red E-donkey.

4.6 EL SOFTWARE

En este punto es hora de profundizar en el software que hace posible el desarrollo y funcionamiento de un sistema de información en Web. En esta sección se asume que la arquitectura de dicho sistema de información es la arquitectura de las tres capas.

4.6.1.1 Las licencias de software

Sin embargo antes de hablar de software hay que entrar en detalle sobre un punto crucial a la hora de seleccionar el software en donde habrá de desarrollarse y en donde habrá de funcionar cualquier sistema: la licencia de uso.

Una licencia de uso no es más que un acuerdo legal entre el usuario y el propietario o autor de un software que define el uso legal de éste. Una licencia posee, entre muchas otras, cláusulas sobre:

- Tiempo de uso legítimo del software, es decir, por cuanto tiempo el usuario podrá utilizar dicho software.
- Finalidad de la explotación del software, es decir, para qué fin será usado dicho software, si para uso comercial, educativo, investigación científica, etc.
- Cobertura del software, es decir, a cuantos computadores es posible instalarle copias del software con la adquisición de una única licencia de uso.
- Redistribución del software, es decir, cuantas copias del software un usuario podrá distribuir, si podrá vender su licencia, etc.
- Derechos sobre el código fuente del software, es decir, si el usuario puede modificar el código fuente de la aplicación con el objetivo de hacerle mejoras, si puede usar partes de dicho código en otro software y si puede redistribuir dicho código para que otros le hagan modificaciones.
- Soporte técnico y garantía, es decir, si los autores del software brindarán a quien lo adquiera garantía y soporte, y las condiciones en los cuales dicha garantía y soporte serán prestados.

4.6.1.2 Licencias de código cerrado

Una licencia de código cerrado es aquella donde quien adquiere la licencia no puede ver, modificar o redistribuir el código fuente de la aplicación. Este tipo de licencia es la más común en el software comercial. Las licencias de código cerrado más comunes son:

- Licencia Freeware: Esta licencia permite el uso del software, bajo ciertas condiciones de finalidad, sin costo o cargo alguno por un periodo determinado que puede ser indefinido. La

redistribución del software puede ser libre o restringida según los términos de la licencia en particular.

- Licencia Shareware: Esta licencia promueve la libre redistribución (copia) del software, pero restringe su uso a un periodo determinado de tiempo tras el cual el usuario debe comprar el software para continuar usándolo.
- Licencia de prueba (Trial): Esta licencia no promueve la libre redistribución o copia del software, es decir que es indiferente a la copia del software, y restringe al usuario a usar el sistema mientras decide si le agrada o no. Una vez expira el periodo de prueba, el usuario debe comprar el software si desea continuar usándolo.
- Licencia de demostración: En este caso el usuario puede usar por un tiempo muy largo o indefinido el software, pero éste no viene con todas sus funciones disponibles. Esta licencia no promueve la redistribución del software.
- Licencia Adware: En esta licencia el usuario puede usar el producto, sin costo o con un costo menor, siempre y cuando vea la publicidad que los autores del software colocan dentro del mismo. Esta licencia promueve la redistribución del software.
- Licencia comercial: La licencia comercial es común en productos que tienen un nombre comercial consolidado y un gran prestigio entre la comunidad. Esta licencia supone una retribución económica a los autores por el derecho a usar el producto, y no hay ninguna versión de prueba para que el usuario decida si le gusta o no. De igual forma, la redistribución de la aplicación es, pro lo general, prohibida.

4.6.1.3 Licencias de código abierto

Como contraparte a las licencias de código cerrado en donde el código fuente no está disponible para el usuario, y por tanto no se pueden hacer modificaciones al mismo, las licencias de código abierto distribuyen el código fuente del producto promoviendo que desarrolladores independientes le hagan mejoras sin pedir autorización alguna siempre y cuando respeten los términos de la

licencia. Como SIMUPAT ha sido desarrollado usando en su mayor parte software de código abierto, en esta sección se profundizará sobre las licencias más comunes en esta modalidad.

4.6.1.3.1 El software libre y la licencia pública GNU

El software libre es el software que una vez adquirido puede ser modificado, estudiado, copiado, usado y redistribuido libremente. Aunque en inglés los términos libre y gratis son generalizados en una sola palabra (*free*), el software libre *no* implica que sea gratuito, lo que implica es que el que adquiera una copia del software puede distribuirlo como desee, ya sea cobrando, alquilando o regalando copias del software adquirido.

Hay que hacer diferencia entre el software libre y el software de dominio público, éste último significa que el software hace parte del conocimiento científico de la humanidad, luego no se debe seguir ninguna licencia para su uso por débil que sea, siempre y cuando se respete el crédito y se use el producto con fines legales.

En la década de los 70, el software no se consideraba un producto comercial independiente, sino un requisito (un anexo) proveído por los grandes fabricantes de computadoras para poder usar el hardware que desarrollaban. En aquella época era común que tanto programadores como usuarios compartieran sus programas unos con otros. A finales de los 70 se cambió de forma radical esta concepción, el software se independizó de los fabricantes de hardware y al constituirse como un producto comercial nacieron las restricciones a la copia y las licencias de uso.

Queriendo volver al esquema de los 70, Richard Stallman comenzó su ambicioso proyecto de un sistema operativo libre (como contraparte al creciente DOS; al robusto UNIX y al notable Macintosh), el proyecto se llamó GNU que significa: GNU No es Unix. El proyecto GNU decidió crear una licencia que reglamentara el uso del software libre y que garantizara estas libertades:

- Libertad 0: Libertad de estudiar el programa con cualquier propósito.
- Libertad 1: Libertad de estudiar y modificar el programa, para lo cual debe distribuirse el código fuente.
- Libertad 2: Libertad para copiar el programa y distribuirlo.

- Libertad 3: Libertad para mejorar el programa y publicar las mejoras de tal forma que la comunidad se beneficie y el programa tenga una mayor calidad.

La licencia GNU/GPL impone ciertas restricciones respecto al uso del programa dentro de otro desarrollo, es decir, si un programa B tiene código proveniente de un programa A, el programa B debe ser licenciado GNU/GPL de forma obligatoria si el programa A es GNU/GPL. Esto ha dividido las licencias de código abierto en dos: las que son compatibles con GNU/GPL y las que no lo son.

4.6.1.3.2 La licencia pública LGPL

La licencia pública LGPL es como la licencia GNU/GPL pero excluye del alcance de la licencia al software que para su construcción ha necesitado la inclusión de código licenciado LGPL (En la GPL, si un producto usa código licenciado GPL, el producto resultante debe ser también GPL). Particularmente, la licencia LGPL hace hincapié en los términos “trabajo derivativo” y “trabajo que usa la librería” diciendo, en el inciso quinto, que un trabajo que simplemente use la librería o código licenciado LGPL está fuera del alcance de la licencia y por lo tanto no se aplica la obligatoriedad de licenciar el producto desarrollado con LGPL-

4.6.1.3.3 La licencia BSD

La licencia BSD (*Berkeley Software Distribution*) es una licencia de código abierto que tiene menos restricciones que la GNU/GPL y que está muy cerca de volverse una “licencia” de dominio público. La licencia BSD exige que:

- Las distribuciones de código fuente deben retener el texto de la licencia.
- Las distribuciones de archivos binarios deben incluir esta licencia en la documentación entregada con los mismos (y por ello, valga la acotación, esta licencia está citada en el presente libro ya que SIMUPAT utiliza PRADO y ADODB los cuales son componentes regulados por la licencia BSD)
- Ni los nombres de los autores, ni los nombres de los colaboradores, pueden ser incluidos para promocionar productos derivados del software sin su permiso.

4.6.1.3.4 La licencia MPL

La licencia MPL, que posibilitó la liberación de Netscape Navigator 4.0 y dio origen a un proyecto importantísimo en la industria del software libre: el navegador Mozilla. La licencia MPL incorpora los mismo términos que la licencia GPL, sin ser tan permisiva como la BSD, pero elimina la restricción que obliga a usar la misma licencia GNU/GPL si se ha utilizado código GNU/GPL.

4.6.2 El sistema operativo

Sin profundizar demasiado en el tema, un sistema operativo es un conjunto de programas que interactúan entre el hardware del computador y el software de aplicación construido para trabajar en dicho computador. El más popular es Windows, marca registrada de Microsoft Corporation, que es el sistema operativo más común en los hogares y equipos de oficina. Existe también el Macintosh (Mac OS) que fue desarrollado por la empresa Apple para sus computadores PowerPC, UNIX especializado en servidores y GNU/LINUX el sistema operativo lanzado bajo la licencia GNU/GPL.

4.6.3 El servidor Web

El servidor Web es un programa cuya función es brindar comunicación con clientes a través de HTTP y que por lo tanto es el encargado de servir los documentos Web de un sitio al cliente que lo navega. Aunque no es obligatorio, el servidor Web se comunica con el cliente usando el puerto 80 (en ocasiones el 8080) y se encarga de procesar las URL para localizar el documento que se quiere acceder en los archivos físicos dispuestos en el servidor. Los servidores web más comunes en el mercado son Apache, de la fundación Apache (software libre) e IIS (*Internet Information Server*) de Microsoft Corporation.

4.6.4 La base de datos

La base de datos es, por decirlo de alguna manera, la memoria a largo plazo de un sistema de información. Una base de datos, conceptualmente hablando, es una colección de datos sistemáticamente organizados para su uso posterior. Un ejemplo muy utilizado en la literatura es la biblioteca, en donde la colección de libros está debidamente organizada para su uso por parte de los lectores.

Aunque una base de datos puede existir en formato impreso, como las fichas bibliográficas, el desarrollo tecnológico ha hecho que las bases de datos en formato electrónico sean las más utilizadas, a tal punto que el término es frecuente relacionado con el área de la Ingeniería de Sistemas y en ese contexto se usará de ahora en adelante en esta sección.

Las bases de datos tuvieron sus inicios en las colecciones de archivos secuenciales, en discos o cintas magnéticas, los cuales conformaban los datos de una aplicación. Los programadores de dichas aplicaciones daban su propio formato a tales archivos y por lo tanto una base de datos no podía ser usada varias aplicaciones diferentes, de igual forma, los programadores debían implementar los mecanismos de consulta en la base datos, las rutinas de ordenamiento, copia de seguridad, presentación de los datos, etc. Los objetivos principales de cualquier base de datos son:

- Reducir la redundancia, es decir, el tamaño que ocupan los datos físicamente en los medios secundarios de almacenamiento.
- Compartir los datos, es decir, que los mismos datos puedan ser usados por varios usuarios en distintas terminales de cómputo.
- Dar seguridad y privacidad, es decir, que los datos sean accedidos únicamente por las personas autorizadas para ello.
- Brindar integridad, es decir, que los datos mantengan coherencia entre si. Esta coherencia implica que no hayan registros incompletos, con datos fuera de rangos válidos, con datos en formato no válido, etc.

Un concepto de extrema importancia es el de integridad referencial. Este concepto significa que las relaciones entre los datos deben mantenerse, por ejemplo: si un dato X se relaciona con otro Y, si se modifica X, Y debe actualizarse o prohibirse la actualización. Lo mismo aplica para la eliminación, si un dato X se relaciona con uno Y, al borrarse X debe borrarse Y o prohibir la eliminación de X.

Según el libro Bases de Datos Relacionales, publicado por el profesor José Cárcamo Sepúlveda y editado por la UIS, podemos observar la arquitectura de la base de datos en tres capas, o niveles de abstracción así:

- Nivel físico: Es el más cercano al almacenamiento físico de los datos, es decir, a la forma en la cual la información binaria está organizada en los archivos que conforman la base de datos.
- “Nivel conceptual: Describe cuales son los datos y la relación existente entre ellos. Es la visión que tiene el administrador de la base de datos (en términos de campos y registros)”¹
- Nivel de visión: Describe como el usuario final ve los datos, y se refiere a un subconjunto de la base de datos.

El *modelo relacional* fue lo que impulsó de manera notable el desarrollo y aplicación de las bases de datos, reduciendo la redundancia de información y mejorando la accesibilidad a la misma. El modelo relacional es, básicamente, una representación de los datos en forma de un conjunto de tablas, cada una de las cuales está conformada por campos y registros, que se interrelacionan por columnas (campos) especiales llamados *llaves*.

El modelo relacional posibilitó la creación de mecanismos estándar que aumentaran la interoperabilidad entre las diferentes aplicaciones que usaban bases de datos. El más significativo de estos mecanismos fue la creación del SQL (*Structured Query Language*) o lenguaje estructurado de consulta que provee de un lenguaje común para que las aplicaciones interactúen con una base de datos relacional.

El SQL se concibió como un lenguaje fácil de entender y su sintaxis es muy parecida al inglés. Hay que aclarar que el SQL no es un lenguaje de programación como tal, ya que carece de estructuras de control, sino un lenguaje orientado a ejecutar operaciones sencillas relacionadas con las bases de datos. Sin embargo, en la práctica, es muy poco frecuente que el usuario final interactúe con las bases de datos usando directamente SQL, ya que a pesar de sus esfuerzos por ser sencillo

¹ CÁRCAMO Sepúlveda, José. Bases de Datos Relacionales, Publicaciones UIS, página 25.

requiere de esfuerzo para su aprendizaje, sino que lo haga con Interfaces Gráficas de usuario, dejando al programador la creación de las instrucciones SQL necesarias para llevar a cabo las tareas definidas por el usuario.

A nivel de diseño, el esquema usado con mayor frecuencia para representar la base de datos es el diagrama *entidad-relación*, que muestra las diferentes entidades (unidades de información) que conforman la base de datos y las relaciones entre ellas. Regresando al libro del profesor Cárcamo, en la página 27 nos dice que:

- Entidad: Es un objeto, real o abstracto, que existe y puede distinguirse de otros objetos. Denota una persona, lugar, cosa o evento de interés informacional. Está formada por un conjunto de atributos.
- Atributo: Detallan las entidades para asignarles identidad y descripción. Existen dos tipos de atributos: identificadores (llaves) y descriptores (información).
- Relación: Es una asociación entre varias entidades. Para cada relación se especifica: grado, conectividad, clase de membresía y atributos.

Al separarse el manejo de la base de datos de las aplicaciones que las utilizan, nacieron los DBMS (*Data Base Management System*) o sistemas manejadores de bases de datos, que son programas que interactúan con el nivel físico de las bases de datos implementando la funcionalidad del SQL para que las aplicaciones usen dichas bases de datos. Adicionalmente, los DBMS incorporan funciones de cifrado de datos, autenticación y otras políticas de seguridad encaminadas a proteger la privacidad y seguridad de las bases de datos que administran. Entre los DBMS más utilizados en el mercado están:

- MySQL
- Oracle
- PostgreSQL
- MsqI

- Informix.

4.6.5 Los lenguajes del lado del servidor

El HTML, a pesar de su gran éxito, está muy lejos de ser un lenguaje de programación completo y es insuficiente para crear sitios Web complejos y mucho menos sistemas de información. De hecho, cuando un sitio Web atraviesa las barreras de la mera presentación de contenidos e incursiona en la administración de datos y la implementación de la lógica de un negocio ya no puede ser construido usando HTML ni lenguajes que operan en la capa de presentación como JavaScript o los CSS. Por lo tanto las labores de procesamiento de la lógica del negocio y el acceso de la base de datos deben ser implementados por otras tecnologías: los lenguajes del lado del servidor.

Un lenguaje del lado del servidor es un lenguaje de programación que sirve para crear programas que se ejecutan en el servidor, pero cuyas entradas y salidas provienen y van hacia el cliente, de ahí el nombre de “del lado del servidor”.

Entre estos lenguajes hay dos grandes categorías: los lenguajes que generan código ejecutable y los que se ejecutan “al vuelo” llamados lenguajes “script” o de guión.

Los lenguajes que generan código ejecutable son aquellos en donde la compilación y la ejecución se hacen en etapas bien separadas en el intervalo temporal. El proceso de compilación se efectúa una vez por versión de la aplicación tras lo cual sólo ocurre la ejecución. El código ejecutable es muy rápido y permite realizar operaciones más rápidamente que los lenguajes de guión. Tecnologías de este tipo son el PERL y entornos de programación como Delphi, Java (aunque una aplicación Java no posee código compilado real en el sentido que se ejecute directamente sobre el sistema operativo), Visual C++, etc.

Sin embargo en la programación Web se han popularizado los otros lenguajes del lado del servidor, es decir, los lenguajes de guión. Pese a su tiempo de ejecución mayor, pues las operaciones de compilación y ejecución son casi simultáneas, los lenguajes de guión presentan las siguientes ventajas:

- Se pueden embeber dentro de código HTML de forma transparente para el cliente.
- No necesitan compilarse, luego si se hace un cambio en el código, el cambio queda funcional inmediatamente.
- Como no hay compilación, el código siempre está disponible, lo cual es muy apreciado por desarrolladores de código libre. (Esto hace que hayan muchas herramientas y tecnología disponible al desarrollador de aplicaciones Web).
- Sin embargo presentan una gran desventaja que se deriva de la primera ventaja listada: como se puede embeber en el código HTML (un componente presentacional) es frecuente que se revuelva el código de las capas de presentación, reglas del negocio, y datos.

Las tecnologías más comunes en esta categoría son:

- PHP
- ASP
- JSP

4.6.6 La API

Una API (*Application Program Interface*) o Interfaz de Programación de Aplicación, es un conjunto de clases, métodos y funciones que están listos para ser usados por otro programa. Este “programa” puede ser una capa de la aplicación u otra aplicación diferente.

El propósito de una API es maximizar la reutilización del código, de tal forma que si una tarea específica se repite una y otra vez no haya que programar el mismo código una y otra vez, sino simplemente llamar las funciones de la API que ya poseen dicho código. Otra ventaja de una API es la separación del código, en forma de capas, lo cual facilita el mantenimiento y depuración del software.

4.6.7 Los framework

Cuando una API ha adquirido cierta robustez, magnitud y sobre todo una estructura orientada a objetos, podemos decir que estamos ante la presencia de un framework. Un framework es una API

que provee una amplia gama de funciones que brindan una capa de soporte a una aplicación, de tal forma que el programador se dedique de forma exclusiva a la lógica del negocio en vez de otros aspectos como: interacción con las bases de datos, manejo de archivos, presentación, etc.

4.7 CÓDIGO DE BARRAS

Los conceptos enunciados a continuación fueron traducidos al español basándose en la información original publicada en la página Web <http://www.barcodeisland.com>, en la cual se puede encontrar información muy valiosa acerca de las diferentes simbologías de código de barras existentes.

4.7.1 Simbologías

Las simbologías son sistemas de codificación de datos diseñadas para que un lector o sistema de decodificación pueda leer y decodificar los datos del código de barras. Actualmente existen muchas técnicas para la codificación, una de ellas es la code 93 que es la utilizada por SIMUPAT.

4.7.2 Código 93

La simbología código 93 (code 93) produce un código denso que incluye un juego de 47 caracteres y dos dígitos de chequeo.



Figura 4.1. Ejemplo típico de un código de barras utilizando la simbología código 93.

4.7.3 Cálculo de los dígitos de chequeo

El código 93 siempre incluye dos caracteres de chequeo que llamaremos "C" y "K". Estos dos caracteres aparecen de forma consecutiva.

El carácter "C" se calcula utilizando el módulo 47 de la suma de los valores ponderados de cada uno de los caracteres de la cadena a codificar. Los pesos de los caracteres comienzan en 1 para el carácter que se encuentra más a la derecha, 2 para el siguiente y así sucesivamente hasta llegar a

20. Después del 20, la secuencia vuelve a iniciar en 1. Para comprender mejor se puede observar el siguiente ejemplo:

Tabla 4.1. Cálculo de los pesos para los caracteres de la cadena "TEST93".

Datos	T	E	S	T	9	3
Valores	29	14	28	29	9	3
Pesos	6	5	4	3	2	1
Valores con peso	174	70	112	87	18	3

Primero se suman todos los valores con pesos (tabla 4.1), $174 + 70 + 112 + 87 + 18 + 3 = 464$, después se calcula el módulo 47 de esta suma, 41 en este caso. El valor 41 esta asociado con el símbolo "+", así que "C" asumirá este valor.

El carácter "K" es calculado básicamente de la misma forma excepto porque los valores de los pesos van de 1 a 15. Para continuar con el ejemplo anterior se calculará el valor de "K":

Tabla 4.2. Cálculo de los pesos para los caracteres de la cadena "TEST93+".

Datos	T	E	S	T	9	3	+
Valores	29	14	28	29	9	3	41
Pesos	7	6	5	4	3	2	1
Valores con peso	203	84	140	116	27	6	41

Si se suman los valores con peso (tabla 4.2) se obtiene 617 cuyo módulo 47 es 6. El valor 6 esta asociado al carácter ASCII "6", así que "K" será codificado con este valor.

4.7.4 Codificación

Una vez se hayan calculado los caracteres de chequeo se procede a codificar la cadena de caracteres completa utilizando barras y espacios.

La codificación se realiza utilizando una tabla (tabla 4.3) que contiene los 47 caracteres y su respectivo código:

Tabla 4.3. Tabla de codificación para la simbología código 93.

Valor	carácter	Código	Valor	carácter	Código
0	0	100010100	24	O	100101100
1	1	101001000	25	P	100010110
2	2	101000100	26	Q	110110100
3	3	101000010	27	R	110110010
4	4	100101000	28	S	110101100
5	5	100100100	29	T	110100110
6	6	100100010	30	U	110010110
7	7	101010000	31	V	110011010
8	8	100010010	32	W	101101100
9	9	100001010	33	X	101100110
10	A	110101000	34	Y	100110110
11	B	110100100	35	Z	100111010
12	C	110100010	36	-	100101110
13	D	110010100	37	.	111010100
14	E	110010010	38	SPACE	111010010
15	F	110001010	39	\$	111001010
16	G	101101000	40	/	101101110

17	H	101100100	41	+	101110110
18	I	101100010	42	%	110101110
19	J	100110100	43	(\$)	100100110
20	K	100011010	44	(%)	111011010
21	L	101011000	45	(/)	111010110
22	M	101001100	46	(+)	100110010
23	N	101000110	-	* Start/Stop	101011110

Nota: los caracteres (\$), (%), (/) y (+) son caracteres especiales usados para codificar todos los 128 caracteres ASCII usando el Código 93 en su modo ASCII completo.

Gráficamente un “1” representa una barra y un “0” representa un espacio, para representar barras y espacios dobles se utiliza “11” y “00” respectivamente. Entonces por ejemplo si tenemos el número 1101 se codificaría de la siguiente forma:



Figura 4.2. Codificación grafica del número 1101.

4.7.5 Estructura del código 93

Un código de barras código 93 tiene la siguiente estructura:

1. Un carácter inicial representado por el carácter asterisco “*”.
2. Cualquier número de caracteres codificados utilizando la tabla 4.3.
3. Los caracteres de chequeo calculados como se describe en la sección 4.7.3.
4. Un carácter final representado por un asterisco.

5. Una barra de terminación es decir un "1".

Para completar el ejemplo citado en las secciones anteriores se codificará el texto TEST93, utilizando la estructura mencionada y los caracteres de chequeo previamente calculados.

1. carácter inicial "*": **101011110**.
2. carácter "T": codificado como **110100110**.
3. carácter "E": codificado como **110010010**.
4. carácter "S": codificado como **110101100**.
5. carácter "T": codificado como **110100110**.
6. carácter "9": codificado como **100001010**.
7. carácter "3": codificado como **101000010**.
8. Primer carácter de chequeo "+": codificado como **101110110**.
9. Segundo carácter de chequeo "6": codificado como **100100010**.
10. Carácter final "*": **101011110**.
11. Barra de terminación: **1**.

En la siguiente gráfica se muestra el resultado de la codificación. Se han resaltado cada una de las secciones mencionadas con franjas alternas para que puedan distinguirse.



Figura 4.2. Ejemplo de la estructura de un código de barras que utiliza la simbología código 93.

5 DESCRIPCIÓN DEL SISTEMA

En esta sección los autores describen el proyecto de grado del cual es objeto este libro (SIMUPAT) para que el lector tenga una visión global de las funciones y módulos implantados en el mismo con base en la Especificación de Requerimientos de Software contenida en este documento. En esta sección se muestran los módulos y características que posee SIMUPAT, así como el marco legal que reglamenta su uso, redistribución y demás acciones que comprometan los derechos de autor.

5.1 LA DISTRIBUCIÓN DE SIMUPAT

Antes de enunciar el marco legal que reglamenta el uso y distribución de SIMUPAT, los autores hacen hincapié en los siguientes puntos:

- Se entiende como SIMUPAT no sólo a los archivos fuente contenidos en el CD que contiene la distribución del proyecto, sino este libro y la documentación del programador, tanto impresa como en formato HTML y derivados.
- SIMUPAT hace uso del framework PRADO, **desarrollado por terceros**, para su funcionamiento, pero por posibles conflictos entre las licencias BSD y la legislación de derechos de autor, los autores han optado por **no** redistribuir dicho paquete junto a la distribución de SIMUPAT. Lo mismo aplica para el paquete ADODB.
- Aunque para la sustentación pública de este proyecto de grado, SIMUPAT sea exhibido con material fotográfico y textual correspondiente a la información anatomopatológica de los especímenes pertenecientes al Museo de Patología UIS, dicho material fotográfico y textual **no** es parte del presente trabajo de grado (SIMUPAT).

5.2 MARCO LEGAL

En esta sección se enuncian las licencias que reglamentan el uso y la distribución de el presente trabajo de grado por cualquier medio, ya sea electrónico o no, en cualquier formato conocido o por conocer.

5.2.1 Licencia de uso de SIMUPAT

Los autores de este proyecto: Omar Saúl Duarte Ballén y Álvaro Martínez Duarte haciendo uso de los derechos de autor que les corresponden y cumpliendo el acuerdo 164 de 2003 que refiere a la entrega de trabajos de grado en la Universidad Industrial de Santander estipulan las siguientes condiciones para el uso, distribución y modificación del proyecto SIMUPAT.

- La Universidad Industrial de Santander y el grupo GIIB están autorizados para divulgar, utilizar, reproducir, enajenar, alquilar, prestar, transformar y distribuir SIMUPAT en cualquier forma y bajo cualquier medio de divulgación y en cualquier formato conocido y por conocer, de acuerdo a los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, decisión Andina 351 de 1993, decreto 460 de 1995 y demás normas sobre la materia en la legislación colombiana.
- SIMUPAT es una producción original protegida por las leyes colombianas de derechos de autor por lo que la presente obra es de exclusiva autoría de Omar Saúl Duarte y Álvaro Martínez, luego queda prohibida su reproducción total o parcial, transformación, enajenación, préstamo, alquiler, instalación y uso general sin el consentimiento de los autores, de la Universidad Industrial de Santander o del Grupo de Investigación en Ingeniería Biomédica (GIIB).
- SIMUPAT se distribuye “como es”, por lo que los autores no se hacen responsables de daños o pérdidas, directos o indirectos, ocasionados por el uso, bajo cualquier modalidad, de SIMUPAT.
- El usuario, con el sólo hecho de instalar SIMUPAT y/o leer este libro, acepta los términos contenidos en esta licencia y asume cualquier responsabilidad legal por la violación de los términos de la misma.

- SIMUPAT debe utilizarse para un propósito legal y se prohíbe la divulgación de material fotográfico o textual protegido por derechos de autor utilizando cualquier copia de SIMUPAT. Quien incurra en falta a este inciso asume la responsabilidad legal por el uso indebido de dicho material fotográfico y textual además de las demandas y demás acciones legales causadas por la violación de esta licencia de uso.
- Cualquier disposición legal dictaminada por la Universidad Industrial de Santander que contradiga alguno o todos los términos de esta licencia tendrá preferencia sobre los mismos y derogará los términos que entren en conflicto con la nueva norma o legislación.

5.2.2 Licencia de uso del framework PRADO

Los autores de SIMUPAT, dando cumplimiento a los términos de licencia de uso y distribución del paquete PRADO, disponible para descarga e instalación, en la página de dicho proyecto (www.pradosoft.com, activa a la fecha de escritura de esta sección, el día 14 de octubre de 2006) reproducen la licencia BSD (*Berkeley Software Distribution*) en español y en inglés, licencia bajo la cual PRADO ha sido utilizado por los autores como soporte para la capa de Presentación de SIMUPAT.

Los autores aclaran que si bien **no distribuyen** este paquete y su trabajo no es derivado del mismo (aunque lo utilice), acatan esta licencia por el uso de dicho paquete en el proyecto SIMUPAT.

5.2.2.1 Licencia BSD de PRADO original, en inglés¹

PRADO is free software released under the terms of the following BSD license.

Copyright © 2004-2006, PradoSoft (<http://www.pradosoft.com>).

All rights reserved.

¹ Tomado de la documentación que viene en la distribución de PRADO, específicamente en el archivo `index.html` de dicha distribución. Esta información también se encuentra disponible en el sitio Web de PRADO: www.pradosoft.com.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the developer nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

5.2.2.2 Licencia BSD de PRADO en español

Los autores de SIMUPAT hacen énfasis en que la traducción al español, efectuada por los autores de este trabajo de grado, de esta licencia no es de ninguna manera oficial y se provee únicamente para que las personas no angloparlantes conozcan las condiciones de uso y distribución de PRADO.

PRADO es software libre distribuido bajo los términos de la siguiente licencia BSD.

Todos los derechos reservados © 2004-2006, PradoSoft (<http://www.pradosoft.com>).

La distribución o el uso en forma binaria o de código fuente, con o sin modificación, se permite siempre y cuando se cumplan las siguientes condiciones:

- Las redistribuciones de código fuente deben retener el aviso de derechos de autor, esta lista de condiciones, y el aviso siguiente.
- Las redistribuciones en forma binaria deben reproducir el aviso de derechos de autor, esta lista de condiciones y el siguiente aviso en la documentación y/u otros materiales proveídos con la distribución.
- Ni el nombre de los desarrolladores o colaboradores puede ser usado para promover productos derivados de este software sin su consentimiento explícito.

Este software es proveído por los dueños de los derechos de autor y colaboradores “como es” y cualquier garantía explícita o implícita, incluyendo, pero no limitándose a, las garantías implícitas de comercialización y ajuste para un propósito particular son incluidas en este aviso. En ningún caso el dueño de los derechos de autor o los colaboradores serán responsables por cualquier daño directo, indirecto, incidental, especial, colateral o de consecuencia (incluyendo, pero no limitándose a, la consecución de bienes o servicios sustitutos; pérdidas o ganancias derivadas del uso; o interrupción de negocios) aún cuando sea causado en cualquier teoría de responsabilidad, aún en un contrato, la responsabilidad estricta o el ilícito (incluyendo sabotaje u otros) que se deriven de cualquier forma, del uso de este software aún si se especificase la posibilidad de dicho daño.

5.2.3 Licencia de uso de ADODB

Los autores de SIMUPAT, dando cumplimiento a la licencia de uso y distribución del paquete ADODB, disponible para su descarga e instalación en la página principal de dicho proyecto (www.adodb.org activa a la fecha de escritura de esta sección, 14 de octubre de 2006) reproducen la licencia BSD/LGPL en español y en inglés, licencia bajo la cual ADODB ha sido utilizado por los autores como soporte a la capa de datos de SIMUPAT.

Los autores aclaran que si bien **no distribuyen** este paquete y su trabajo no es derivado del mismo (aunque lo utilice), acatan la licencia por el uso de dicho paquete en el proyecto SIMUPAT. La licencia BSD estándar ya fue colocada en este documento, por lo que no se repetirá en esta sección, y la licencia LGPL puede consultarse en la página principal de la Free Software Foundation.

5.2.3.1 Licencia BSD/LGPL de ADODB original en inglés¹

ADODB V4.90 8 June 2006 (c) 2000-2006 John Lim (jlim@natsoft.com.my). All rights reserved.

Released under both BSD license and Lesser GPL library license.

Whenever there is any discrepancy between the two licenses, the BSD license will take precedence.

This means you can use it in proprietary products.

5.2.3.2 Licencia BSD/LGPL de ADODB en español.

Los autores declaran que de ninguna manera esta traducción al español es oficial y se provee únicamente como medio para que lectores no angloparlantes conozcan los términos de la licencia de ADODB.

ADODB v4.90 8 Junio de 2006 (c) 2000-2006 John Lim (jlim@natsoft.com.my). Todos los derechos reservados.

Distribuido bajo las licencias BSD y la licencia GPL menor para librerías.

Si hay cualquier discrepancia entre las dos licencias, la licencia BSD tendrá prioridad.

Esto significa que la puede usar en productos no distribuidos bajo licencias de software libre.

5.2.4 Licencia de uso de PHP

Los autores de SIMUPAT al usar PHP como lenguaje de programación de SIMUPAT y considerando el numeral cuarto de la licencia de uso de dicho lenguaje (disponible en el archivo license.txt en la distribución de PHP) declaran que el uso del nombre PHP en este documento es de carácter informativo y de documentación mas no para promocionar de ninguna forma el presente proyecto de grado. Por tal motivo el nombre de este proyecto (SIMUPAT) no incluye el nombre PHP.

¹ Tomado de la documentación que viene con la distribución de ADODB, específicamente en el archivo readme.txt y en los fuentes de dicho paquete. Más información se encuentra disponible en la página Web de ADODB: www.adodb.org.

5.2.5 Nota sobre el uso de nombres y abreviaturas registradas, en este libro

Los autores aclaran que este libro contiene nombres y abreviaturas que pueden estar registrados. El uso de tales nombres y abreviaturas tiene un objetivo de carácter estrictamente informativo y no son usados para promocionar la venta de ningún producto en absoluto. Además los autores de SIMUPAT declaran que tales nombres y abreviaturas son propiedad de sus respectivos dueños. En el caso de nombres o abreviaturas supuestas por los autores como ejemplo dentro de cualquier sección de este libro se colocará el aviso: “este nombre es ficticio, cualquier parecido con la realidad es coincidencia”.

5.3 TECNOLOGÍA UTILIZADA EN EL PROCESO DE DESARROLLO DE SIMUPAT

Esta sección tiene por objetivo describir la tecnología utilizada para construir SIMUPAT, no solo la utilizada para el desarrollo del software como tal, sino las diferentes herramientas como lenguajes de programación, servidores Web usados en la pruebas de la fase de desarrollo, etc. Además los autores explican de forma breve las razones que tuvieron para la selección de la tecnología y comparan de forma subjetiva las alternativas a dichas tecnologías.

Por último los autores aclaran que ninguna de estas tecnologías es distribuida junto con SIMUPAT y que no necesariamente son indispensables para el funcionamiento del sistema. Para más información sobre los requerimientos de instalación de SIMUPAT refiérase a la sección 5.5.

5.3.1 ¿Por qué usar la Web?

Son muchas las personas que han hecho a los autores la misma pregunta y este es el momento para darle respuesta: SIMUPAT utiliza la Web por las siguientes razones, en concreto:

- Hacer que SIMUPAT sea accedido desde cualquier lugar al costo mas bajo posible. Como Internet se ha masificado de una forma notable en estos años, un sistema en Internet es la forma más fácil de lograrlo. Además un sistema en Web no requiere la instalación, ni desarrollo, de software cliente (basta con un navegador Web) lo cual brinda beneficios tanto para el cliente como para el desarrollador y la Web es idónea para la publicación de material multimedia como las fotografías de piezas anatomopatológicas que publica SIMUPAT.

- Hacer que SIMUPAT sea administrado desde cualquier parte con acceso a Internet, de tal forma que los usuarios administradores, residentes, profesores y técnicos puedan depurar la información anatomopatológica publicada en SIMUPAT aún desde sus hogares a través de Internet.

5.3.2 Herramientas de la fase de desarrollo

5.3.2.1 Php

Los autores han escogido como lenguaje de programación en la fase de desarrollo de SIMUPAT a PHP, un lenguaje de guión del lado del servidor que está regulado por licencia de software libre (vea la sección 5.2.4) y que ofrece, a juicio de los autores, algunas ventajas sobre otros lenguajes de programación de su tipo, ventajas que se listan a continuación:

- PHP puede ser ejecutado en los sistemas operativos de servidor más representativos como GNU/Linux en cualquiera de sus distribuciones y Windows de Microsoft Corp.
- PHP es software libre, por lo que puede ser distribuido, utilizado, y transformado libremente para ajustarse a las necesidades particulares de los desarrolladores.
- PHP es distribuido de forma gratuita por sus creadores y por numerosas compilaciones existentes en el mercado.
- PHP ha integrado gran cantidad de código desarrollado por programadores de software libre en todo el mundo, código que lo dota de un repositorio de clases y funciones (una API de hecho) muy extenso y robusto.
- PHP posee gran cantidad de código desarrollado por programadores de software libre de todo el mundo, que si bien no vienen integrados con PHP de forma nativa, vienen disponibles para su uso en cientos de librerías, clases y extensiones para dicho lenguaje.
- PHP posee controladores (drivers) de acceso a base de datos escritos de forma nativa, dentro del núcleo de PHP, luego el acceso a bases de datos, especialmente MySQL, es muy rápido.
- PHP tiene dos sintaxis que pueden usarse de forma indistinta, una sintaxis similar a la del C++ y una similar a la del Basic. Ambas sintaxis son fáciles de aprender y usar.

- PHP ha sido diseñado para ser usado con el servidor Web Apache, que es uno de los más importantes actualmente en el mercado.
- Los autores ya tenían experiencia en el uso de PHP.

Sólo se discutirán aquí las razones por las cuales las diferentes tecnologías que compiten con PHP, específicamente ASP, ASP.NET y JSP, no fueron seleccionadas. Un análisis completo de ventajas y desventajas no es el objetivo de esta sección. Las tecnologías desarrolladas por Microsoft: ASP y ASP.NET no fueron seleccionadas por:

- Tener un costo alto, al tener que pagar por la licencia de uso e instalación del entorno de desarrollo para dichos lenguajes y por el uso del lenguaje en sí; además del costo de adquisición de licencias para la instalación de los servidores y demás software necesario para el funcionamiento de los programas desarrollados con estos lenguajes.
- Aunque son independientes del sistema operativo y navegador del cliente, requieren el uso de un servidor Web proveído por Microsoft conocido como IIS (*Internet Information Server*). A su vez el uso del IIS implica la instalación, y por ende la adquisición, de un sistema operativo Windows como Windows 2003 Server, Windows NT o Windows XP Professional.
- Como el desarrollo en ASP y ASP.NET implica la adquisición de licencias de desarrollo, por lo general se debe pagar por licencias de código desarrollado por terceros que contribuyen al desarrollo del sistema como componentes, librerías y frameworks.

La tecnología JSP no fue seleccionada por:

- Los autores no tenían experiencia usando esta tecnología.

5.3.2.2 MySQL

Aunque SIMUPAT puede funcionar con diferentes manejadores de bases de datos, gracias a ADODB, los autores de este proyecto utilizan y recomiendan MySQL como manejador de bases de datos de SIMUPAT por las siguientes razones:

- MySQL es software libre por lo que puede distribuirse, usarse y transformarse libremente, de acuerdo a las necesidades particulares del sistema que se esté desarrollando.
- MySQL es distribuido gratuitamente por diversas fuentes en Internet.
- MySQL posee InnoDB que es una tecnología robusta para el manejo de integridad referencial.
- Existen, gracias a la filosofía del código libre, gran cantidad de controladores y APIs para interactuar con MySQL desde cualquier lenguaje de programación.
- Las bases de datos MySQL son fáciles de administrar, ya sea con la utilidad de código libre phpMyAdmin o con la consola de MySQL.
- Los autores tenían experiencia en la administración y construcción de bases de datos con MySQL.

5.3.2.3 Adodb

ADODB es un paquete, licenciado por las licencias BSD y LGPL de código libre, que encapsula la conexión a las diferentes bases de datos mediante objetos, tal y como lo hace el popular ADO de Microsoft Corp. Esta abstracción permite que las clases pertenecientes a la capa de Interfaz con la capa de datos de SIMUPAT sean independientes de la base de datos utilizada ya que de ello se encargan las clases de ADODB.

ADODB fue seleccionado por estar escrito en PHP, ser código libre y porque los nombres de sus clases y métodos son similares a los de ADO siendo su periodo de aprendizaje muy corto si ya se tiene experiencia en manejo de bases de datos con ADO de Microsoft.

5.3.2.4 Prado

PRADO es un framework orientado a la presentación y escrito en PHP por el grupo Pradosoft y licenciado por el acuerdo BSD de software libre. PRADO provee un esquema de manejo de eventos muy similar al de ASP.NET, además de proveer una API encapsulada en clases y controles que permite construir y manipular todos los elementos que conforman la Interfaz Gráfica de Usuario. PRADO además brinda un esquema de autorización y autenticación basado en archivos XML (la protección de las claves con algoritmos HASH es obra de los autores de

SIMUPAT, como se ve en la sección 8.4.6.1) y diferentes utilidades para el manejo de presentación de sitios Web. No es necesaria una explicación muy profunda de por qué los autores escogieron esta tecnología, el manejo de eventos fue determinante en este aspecto.

5.3.3 Herramientas para la fase de prueba

Aún cuando los objetivos del presente proyecto cubren únicamente el desarrollo, mas no la implantación, del sistema, siempre es necesario hacer pruebas y para ello el software debe implantarse, así sea parcialmente, de alguna manera. En esta sección se enuncia el software utilizado por los autores para ejecutar SIMUPAT.

5.3.3.1 El servidor Apache y Postfix

Podría decirse que el servidor Apache fue una elección obligada ya que SIMUPAT iba a ser, y de hecho está, desarrollado usando el lenguaje PHP. Apache no sólo es un servidor Web de software libre y distribución gratuita, sino que también puede ejecutarse en Windows y GNU/Linux y es mucho más fácil de configurar que el Internet Information Server de Microsoft. De hecho es suficiente instalar GNU/Linux para tener Apache con PHP y MySQL configurados para trabajar juntos, y si hay que hacer ajustes, basta con modificar un simple archivo de texto plano. En cuanto a Postfix, es un servidor de correo (tipo Sendmail) que viene con la distribución Mandriva 2006 de Linux y que sirvió para probar el mecanismo de envío automático de correo electrónico que tiene SIMUPAT como medio de avisar al usuario de la caducidad de una contraseña.

5.3.3.2 Lector de código de barras Metrologic

Para dotar a SIMUPAT de la capacidad de generar, imprimir y leer código de barras, el comité de sistemas de información creado para el desarrollo de SIMUPAT invirtió dinero en la adquisición de un lector de código de barras marca Metrologic. Los códigos de barras utilizados en SIMUPAT obedecen a la codificación 93 que emplea doble dígito de chequeo¹.

¹ Más información está disponible en www.barcodeisland.com.

5.4 DESCRIPCIÓN DEL SISTEMA POR MÓDULOS

En esta sección se muestran, de la manera más clara posible, los módulos que conforman SIMUPAT y que corresponden al logro de los objetivos específicos de este proyecto. Con el propósito de facilitar al lector la interrelación entre las diferentes fases del proceso de desarrollo, los objetivos específicos del proyecto y el resultado de dicho proceso, es decir, el módulo ya desarrollado, la descripción de cada módulo está compuesta por las siguientes subsecciones:

- **Descripción:** En este ítem se da una descripción textual del módulo.
- **Requerimientos asociados:** Aquí se listan los requerimientos, consignados en la especificación de requerimientos de software (véase la sección 7.5), respectivos a la fase de análisis del módulo descrito.
- **Funciones:** En este ítem se listan las funciones que puede realizar el usuario y que se relacionan con el módulo descrito. En esta parte se adjuntan capturas de pantalla que muestran la interfaz gráfica de estas funciones en SIMUPAT. Como dichas capturas de pantalla son muy grandes, se han recortado para adjuntarlas en este documento.

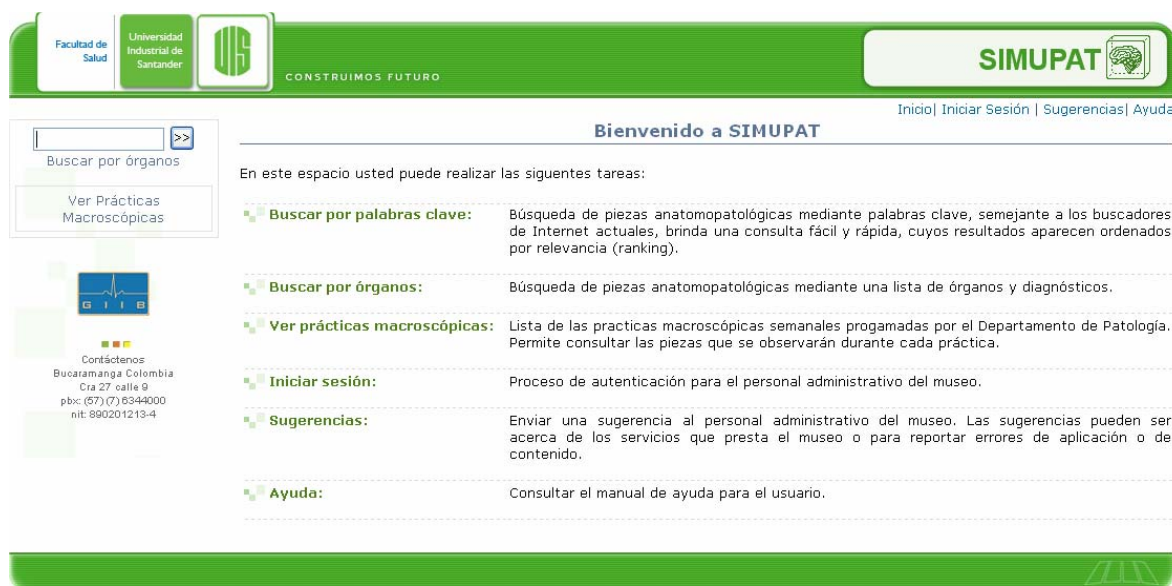


Figura 5.1. Captura de pantalla de la interfaz gráfica de usuario de SIMUPAT correspondiente a la página de bienvenida.

5.4.1 Módulo de administración de piezas

5.4.1.1 Descripción

- Este módulo es, sin duda alguna, el módulo más complejo desarrollado y corresponde al módulo que administra la información de los especímenes anatomopatológicos del Museo. Este módulo fue el primero en desarrollarse e incluyó labores como selección de la tecnología, preparación del entorno de trabajo y la definición estructural de las capas y paquetes que conforman SIMUPAT. Por estas razones fue el que más tiempo llevó terminar y el más importante del sistema.

5.4.1.2 Requerimientos asociados

Los requerimientos asociados, consignados en la Especificación de Requerimientos de Software, corresponden a los requerimientos funcionales desde el RF-01 (Crear Pieza) hasta el RF-19 (Ver Detalles de la Pieza).

5.4.1.3 Funciones

El módulo de administración de piezas permite al usuario administrador del museo (véase sección 7.5) llevar a cabo las siguientes funciones relacionadas con su rol dentro de la organización. Estas funciones son:

5.4.1.3.1 Administrar la información anatomopatológica de los especímenes del museo de patología

5.4.1.3.1.1 Crear, modificar y eliminar piezas de la colección

Esta función se refiere a que el usuario administrador dispone de controles para introducir toda la información anatomopatológica, fotográfica y textual, de una nueva pieza que desea incluir en la colección de SIMUPAT. Lo anterior aplica también para actualizar la información de una pieza determinada, caso de uso llamado Modificar Pieza, y para la eliminación de las mismas de la colección.

La interfaz gráfica de esta función fue desarrollada de tal manera que el usuario pudiera observar las fotografías que va subiendo al servidor mientras describe la pieza, de tal forma que pueda percatarse de errores mientras ingresa o actualiza los datos de dicha pieza.

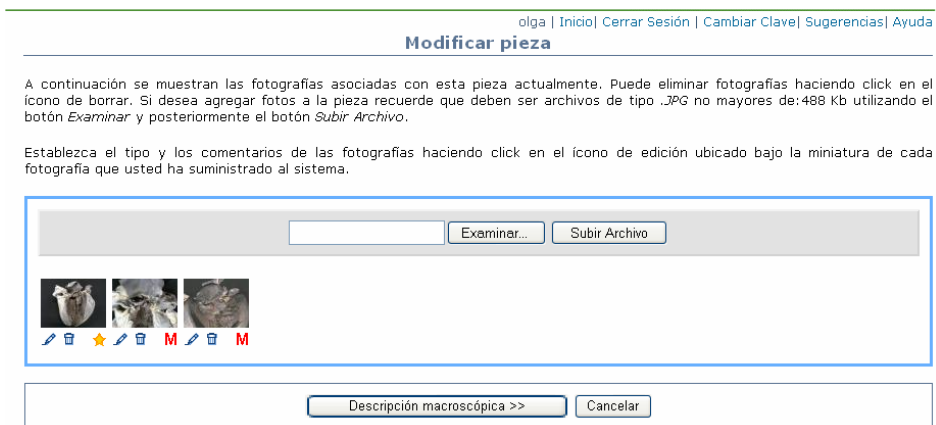


Figura 5.2. Fragmento de la captura de pantalla del control administrador de fotos de SIMUPAT.

Teniendo lo anterior en mente, SIMUPAT cuenta con varios controles especializados que actúan en el momento de la creación y modificación de piezas, diseñados con el fin de hacer el proceso lo más fácil posible al usuario. Ejemplo de estos controles, agrupados bajo el paquete *Controles* de la capa de Interfaz con Presentación (véase la sección 8.4.2) es el administrador de fotos y el de diagnósticos, mostrados en las figuras 5.2 y 5.3 respectivamente en donde se observa un usuario administrador “olga” (Se refiere a la codirectora de este proyecto, la doctora Olga Mercedes Álvarez) utilizando la interfaz gráfica de el caso de uso Modificar Pieza.

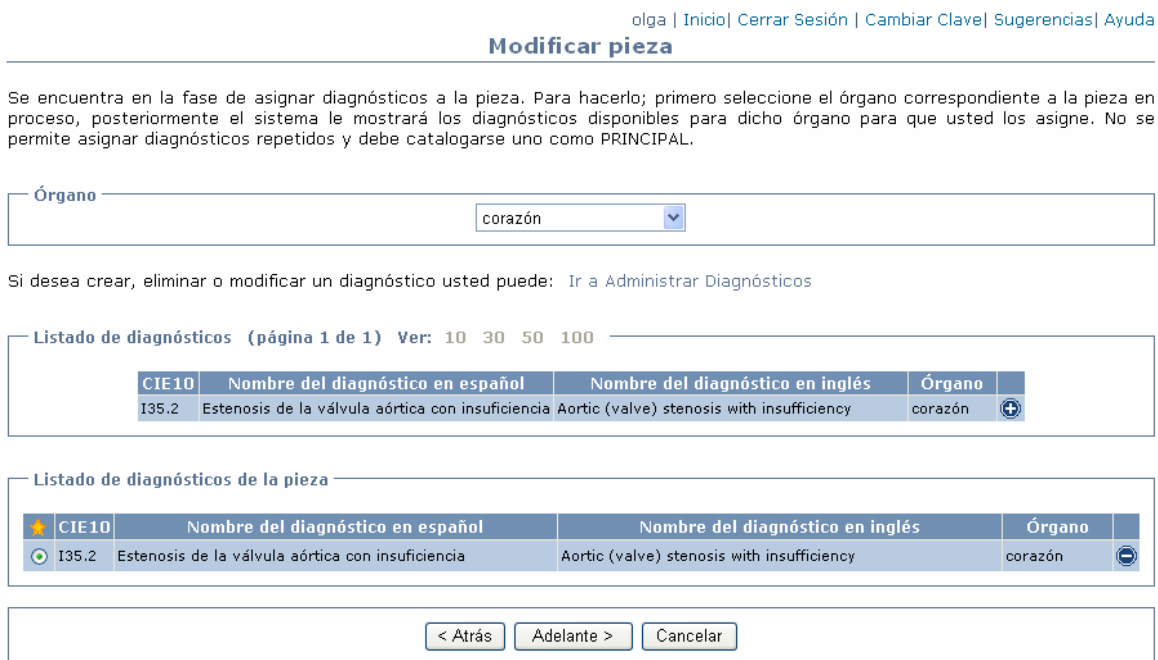


Figura 5.3. Fragmento de la captura de pantalla del administrador de diagnósticos de la pieza en SIMUPAT. Se observan el órgano y los diagnósticos de una pieza en particular para su edición.

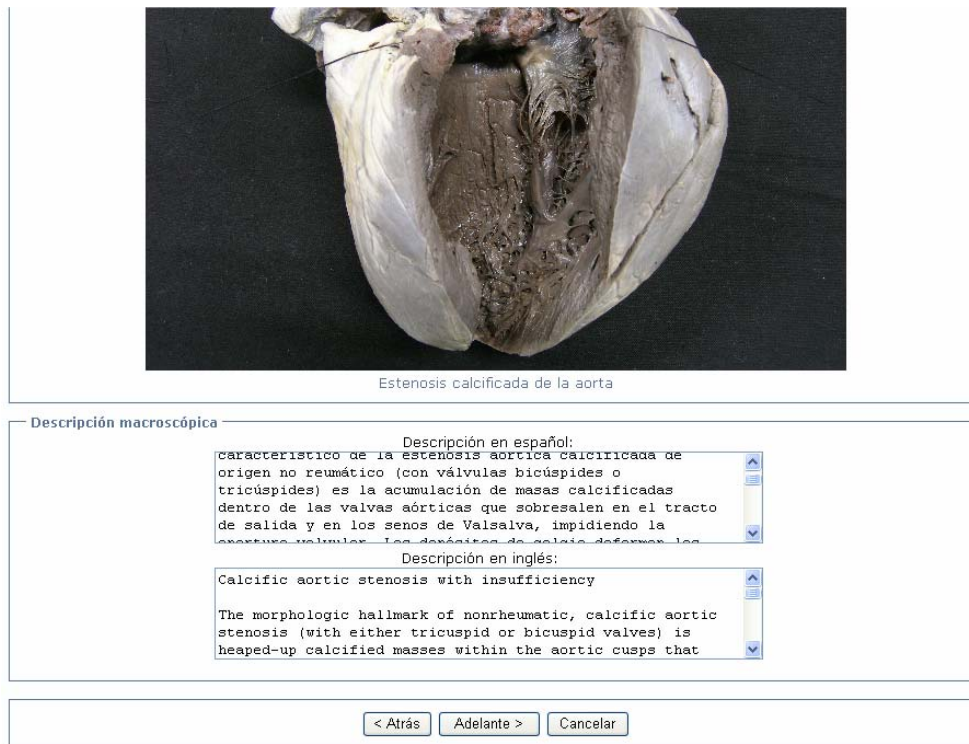


Figura 5.4. Fragmento de la captura de pantalla del paso numero 2 del proceso de modificación de una pieza, en donde se muestra al usuario la descripción macroscópica actual para que haga los cambios pertinentes.

5.4.1.3.1.2 ***Crear, modificar, eliminar y administrar la información sobre los diagnósticos presentes en la colección de especímenes del museo***

Este módulo también tiene a su cargo la administración de los diferentes diagnósticos relacionados con las piezas almacenadas en la base de datos de SIMUPAT. El caso de uso Administrar Diagnóstico es el que, a nivel de requerimientos, define esta función del sistema. De igual forma que el punto anterior, los autores de SIMUPAT han diseñado controles gráficos para estas funciones, de tal forma que se logre una interfaz gráfica agradable estéticamente y fácil de usar. La figura 5.5 muestra un fragmento de la captura de pantalla de la página que corresponde a la administración de diagnósticos. Se pueden observar los diagnósticos existentes, organizados en una rejilla que se puede ordenar alfabéticamente, en varias páginas y junto a cada diagnóstico, íconos que activan las funciones de edición y eliminación de diagnósticos, así como un botón para crear uno nuevo.

Administrar diagnósticos

A continuación se encuentra el listado de diagnósticos disponibles en este momento. Si desea ordenar esta lista alfabéticamente pulse sobre el título de la columna correspondiente al campo que desea ordenar. Si desea crear un nuevo diagnóstico pulse el botón "Nuevo diagnóstico" al final de esta lista.

Listado diagnósticos (página 1 de 2) Ver: 10 30 50 100

CIE10	Nombre del diagnóstico en español	Nombre del diagnóstico en inglés	Órgano		
K01	Amigdalitis	Amigdalitis	amígdala		
J01	apendicitis	apendicitis	apendice		
K74.6	Cirrosis hepática micronodular	Micronodular cirrhosis	hígado		
K76.1	Congestión pasiva crónica del hígado	Chronic passive congestion of liver	hígado		
N50.1	Desordenes vasculares de los órganos genitales masculinos	Vascular disorders of male genital organs	testículo		
I26	Embolismo pulmonar	Pulmonary embolism	pulmón		
I35.2	Estenosis de la válvula aórtica con insuficiencia	Aortic (valve) stenosis with insufficiency	corazón		
I63	infarto cerebral	cerebral inartction	cerebro		
C24.0	Neoplasia maligna de los conductos biliares extrahepaticos	Malignant neoplasm of the extrahepatic bile duct	vía biliar		
C78.7	Neoplasia maligna secundaria del hígado	Secondary malignant neoplasm of liver	hígado		

Figura 5.5. Fragmento de la captura de pantalla de la página de administración de diagnósticos.

CONSTRUIMOS FUTURO

olga | Inicio | Cerrar Sesión | Cambiar Clave | Sugerencias | Ayuda

Buscar por órganos

Ver Prácticas Macroscópicas

Administración de piezas

- [Crear Pieza](#)
- [Modificar Pieza](#)
- [Administrar Palabras Clave](#)
- [Administrar Diagnósticos](#)

Administración de bloques

- [Administrar bloques](#)
- [Imprimir Rótulo](#)
- [Reportes](#)

Administración de prácticas

- [Crear Práctica](#)
- [Modificar Práctica](#)
- [Agregar Piezas a la Práctica](#)

Editar diagnóstico

Información antes de la edición

CIE-10: **C24.0**

Nombre del diagnóstico en español: **Neoplasia maligna de los conductos biliares extrahepaticos**

Nombre del diagnóstico en inglés: **Malignant neoplasm of the extrahepatic bile duct**

Órgano: **vía biliar**

A continuación se encuentra el formulario que debe diligenciar para editar del diagnóstico. Recuerde que no debe dejar campos vacíos; en caso contrario, no podrá continuar.

Edición de datos

Si desea crear, eliminar o modificar un órgano usted puede: [Ir a Administrar Órganos](#)

CIE-10:

Nombre del diagnóstico en español:

Nombre del diagnóstico en inglés:

Órgano:

Figura 5.6. Captura de pantalla luego de activar la función de modificación de diagnósticos.

5.4.1.3.1.3 Crear, modificar, eliminar y administrar las palabras clave que sirven como descriptores para las piezas presentes en SIMUPAT

Este módulo también se encarga de administrar las palabras clave descriptoras de las piezas y que son utilizadas por el módulo de consulta como mecanismo para los controles de búsqueda, específicamente, la búsqueda por palabras clave al estilo de los buscadores de Internet. El sistema de consulta y administración de piezas están profundamente ligados, en especial por el caso de uso "Ver detalles de la pieza" que pertenece al presente módulo. Como se hizo en la sección anterior, se incluyen capturas de pantalla para esta sección.

Facultad de Salud | Universidad Industrial de Santander | CONSTRUIMOS FUTURO | SIMUPAT

Inicio | Inicio | Cerrar Sesión | Cambiar Clave | Sugerencias | Ayuda

Administrar palabras clave

A continuación se encuentra el listado de palabras clave disponibles en este momento. Si desea ordenar esta lista alfabéticamente pulse sobre el título de la columna correspondiente al campo que desea ordenar. Si desea crear una nueva palabra clave pulse el botón "Nueva palabra clave" al final de esta lista.

Listado de palabras clave (página 1 de 4) Ver: 10 30 50 100

Palabra clave en español	Palabra clave en inglés		
Adenocarcinoma	Adenocarcinoma		
cicatrización	scarring		
cirrosis	cirrhosis		
Conducto hepático	hepatic duct		
Congestión pasiva	Passive congestion		
Doble circulación	dual circulations		
Edema	edema		
esclerosis aórtica	aortic valve sclerosis		
estenosis aórtica	Aortic stenosis		
estenosis valvular calcifica	calcific valvular stenosis		

< Atrás | Siguiente >

Nueva palabra clave

Figura 5.7. Captura de pantalla de la página de administración de palabras clave. A diferencia de la figura 5.5, esta captura no se ha fragmentado para que se observe como encaja la interfaz gráfica de usuario con la presentación general de SIMUPAT.

5.4.2 Módulo de consulta y visualización de material

5.4.2.1 Descripción

Como su nombre lo indica, el módulo de consulta y visualización de material es el encargado de proveer al usuario la información anatomopatológica almacenada en la base de datos SIMUPAT y de proveer mecanismos para la consulta de dicha información. En cumplimiento con el objetivo

general, si bien SIMUPAT no es una herramienta didáctica en si misma pues se encamina más a la administración del museo, puede ser usada como una herramienta de consulta por los estudiantes y profesionales de la medicina en todo el mundo a través de Internet. Por ello, los autores de SIMUPAT han hecho especial hincapié en este módulo, construyéndolo de tal forma que brinde la máxima información posible, en una forma organizada y estética.

5.4.2.2 Requerimientos asociados

Los requerimientos, consignados en la Especificación de Requerimientos de Software, que dieron origen a este módulo son los comprendidos entre el RF-17 Buscar Piezas y el RF-20 Buscar por órganos.

5.4.2.3 Funciones

5.4.2.3.1 Visualización de la información anatomopatológica

Esta función, aunque corresponde al primer módulo, tiene que ver mucho con el segundo, que es el encargado de las consultas en SIMUPAT. Los autores han diseñado y construido una interfaz gráfica para la visualización de la información anatomopatológica de las piezas de tal forma que el usuario que consulta la base de datos SIMUPAT interactúe de forma agradable y estética con el sistema, facilitando la lectura y el acceso a la información. La visualización de la descripción macroscópica, fotografías, diagnósticos y palabras clave es ilustrada por la figura 5.8 que es fragmento de captura de pantalla de SIMUPAT mostrando al usuario la información de una pieza. Cabe señalar en este punto cabe señalar que gran parte de la información necesaria para diseñar la interfaz gráfica de esta sección fue recopilada en forma de ideas y sugerencias de parte de las estudiantes de medicina que integraron el comité de sistemas de información (véase sección 3.3).

General
Fotos Micro
Historia Clínica

Órgano: pulmón
 Diagnóstico principal: J15 Neumonía bacteriana
 Diagnósticos secundarios: No hay diagnósticos secundarios asociados a esta pieza

Palabras clave: Neumonía, hepatización, inflamación aguda,

Neumonía lobar: se muestra área de consolidación de todo el lóbulo inferior izquierdo en estadio de hepatización gris, el cual da una apariencia macroscópica de superficie de corte seca y color pardo grisáceo. La neumonía lobar es una infección bacteriana aguda que conlleva a una consolidación fibrinosupurativa de una porción grande de un lóbulo o de todo el lóbulo pulmonar; en la neumonía lobar se describen cuatro estadios de respuesta inflamatoria que incluyen: congestión, hepatización roja, hepatización gris y resolución.





Departamento de Patología - UIS

Pulmón con área de consolidación neumónica en el lóbulo inferior izquierdo.



Figura 5.8. Fragmento de la captura de pantalla de la interfaz gráfica de usuario de visualización de detalles de la pieza. Se pueden observar los diagnósticos asociados, el diagnóstico principal, el órgano, las miniaturas de las otras fotografías de la pieza (si el usuario hace clic, se amplían a tamaño real) y la fotografía principal con un comentario añadido.

5.4.2.3.2 Protección del material fotográfico

SIMUPAT no solamente visualiza la información fotográfica de los especímenes, sino que protege dichas fotografías colocando el mensaje “Departamento de Patología UIS”, para proteger las fotografías contra la publicación no autorizada de las mismas en otros sitios Web no autorizados por el departamento de patología UIS, dueño de dichas fotografías. Esta leyenda es colocada de forma automática por el sistema siempre que una foto es almacenada en el servidor del sistema y se pone de forma que nunca oculte la fotografía en sí. Como protección adicional, el personal del departamento encargado de la toma de fotografías, han colocado un testigo métrico con la imagen institucional de la Universidad. Si se observa atentamente, en la figura 5.8 se puede ver dicha leyenda, escrita en fondo negro con letra blanca, en la parte inferior de la fotografía.

5.4.2.3.3 Consulta de material mediante búsqueda temática ramificada

Esta función, llamada en SIMUPAT “Buscar por órganos”, permite que el usuario navegue por la colección de piezas de una forma jerárquica, como en un árbol, en donde se comienza especificando el órgano, luego el diagnóstico y por último el sistema muestra la lista de las piezas que poseen dicho diagnóstico para que posteriormente el usuario escoja cual visualizar. Los autores han diseñado la interfaz gráfica de tal manera que el cambio de una “rama a otra” en la consulta, pueda hacerse en la misma pantalla, ahorrando tiempo para el usuario. Esta función es útil si lo que se desea es dar un vistazo general por la colección de piezas en la base de datos SIMUPAT. Las figuras 5-9, 5-10 y 5-11 muestran capturas de pantalla de esta función de búsqueda ramificada o “por órganos”.

5.4.2.3.4 Consulta de material mediante palabras clave

La búsqueda temática es útil pero no suficiente si se quieren maximizar las capacidades de búsqueda y consulta de información en el sistema. La búsqueda por palabras clave, al estilo de los buscadores de Internet, es un mecanismo por el cual el usuario escribe un número de palabras que reflejan la información que desea encontrar y el sistema muestra los resultados relevantes. Este modelo de búsqueda y consulta se ha implementado en SIMUPAT con un componente adicional importante: **el soporte a lectores de código de barras**. Lo último es importante ya que la propuesta de innovación pedagógica (véase sección 3.2) citada en este libro contempla la

adecuación de un salón con computadores dotados de código de barras para que SIMUPAT visualice la información anatomopatológica del espécimen incluido en una resina polimérica (el bloque) tan pronto como el estudiante pase por el lector de código de barras el rótulo de dicho bloque, de forma similar a la consulta de precios en un supermercado de cadena.

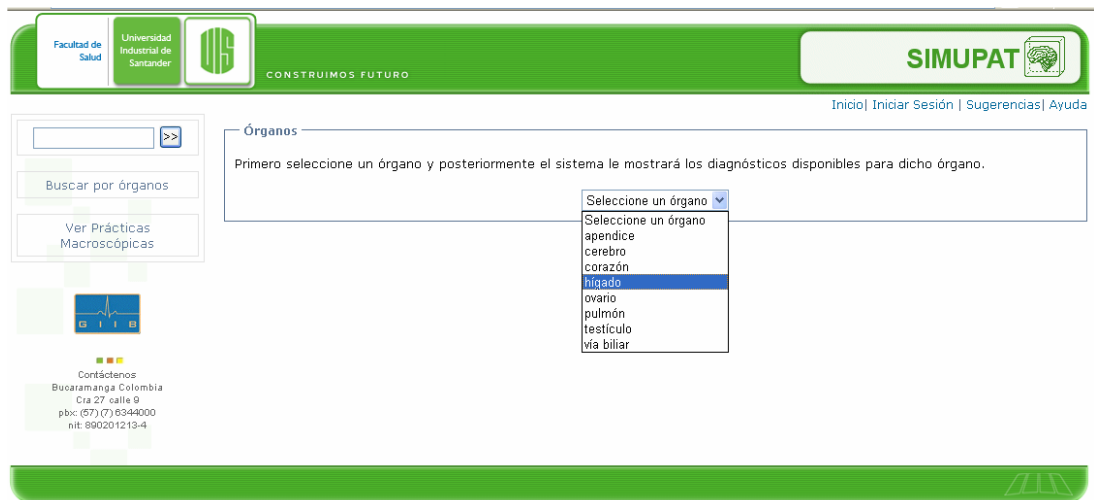


Figura 5.9. Captura de pantalla del primer paso de la búsqueda ramificada por órganos, en donde el usuario selecciona el órgano de su interés.

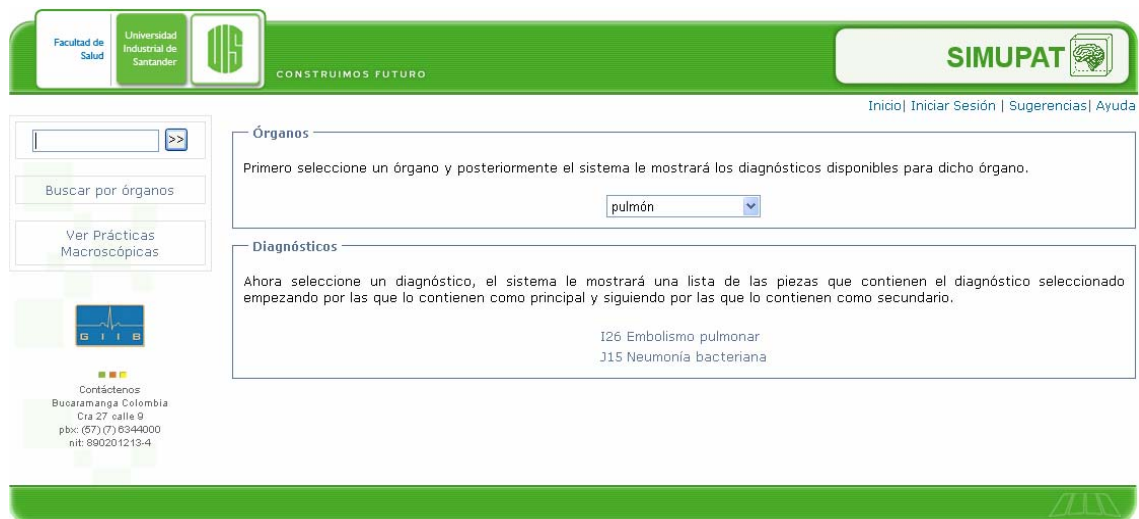


Figura 5.10. Captura de pantalla del segundo paso en la búsqueda ramificada (por diagnóstico).

Órganos

Primero seleccione un órgano y posteriormente el sistema le mostrará los diagnósticos disponibles para dicho órgano.

Diagnósticos

Ahora seleccione un diagnóstico, el sistema le mostrará una lista de las piezas que contienen el diagnóstico seleccionado empezando por las que lo contienen como principal y siguiendo por las que lo contienen como secundario.


I26 Embolismo pulmonar
 J15 Neumonía bacteriana

Piezas

Listado de Piezas

I26+01 Embolismo pulmonar

Infarto pulmonar hemorrágico en forma de cuña con base en la pleura visceral. Estos infartos son hemorrágicos porque cuando el flujo de la arteria pulmonar es obstruido por un trombo las arterias bronquiales suplen dicha área, llevando a una acumulación de grandes cantidades de hemorragia en el esponjoso parénquima pulmonar, de forma que el infarto persiste rojo. Clásicamente, el infarto pulmonar es hemorrágico y en sus primeras etapas aparece como una zona elevada rojo azulada. Con frecuencia, la superficie pleural esta cubierta por exudado fibrinoso. Los glóbulos rojos comienzan a lisarse a las 48 horas, y el infarto se vuelve más pálido y al final pardo rojizo al formarse hemosiderina. Con el paso del tiempo, los bordes del infarto comienzan a ser sustituidos por tejido fibroso, formándose una zona periférica de color blanco grisáceo que acaba convirtiendo al infarto una cicatriz retraída. [Ver detalles >>](#)



I26+02 Embolismo pulmonar

En esta pieza se observan múltiples infartos hemorrágicos que comprometen los lóbulos superior e inferior izquierdos. La causa de esta lesión es la presencia de pequeños y numerosos trombos que obstruyen la luz de arterias de pequeño calibre por este motivo vemos áreas difusas de infarto que se ubican en la periferia del órgano. Clásicamente, el infarto pulmonar es hemorrágico y en sus primeras etapas aparece como una zona elevada rojo azulada. Con frecuencia, la superficie pleural esta cubierta por exudado fibrinoso. Los glóbulos rojos comienzan a lisarse a las 48 horas, y el infarto se vuelve más pálido y al final pardo rojizo al formarse hemosiderina. Con el paso del tiempo, los bordes del infarto comienzan a ser sustituidos por tejido fibroso, formándose una zona periférica de color blanco grisáceo que acaba convirtiendo al infarto una cicatriz retraída. [Ver detalles >>](#)




Figura 5.11. Fragmento de captura de pantalla de los resultados de la búsqueda ramificada. Si el usuario da clic sobre alguno de los resultados, una pantalla como la mostrada en la figura 5.8 con la información de la pieza que desea observar.

Inicio | Iniciar Sesión | Sugerencias | Ayuda


Buscar por órganos

Piezas

Listado de Piezas

I63+01 infarto cerebral

Este infarto no hemorrágico del lóbulo parietal izquierdo muestra necrosis licuefactiva con formación de espacio quístico. Note la disminución en el tamaño del ventrículo derecho. Durante las 6 primeras horas de lesión irreversible, es poco lo que puede apreciarse microscópicamente. A las 48 hora, el tejido adquiere un aspecto pálido, blanco y tumefacto y la unión corticomedular deja de ser claramente identificable. Entre el segundo y décimo día, el cerebro adquiere una consistencia gelatinosa y friable, y el límite que hasta entonces era poco definido entre el tejido normal y el anormal se delimita con mayor claridad, a medida que se resuelve el edema del tejido adyacente que ha sobrevivido. Entre los 10 y las 3 semanas, el tejido sufre una licuefacción y finalmente es eliminado, dejando una cavidad llena de líquido y revestida por un tejido de color gris oscuro, que se expande gradualmente a medida que se va eliminando el tejido muerto. [Ver detalles >>](#)



I26+01 Embolismo pulmonar

Infarto pulmonar hemorrágico en forma de cuña con base en la pleura visceral. Estos infartos son hemorrágicos porque cuando el flujo de la arteria pulmonar es obstruido por un trombo las arterias bronquiales suplen dicha área, llevando a una acumulación de grandes cantidades de hemorragia en el esponjoso parénquima pulmonar, de forma que el infarto persiste rojo. Clásicamente, el infarto pulmonar es hemorrágico y en sus primeras etapas aparece como una zona elevada rojo azulada. Con frecuencia, la superficie pleural esta cubierta por exudado fibrinoso. Los glóbulos rojos comienzan a lisarse a las 48 horas, y el infarto se vuelve más pálido y al final pardo rojizo al formarse hemosiderina. Con el paso del tiempo, los bordes del infarto comienzan a ser sustituidos por tejido fibroso, formándose una zona periférica de color blanco grisáceo que acaba convirtiendo al infarto una cicatriz retraída. [Ver detalles >>](#)




Figura 5.12. Fragmento de captura de pantalla que muestra los resultados de búsqueda directa.

5.4.3 Módulo de mantenimiento de inventario

5.4.3.1 Descripción

El objetivo general de SIMUPAT dice: “Contribuir a la organización del museo de patología por medio de la construcción de una herramienta software basada en Web que permita administrar la colección y sirva como material didáctico de consulta”. ¿Pero como contribuir a la organización? La respuesta de los autores es la siguiente: dotando a SIMUPAT de funciones de mantenimiento de inventario. Mantener el inventario significa que el sistema debe llevar control sobre el estado de cada ítem que conforma el inventario, proveer un medio que identifique cada ítem de manera única y elaborar reportes que permitan saber a los administradores sobre el estado y el nivel del inventario en cualquier momento. SIMUPAT con la implementación del **código de barras** agiliza las labores que debe realizar el técnico encargado del museo al evitar que digite el código de cada ítem en el sistema, basta con que deslice el lector sobre el rótulo del ítem.

El lector se pregunta ¿Cuál es el ítem del que habla el párrafo anterior? Aunque el marco teórico habla sobre este aspecto (véase la sección 2.2.2) vale la pena recordar aquí que la unidad física que conforma el inventario del museo es el **bloque de resina**. Un bloque de resina no es más que un **ejemplar** de una pieza dentro de un bloque de resina polimérica y al cual se le ha asignado un rótulo con código de barras para su identificación. Una pieza, como la neumonía bacteriana mostrada en la figura 5.8 puede tener varios ejemplares en resina, ya que pueden existir en la colección varios pulmones con esas características.

SIMUPAT clasifica a los bloques según su fecha de ingreso, la fecha del último cambio de estado (llamada “de modificación”) y el estado en el que se encuentra. Dichos estados son: **guardado, prestado, averiado, extraviado, en mantenimiento y desechado**. Antes de continuar diremos que un bloque está guardado si está en las instalaciones del museo y prestado si ha sido sacado de allí para realizar una práctica, clase o por algún otro motivo que no sea descartarlo o repararlo.

5.4.3.2 Requerimientos asociados

Los requerimientos funcionales asociados, consignados en la especificación de requerimientos de software, son los numerados del RF-22 Administrar Bloques al RF-28 Generar reporte de inventario.

5.4.3.3 Funciones

5.4.3.3.1 Permitir la creación, desecho y reporte del cambio de estado de los bloques consignados bajo el inventario del museo

Esta función hace referencia a la creación de un nuevo bloque, por ejemplo si se ha adquirido otro espécimen de una pieza ya existente en el sistema, el desecho del mismo y el reporte del cambio de estado de tal bloque, por ejemplo si se descubre que se averió, si se lleva a mantenimiento o si se extravía. Todas estas funciones hacen parte de la administración de inventario, llamada “administrar bloques” en los requerimientos.

[olga](#) | [Inicio](#) | [Cerrar Sesión](#) | [Cambiar Clave](#) | [Sugerencias](#) | [Ayuda](#)

Administrar bloques

A continuación se muestra el listado de bloques asociados a la pieza A01.2+01. Si desea ordenar esta lista alfabéticamente pulse sobre el título de la columna correspondiente al campo que desea ordenar. Si desea crear un nuevo bloque pulse el botón "Crear Bloque" al final de esta lista. Si lo que desea es eliminar, modificar el estado del bloque o imprimir rótulos utilice los controles que aparecen a la derecha de cada bloque













Listado de bloques de la pieza A01.2+01			
Código	Estado	Fecha de modificación	
A01.2+01W01	Guardado	2006-10-11	  
A01.2+01W02	Guardado	2006-10-11	  
A01.2+01W03	Guardado	2006-10-11	  
A01.2+01W04	Guardado	2006-10-11	  

Figura 5.13. Fragmento de la captura de pantalla de la página de administración de bloques de SIMUPAT para la pieza cuyo código es A01.2+01.

Contemplando la posibilidad de préstamos de los bloques de resina a los estudiantes, como en una biblioteca, una vez que la propuesta de innovación pedagógica sea una realidad, los autores han elaborado una interfaz gráfica para hacer efectivo dicho “préstamo” de una forma ágil por parte del técnico y soportada, como ya se dijo, en un lector de código de barras. Existen dos interfaces para tal fin: la de despachar práctica, que se mostrará en el módulo de administración de prácticas y la de “prestar” bloque. Estas interfaces ayudan a controlar en donde se encuentra cada bloque y a detectar pérdidas o averías de forma oportuna.

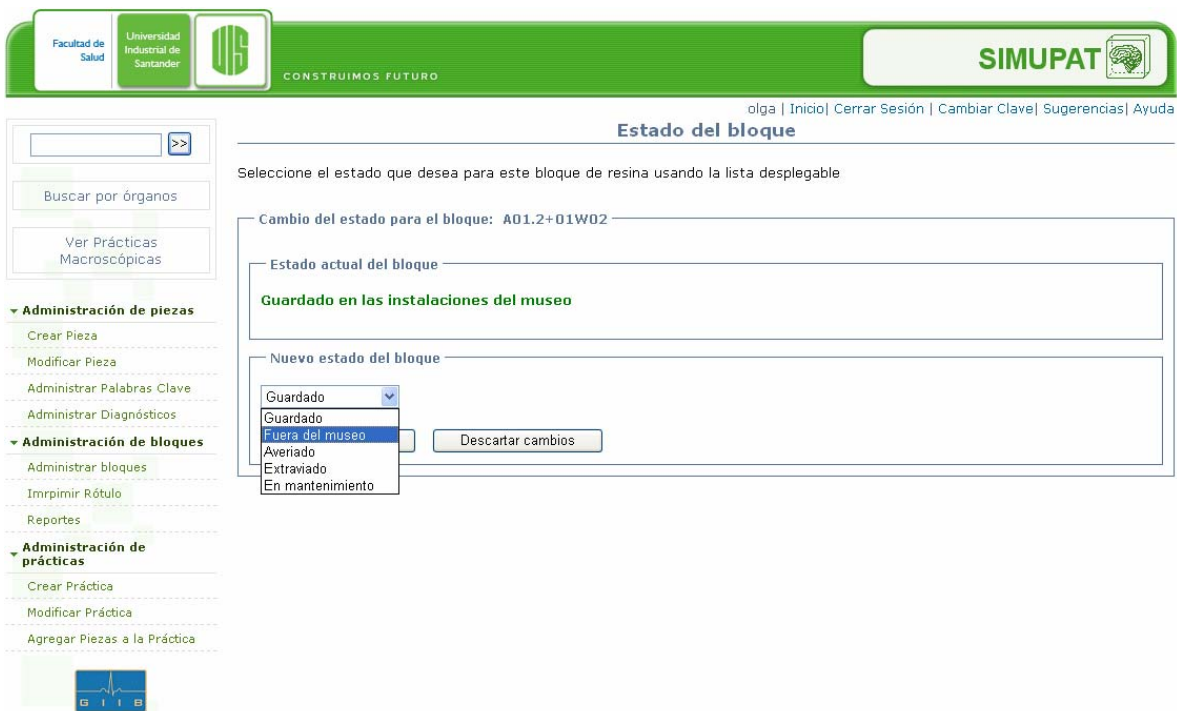


Figura 5.14. Captura de pantalla del programa de cambio de estado del bloque, en donde el usuario puede reportar un bloque como averiado, en mantenimiento, etc. La imagen muestra este caso para el bloque rotulado A01.2+01W02.



Figura 5.15. Rótulo del bloque A01.2+01W04 en código de barras codificación 93, generado por SIMUPAT.

5.4.3.3.2 Generación de reportes de inventario

Una parte importante de cualquier sistema de información es la generación de reportes. Aunque SIMUPAT tiene un generador de reportes muy sencillo, brinda suficiente información sobre el estado del inventario del museo. SIMUPAT no sólo construye reportes que informan del total de bloques en inventario, también lista y muestra la proporción de bloques en cada uno de los diferentes estados, junto con una relación completa que incluye el código de cada bloque bajo el inventario, la fecha en que ingresó, su estado actual y la fecha de la última modificación. Todos estos resultados pueden filtrarse por fecha o por estado. Esta última característica sería de utilidad,

por ejemplo, si el administrador quisiera saber cuantos bloques se han extraviado durante un mes específico.

Generador de reportes de inventario

Bienvenido al programa generador de reportes de inventario. Utilice el formulario para ajustar los parámetros del reporte que desea generar.

Generar reporte general
 Listar bloques

Filtrar por fecha de modificación

Desde: ...
Hasta: ...

Filtrar por estado del bloque

Reportar los bloques:

- Guardados en la colección del museo
- Prestados dentro de la sala
- Averiadados
- En mantenimiento
- Extraviados
- Desechados

Figura 5.16. Fragmento de la captura de pantalla de la interfaz gráfica de generación de reportes de inventario en SIMUPAT.

Adicionalmente, los resultados de estos reportes pueden verse en una versión imprimible, desprovista de color y otros elementos gráficos, para que el administrador pueda imprimir los reportes que necesite.

Si se deshabilitan las casillas de verificación se hará una lista de todos los bloques en inventario, esto es útil para una revisión de inventario por parte del administrador.

Reporte de inventario

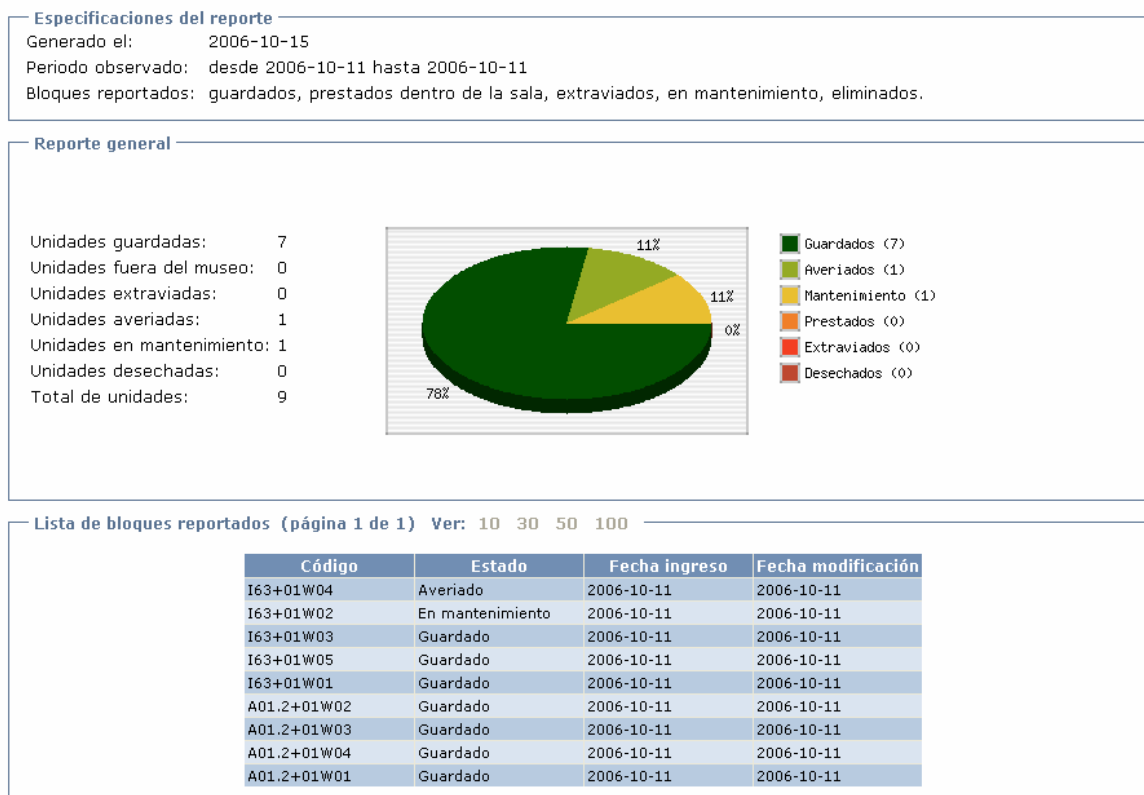


Figura 5.17. Fragmento de captura de pantalla del reporte de inventario generado por SIMUPAT.

5.4.4 Módulo de administración de prácticas

5.4.4.1 Descripción

Tal y como se dice en la sección que trata sobre la definición del problema (véase la sección 3.3), una de las funciones más importantes desempeñadas por el personal del departamento de patología, utilizando el Museo, es la de la realización de prácticas macroscópicas a los estudiantes de patología de la Facultad de Medicina de la UIS. Siendo SIMUPAT el sistema de información del museo, no podía obviar un punto tan importante como el de brindar mecanismos para facilitar la organización de las prácticas impartidas por los profesores de dicha facultad.

Las labores de SIMUPAT se orientan hacia tres actores principales, el estudiante, el profesor y el técnico. Al primero se le brinda la posibilidad de observar las diferentes prácticas definidas por el departamento, junto con la información anatomopatológica correspondiente a las piezas de dicha práctica; al segundo se le da la posibilidad de añadir piezas a la práctica que va a dictar mientras

observa en el sistema los detalles de dichas piezas; y al técnico se le dicen los bloques que debe sacar del inventario para la realización de las prácticas de modo que al inicio de la misma los bloques estén ya ubicados y a disposición de los estudiantes.

5.4.4.2 Requerimientos asociados

Los requerimientos asociados con este módulo son aquellos cuya numeración va desde el RF-30 Ver detalles de la práctica hasta el RF-33 Agregar pieza a una práctica.

5.4.4.3 Funciones

5.4.4.3.1 Administración y despacho de prácticas

Para contribuir al desarrollo las prácticas, SIMUPAT provee una pantalla de visualización de prácticas, controles para la administración de las mismas y una función especial llamada “despachar práctica” para uso del técnico con su lector de código de barras.

Práctica

Nombre: **Lesión celular**
Semana: **10**
Tipo: **Macro 1**

Piezas de la práctica

Listado de Piezas

I63+01 infarto cerebral
Este infarto no hemorrágico del lóbulo parietal izquierdo muestra necrosis licuefactiva con formación de espacio quístico. Note la disminución en el tamaño del ventrículo derecho. Durante las 6 primeras horas de lesión irreversible, es poco lo que puede apreciarse microscópicamente. A las 48 hora, el tejido adquiere un aspecto pálido, blanco y tumefacto y la unión corticomedular deja de ser claramente identificable. Entre el segundo y décimo día, el cerebro adquiere una consistencia gelatinosa y friable, y el límite que hasta entonces era poco definido entre el tejido normal y el anormal se delimita con mayor claridad, a medida que se resuelve el edema del tejido adyacente que ha sobrevivido. Entre los 10 y las 3 semanas, el tejido sufre una licuefacción y finalmente es eliminado, dejando una cavidad llena de líquido y revestida por un tejido de color gris oscuro, que se expande gradualmente a medida que se va eliminando el tejido muerto. [Ver detalles >>](#)



I26+01 Embolismo pulmonar
Infarto pulmonar hemorrágico en forma de cuña con base en la pleura visceral. Estos infartos son hemorrágicos porque cuando el flujo de la arteria pulmonar es obstruido por un trombo las arterias bronquiales suplen dicha área, llevando a una acumulación de grandes cantidades de hemorragia en el esponjoso parénquima pulmonar, de forma que el infarto persiste rojo. Clásicamente, el infarto pulmonar es hemorrágico y en sus primeras etapas aparece como una zona elevada rojo azulada. Con frecuencia, la superficie pleural esta cubierta por exudado fibrinoso. Los glóbulos rojos comienzan a lisarse a las 48 horas, y el infarto se vuelve más pálido y al final pardo rojizo al formarse hemosiderina. Con el paso del tiempo, los bordes del infarto comienzan a ser sustituidos por tejido fibroso, formándose una zona periférica de color blanco grisáceo que acaba convirtiendo al infarto una cicatriz retraída. [Ver detalles >>](#)



I26+02 Embolismo pulmonar
En esta pieza se observan múltiples infartos hemorrágicos que comprometen los lóbulos superior e inferior izquierdos. La causa de esta lesión es la presencia de pequeños y numerosos trombos que obstruyen la luz de arterias de pequeño calibre por este motivo vemos áreas difusas de infarto que se ubican en la periferia del órgano.



Figura 5.18. Fragmento de captura de pantalla de la información brindada al estudiante sobre la práctica “Lesión celular” consistente en la lista de piezas a estudiar, la semana de realización y el tipo de práctica. Si se da clic sobre alguna de estas piezas se verán los detalles de la misma, como en la figura 5.8.

Por ejemplo, en la figura 5.18 puede verse la información de la práctica “Lesión celular” que consiste en las piezas que se van a estudiar, la semana de realización de la práctica según el calendario académico del Departamento de Patología y el tipo de práctica (MACRO-1 en el ejemplo). De esta forma el estudiante puede estudiar las piezas previamente, consultando el sistema por Internet, y llegar más empapado del tema que va a estudiar.

Por otro lado, SIMUPAT provee controles fáciles de usar para que un profesor (con el beneplácito del administrador) designe piezas para su práctica, como se ve en la figura 5.19.

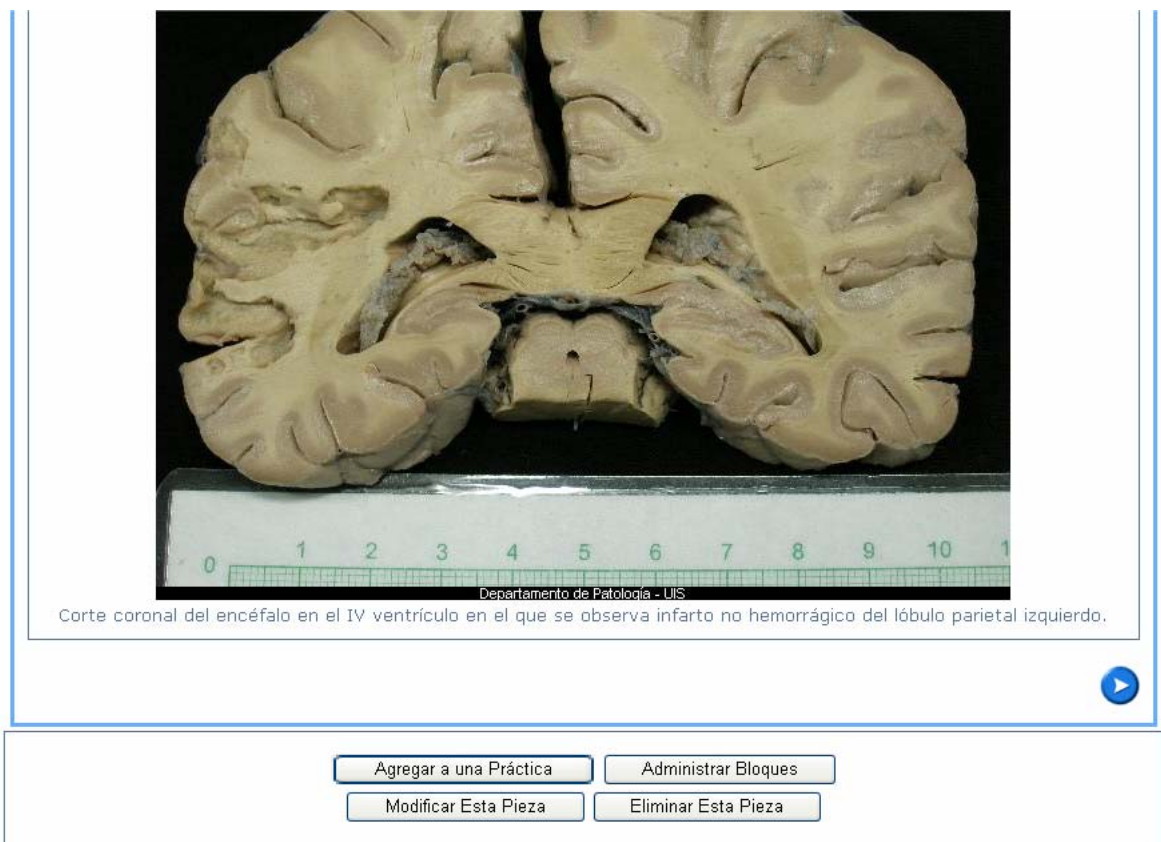


Figura 5.19. Fragmento de captura de pantalla de los detalles de un cerebro en donde se puede apreciar el botón “agregar a una práctica”.

De forma similar a la de las piezas, las prácticas poseen una pantalla de administración, creación, edición y eliminación.

En cuanto al técnico, éste ve las prácticas de un modo muy diferente. En vez de observar la lista de piezas como el estudiante o el administrador, el técnico observa los bloques ya que su trabajo es realizar las operaciones de inventario. La pantalla “despachar práctica” es una pantalla diseñada para que el técnico registre la entrada y salida de piezas hacia y desde el inventario del museo para su uso en las prácticas macroscópicas.

Modificar práctica

Se encuentra en el programa de edición de prácticas. Diligencie el formulario y puse *Aceptar* para continuar, o *Cancelar* para terminar

Nombre: Semana de realización:


Tipo de práctica

MACRO-1 MACRO-2 CORRELATO

Piezas asociadas con la práctica


Listado de Piezas

I63+01 infarto cerebral



Este infarto no hemorrágico del lóbulo parietal izquierdo muestra necrosis lícua con formación de espacio quístico. Note la disminución en el tamaño del ventrículo derecho. Durante las primeras 6 horas de lesión irreversible, es poco lo que puede apreciarse microscópicamente. A las 48 horas el tejido adquiere un aspecto pálido, blanco y tumefacto y la unión corticomedular deja de ser claramente identificable. Entre el segundo y décimo día, el cerebro adquiere una consistencia gelatinosa y friable, y el líquido que hasta entonces era poco definido entre el tejido normal y el anormal se delimita con mayor claridad, a medida que se resuelve el edema del tejido adyacente que ha sobrevivido. Entre los 10 y las 3 semanas, el tejido sufre una licuefacción y finalmente es eliminado, dejando una cavidad llena de líquido y revestida por un tejido de color gris oscuro, que se expande gradualmente a medida que se va eliminando el tejido muerto. [Ver detalles >>](#)

I26+01 Embolismo pulmonar



Infarto pulmonar hemorrágico en forma de cuña con base en la pleura visceral. Estos infartos son hemorrágicos porque cuando el flujo de la arteria pulmonar es obstruido por un trombo las arterias bronquiales suplen dicha área, llevando a una acumulación de grandes cantidades de hemorragia en el esponjoso parénquima pulmonar, de forma que el infarto persiste rojo. Clásicamente, el infarto pulmonar es hemorrágico y en sus primeras etapas aparece como una zona elevada rojo azulada. Con frecuencia, la superficie pleural esta cubierta por exudado fibrinoso. Los glóbulos rojos comienzan a lisarse a las 48 horas, y el infarto se vuelve más pálido y al final pardo rojizo al formarse hemosiderina. Con el paso del tiempo, los bordes del infarto comienzan a ser sustituidos por tejido fibroso, formándose una zona periférica de color blanco grisáceo que acaba convirtiendo al infarto una cicatriz retraída. [Ver detalles >>](#)

Figura 5.20. Fragmento de captura de pantalla de la página de modificación de prácticas.

5.4.5 Módulo de administración de usuarios, envío de sugerencias y ayuda

5.4.5.1 Descripción

Los usuarios de SIMUPAT comprenden todo tipo de roles, desde usuarios estudiantes y público en general hasta usuarios administradores del museo y el administrador del sistema. SIMUPAT provee un módulo de administración de usuarios, muy relacionado con el módulo de Seguridad, para gestionar los datos personales de cada uno de ellos, garantizar la integridad de los datos del sistema mediante mecanismos de autenticación y autorización y gestionar aspectos claves como la administración de contraseñas.

Despachar practica

A continuación se listan los bloques pertenecientes a la práctica: la cual va a ser despachada. Ubique los bloques listados y pase el lector de código de barras sobre cada uno de ellos para registrar su egreso. Cuando termine pulse *Guardar cambios* .

Si ya terminó la práctica y desea guardar los bloques de resina, pase el lector sobre cada bloque y ubíquelo en las instalaciones del museo y cuando termine pulse *Guardar Cambios* .

NOTA: El campo de texto y el botón *Procesar bloque* son de uso opcional, es decir, se usan cuando no hay lector de código de barras disponible.

Bloque en procesamiento

Lista de bloques para la práctica

Código	Estado	Código pieza	Fecha modificación
I63+01W01	AVERIADO	I63+01	2006-10-11
I63+01W02	EXTRAVIADO	I63+01	2006-10-11
I63+01W03	MANTENIMIENTO	I63+01	2006-10-11
I63+01W04	GUARDADO	I63+01	2006-10-11
I63+01W05	GUARDADO	I63+01	2006-10-11

Figura 5.21. Fragmento de la captura de pantalla de la página de “despachar práctica” en SIMUPAT.

Por otra parte en este módulo se incorporan funciones para el envío de sugerencias e inquietudes por parte de los usuarios ya sea que pertenezcan al personal del museo o que sean simplemente visitantes que consultan SIMUPAT hacia el administrador.

Por último, aunque los autores han invertido mucho esfuerzo en hacer que la interfaz gráfica del sistema sea lo más amigable posible, es natural que algunos usuarios la encuentren difícil de usar o que no sepan como utilizar algunas funciones complejas del sistema como es el caso de la creación de una pieza; esto justifica el sistema de ayuda.

5.4.5.2 Requerimientos asociados

Los requerimientos asociados para este módulo son los numerados del RF-34 Listar Usuarios hasta el RF-40 Enviar sugerencia.

5.4.5.3 Funciones

5.4.5.3.1 Administración de usuarios

La administración de usuarios, llevada a cabo por el administrador del sistema (mas no del museo) a quien llamaremos administrador de ahora en adelante dentro de la explicación de este módulo,

comprende la creación, modificación y eliminación de perfiles de usuarios autenticados ante SIMUPAT, usuarios que llevan a cabo alguna labor administrativa dentro del sistema y que necesitan de un mecanismo para autenticarse.

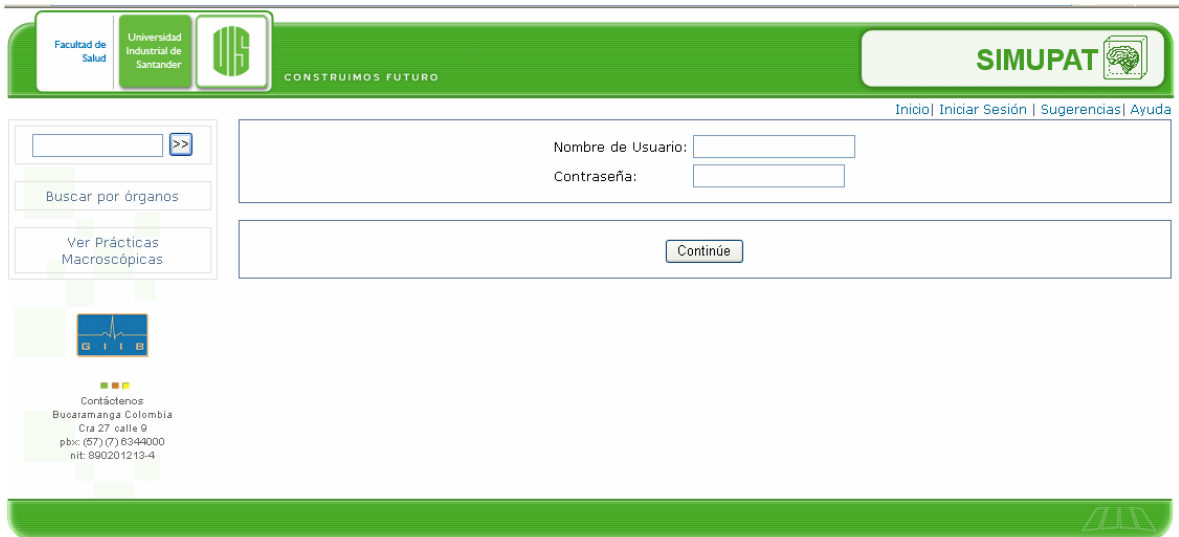


Figura 5.22. Captura de pantalla de la página de inicio de sesión.



Figura 5.23. Captura de pantalla de la página de administración de usuarios.

Aunque una explicación en detalle del proceso de autorización y autenticación en SIMUPAT se da en la sección 8.4.6.1 vale la pena resaltar aquí algunas de las características que SIMUPAT ha implementado en este aspecto. En primer lugar las contraseñas de los usuarios caducan, es decir, quedan inservibles para iniciar sesión en un tiempo determinado. Lo último contribuye a que los

usuarios estén cambiando periódicamente las contraseñas dificultando un ataque al sistema por la técnica de “ingeniería social”. En segundo lugar SIMUPAT valida las direcciones de correo electrónico de los usuarios nuevos enviando siempre un correo de confirmación.

Algo parecido sucede cuando un usuario pierde la contraseña, el administrador caduca su contraseña (previniendo que algún intruso que la posea entre al sistema) e instrucciones para la creación de la nueva contraseña son enviadas al correo electrónico del usuario. Esto es útil también si a un usuario se le quiere negar el acceso al sistema por uno u otro motivo.

5.4.5.3.2 Envío de sugerencias y el sistema de ayuda

SIMUPAT provee un esquema muy sencillo para el envío de sugerencias: la inquietud la diligencia el usuario en una caja de texto y el sistema manda dicha inquietud por correo electrónico. En cuanto al sistema de ayuda, SIMUPAT cuenta con ayudas de las funciones más complejas del sistema en forma de películas en formato *Flash* que muestran, apoyándose en globos y otras herramientas visuales, paso a paso dichas funciones que se desean explicar.

Facultad de Salud | Universidad Industrial de Santander | CONSTRUIMOS FUTURO | SIMUPAT

Inicio | Iniciar Sesión | Sugerencias | Ayuda

Enviar sugerencia

A continuación introduzca el asunto y el contenido de la sugerencia que desea enviar. El personal encargado de la administración de SIMUPAT agradecerá el reporte de cualquier anomalía o error detectado.

Asunto: Cambio de descripción macroscópica

Contenido: Comedidamente me dirijo a usted con el objetivo de reportar un error en la descripción macroscópica...

Cancelar | Enviar

Contáctenos:
Bucaramanga Colombia
Cra 27 calle 9
pbc: (57) (7) 6344000
nit: 890201213-4

Figura 5.24. Captura de pantalla del formulario de envío de inquietudes y sugerencias.

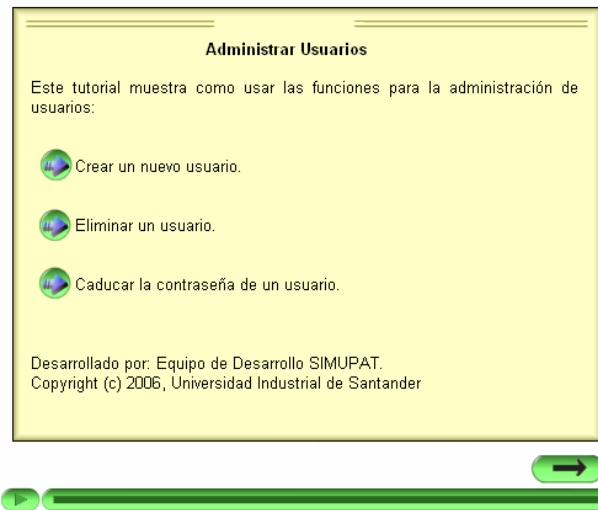


Figura 5.25. Captura de pantalla de la película Flash del tutorial de administración de usuarios.



Figura 5.26. Captura de la película Flash del tutorial de administración de usuarios, en donde se muestra un globo de ayuda.

5.4.6 El módulo de administración de software

5.4.6.1 Descripción

Todo sistema necesita de mantenimiento, y SIMUPAT no es la excepción. Por ello los autores han implementado en SIMUPAT dos casos de uso fundamentales a la hora de administrar cualquier

software: la instalación y la creación de copias de seguridad. Las funciones de este módulo se encaminan hacia estos dos aspectos primordiales.

5.4.6.2 Requerimientos asociados

Los requerimientos que tienen que ver con ese módulo son el RF-41 Instalar y RF-42 Crear copia de seguridad.

5.4.6.3 Funciones

5.4.6.3.1 Realización de copias de seguridad

Como se dice en la sección 8.4.6.2, la creación y recuperación de copias de seguridad es de extrema importancia en todo sistema de información y SIMUPAT provee al administrador del sistema una herramienta para hacerlo: la generación de un archivo de volcado escrito en lenguaje SQL.



Figura 5.27. Fragmento de captura de pantalla de la página de creación de copia de seguridad de la base de datos. El botón "Comenzar volcado" inicia este proceso.

5.4.6.3.2 Instalación del sistema

Aunque para su funcionamiento primario SIMUPAT requiere tan solo que sea descomprimido con los paquetes ADODB y PRADO, el proceso de creación y configuración de la base de datos puede ser una tarea difícil, aún contando con ayudas como el *Manual del Administrador* (véase la sección 9.3). Por ello, SIMUPAT ha automatizado en lo posible la creación y configuración de su base de datos y provee durante la fase de instalación de un mecanismo para recuperar el sistema desde un archivo de volcado SQL.

Configuración de la base de datos

SIMUPAT necesita un usuario y una base de datos para funcionar. Actualmente SIMUPAT opera con bases de datos MySQL cuya versión sea superior a 4.1.12 y que tenga INNODB. Para añadir funcionalidad con otras bases de datos vea la documentación del programador.

NOTA: El usuario especificado debe tener los siguientes permisos

- CREATE TABLE
- SELECT
- INSERT
- DELETE
- UPDATE

— Parámetros del usuario de la base de datos

Dominio (localhost si la base de datos está en el mismo servidor que SIMUPAT):

Nombre de la base de datos:

Nombre de usuario:

Contraseña de acceso:

Figura 5.28. Fragmento de captura de pantalla del proceso de configuración de la base de datos.

5.4.7 Módulo de seguridad

Más que un módulo, SIMUPAT tiene un conjunto de políticas de seguridad aplicadas a la autorización y autenticación de sus usuarios y a la administración de copias de seguridad. Una explicación a fondo de estas políticas se encuentra en la sección 8.4.6.

5.5 REQUERIMIENTOS DE INSTALACIÓN

En esta sección se enuncian los diferentes requerimientos que un cliente y un servidor deben cumplir para consultar y servir los diferentes servicios de SIMUPAT. Los autores aclaran de nuevo que **no redistribuyen** ninguno de estos componentes y que su adquisición escapa de su responsabilidad, aunque se dan las direcciones Internet de donde pueden ser descargados.

5.5.1 Requerimientos de software

SIMUPAT requiere de los siguientes componentes software y de configuración para el cliente.

- Un navegador Web, preferiblemente Internet Explorer 6.0 o Mozilla Firefox 1.5.0.5.

El servidor debe tener estos componentes instalados:

- Servidor Web con soporte para PHP 5 o superior. Se recomienda el uso de Apache versión 2.2 o superior (las versiones anteriores para Windows poseen un *bug* que ocasiona problemas con SIMUPAT, en Linux no importa)
- PHP 5 o superior, se recomienda 5.1 o superior.

- Las extensiones DOM, ICONv y Zlib para PHP (Normalmente vienen con las distribuciones de PHP 5.1 o se pueden conseguir en las diferentes distribuciones de Linux bajo formato RPM).
- PRADO¹ versión 3 o superior instalado en el mismo directorio que SIMUPAT.
- ADODB² versión 4.90 o superior instalado en el mismo directorio que SIMUPAT.
- Servidor de correo instalado compatible con PHP, se recomienda sendmail o Postfix.
- Manejador de bases de datos compatible con ADODB, se recomienda MySQL versión 4.1.12 o superior con INNODB habilitado.
- Sistema operativo con prestaciones de servidor, se recomienda GNU/Linux (los autores usaron Linux Mandriva 2006)

5.5.2 Requerimientos de hardware

Para el cliente:

- Resolución de pantalla de 800*600 (mínimo) o superior a 1024*768 (recomendado) a color verdadero.
- Conexión a Internet.
- Lector de código de barras compatible con la codificación 93.

Para el servidor, los autores se limitan a colocar las especificaciones del equipo usado como servidor de prueba durante el desarrollo.

- Conexión a Internet de alta velocidad.
- Procesador Intel Pentium IV con Hyper Threading a 3.20 GHz.
- Memoria RAM de 1 Gigabyte.
- Capacidad en disco de 160 Gigabytes.

¹ Se puede adquirir en www.pradosoft.com (activo a la fecha de escritura de esta sección 15 de Octubre de 2006)

² Se puede adquirir en www.adodb.org (activo a la fecha de escritura de esta sección)

6 METODOLOGÍA

6.1 PROCESO UNIFICADO DE RATIONAL

La metodología que se siguió durante el proyecto fue la del Proceso Unificado de Rational (RUP Rational Unified Process)¹ debido a que es un proceso con un enfoque iterativo en el cual se propone la comprensión incremental del problema a través de una serie de refinamientos sucesivos y un crecimiento incremental de una solución a través de varios ciclos. Su objetivo es permitir la producción de software de la mayor calidad que satisfaga las necesidades de los usuarios finales, dentro de planificaciones y presupuestos predecibles. Como parte del enfoque iterativo se encuentra la flexibilidad para acomodarse a nuevos requisitos o a cambios tácticos en los objetivos del negocio. También permite que el proyecto identifique y resuelva los riesgos rápidamente. En esta sección se describirá en que consiste la metodología del RUP y en las secciones 8 y 9 (Plan de trabajo y Cronograma) se expondrá como se aplicará dicha metodología al desarrollo del proyecto.

El desarrollo bajo el RUP está **centrado en la arquitectura**. El proceso se centra en establecer al principio una arquitectura software que guía el desarrollo del sistema. Con ello se facilita el desarrollo en paralelo, se minimiza la repetición de trabajos y se incrementa la probabilidad de reutilización de componentes y el mantenimiento posterior del sistema. Este diseño arquitectónico sirve como una sólida base sobre la cual se puede planificar y manejar el desarrollo de software basado en componentes.

Las actividades de desarrollo bajo el RUP están **dirigidas por los casos de uso**. El Proceso Unificado pone un gran énfasis en la construcción de sistemas basada en una amplia comprensión

¹ Rumbaugh, James; Jacobson, Ivar; Booch, Grady, El Lenguaje Unificado de Modelado, Addison Wesley, Madrid, 1999.

de cómo se utilizará el sistema que se entregue. Las nociones de los casos de uso y los escenarios se utilizan para guiar el flujo de procesos desde la captura de los requisitos hasta las pruebas, y para proporcionar caminos que se pueden reproducir durante el desarrollo del sistema.

El Proceso Unificado de Rational es un **proceso configurable**. Aunque un único proceso no es adecuado para todas las organizaciones de desarrollo de software, el Proceso Unificado es adaptable y puede configurarse para cubrir las necesidades de proyectos que van desde pequeños equipos de desarrollo de software hasta grandes empresas de desarrollo. También se basa en una arquitectura de proceso simple y clara, que proporciona un marco común a toda una familia de procesos y que, además, puede variarse para acomodarse a distintas situaciones. Dentro del propio Proceso Unificado se encuentran las guías sobre cómo configurar el proceso para adaptarse a las necesidades de una organización.

El Proceso Unificado soporta las **técnicas orientadas a objetos**. Cada modelo es orientado a objetos. Los modelos del Proceso Unificado se basan en los conceptos de objeto y clase y las relaciones entre ellos, y utilizan UML como la notación común.

El Proceso Unificado impulsa un **control de calidad** y una **gestión del riesgo**, objetiva y continua. La evaluación de la calidad va contenida en el proceso, en todas las actividades, e implicando a todos los participantes, mediante medidas y criterios objetivos. No se trata como algo a posteriori o una actividad separada. La gestión del riesgo va contenida en el proceso, de manera que los riesgos para el éxito del proyecto se identifican y se acometen al principio del proceso de desarrollo, cuando todavía hay tiempo de reaccionar.

6.1.1 Fases del RUP

El RUP posee una estructura bidimensional. En su eje horizontal se presentan los aspectos referentes al ciclo de vida del proceso y en el eje vertical las características referidas a los flujos de trabajo del proceso. También en los ejes horizontal y vertical se representan respectivamente los aspectos dinámicos y estáticos del proceso, estableciéndose de esta forma los mecanismos necesarios para asegurar que el seguimiento de un proyecto informático se pueda hacer desde diferentes perspectivas, ver figura 6.1. El proceso de desarrollo en el RUP comprende **Ciclos o**

Iteraciones, en los cuales se logra siempre un producto final ejecutable. El proceso se divide en cuatro fases: Inicio, Elaboración, Construcción, Transición. Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

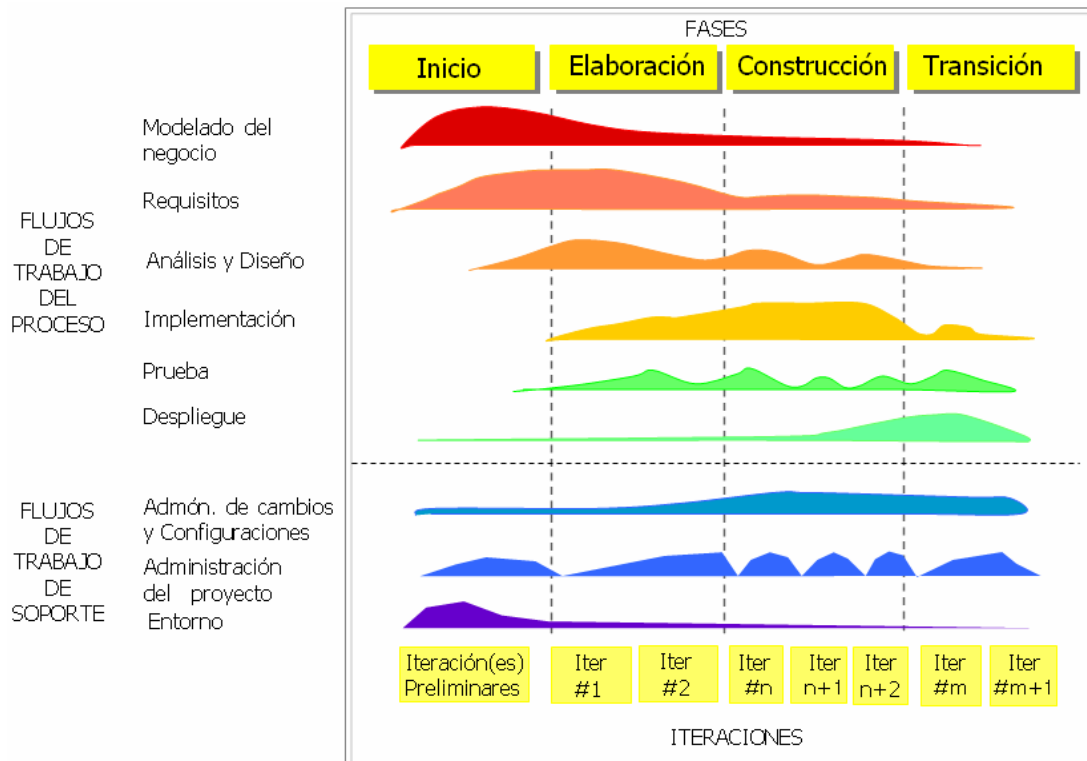


Figura 6.1. El ciclo de vida del desarrollo de software.

Inicio: Durante la fase de inicio, se establece la planificación del proyecto y se delimita su alcance. La planificación del proyecto incluye los criterios de éxito, la evaluación del riesgo, estimaciones de recursos que se necesitarán y un plan de fases que muestre la planificación de los hitos principales. Durante la iniciación, es frecuente crear un prototipo ejecutable que sirva para probar los conceptos. Al final de la fase de iniciación se examinan los objetivos del ciclo de vida del proyecto y se decide si proceder con el desarrollo del sistema.

Elaboración: Los objetivos de la fase de elaboración son analizar el dominio del problema, establecer una base arquitectónica sólida, desarrollar el plan del proyecto y eliminar los elementos de más alto riesgo del proyecto. Las decisiones arquitectónicas deben tomarse con una comprensión del sistema global. Esto implica que se deben describir la mayoría de los requisitos

del sistema. Para verificar la arquitectura, se implementa un sistema que demuestre las distintas posibilidades de la arquitectura y ejecute los casos de uso significativos. Al final de la fase de elaboración se examinan el alcance y los objetivos del sistema, la elección de la arquitectura y la resolución de los riesgos más grandes, y se decide si se debe pasar a la construcción.

Construcción: Durante la fase de construcción, se desarrolla de forma iterativa e incremental un producto completo que está preparado para la transición hacia la comunidad de usuarios. Esto implica describir los requisitos restantes y los criterios de aceptación, refinando el diseño y completando la implementación y las pruebas del software. Al final de la fase de construcción se decide si el software, los lugares donde se instalará y los usuarios están todos preparados para empezar a funcionar.

Transición: Durante la fase de transición, el software se despliega en la comunidad de usuarios. Una vez que el sistema ha sido puesto en manos de los usuarios finales, a menudo aparecen cuestiones que requieren un desarrollo adicional para ajustar el sistema, corregir algunos problemas no detectados o finalizar algunas características que habían sido pospuestas. Esta fase comienza normalmente con una versión beta del sistema, que luego será reemplazada con el sistema de producción. Al final de la fase de transición se decide si se han satisfecho los objetivos del ciclo de vida del proyecto, y se determina si se debería empezar otro ciclo de desarrollo. Este es también un punto en el que se asimilan las lecciones aprendidas en el proyecto para mejorar el proceso de desarrollo, que será aplicado al próximo proyecto.

Flujos de Trabajo¹

La otra dimensión que propone el RUP es la de los flujos de trabajo, en si un flujo de trabajo representa una secuencia de actividades que produce un resultado valioso. Los flujos de trabajo

¹ BOLAÑOS CASTRO, Sandro Javier. Análisis Y Diseño del Portal de Conocimiento de la Universidad Distrital Francisco José de Caldas. Bogotá 2004.

propuestos por el RUP son: requerimientos, análisis y diseño, implementación, prueba, despliegue, administración de cambios y configuraciones, administración del proyecto y entorno.

Requerimientos: en este flujo de trabajo se responde al Qué del sistema, es decir a que debe responder el sistema. La mejor manera de representar los requerimientos es a través de los diagramas de casos de uso.

Análisis y Diseño: En este flujo de trabajo se responde al Cómo del sistema, es decir la manera como se implementará el sistema. El diseño debe proveer al producto de la manera de ejecutar las tareas y funciones descritas en los casos de uso para satisfacer todos los requerimientos y ser flexible al cambio. El modelo RUP consta de clases estructuradas en paquetes, subsistemas e interfaces definidas y la forma de colaboración entre las clases.

Implementación: En este flujo de trabajo se define la organización del código, las clases y objetos en forma de componentes. Prueba los componentes desarrollados y los integra en un sistema ejecutable.

Prueba: En este flujo de trabajo se verifica la interacción entre objetos, la integración de componentes, la satisfacción de requerimientos. Identifica los errores y los corrige antes de la instalación. El RUP describe cómo planear y ejecutar estas pruebas y además, propone probar los componentes desde el principio para ganar confiabilidad, funcionalidad y desenvolvimiento.

Despliegue: En este flujo de trabajo se hace llegar el producto a los usuarios finales durante la Transición.

Administración del proyecto: En este flujo de trabajo se propone lograr un equilibrio de objetivos contrarios, manejar riesgos y producir software que satisfaga a clientes y usuarios. RUP incluye unos marcos para el manejo de proyectos, guías para la planificación y personal, ejecución y monitoreo de planes y de riesgos.

Administración de configuración y cambios: En este flujo de trabajo se controlan los artefactos producidos, actualizaciones simultáneas, múltiples versiones. El RUP provee de guías para desarrollos en paralelo, para automatizar la construcción y administrar defectos.

Entorno: En este flujo de trabajo se propone el medio y las herramientas de desarrollo que harán posible el proyecto. El RUP apoya la creación de un ambiente apropiado para la escala del proyecto.

6.1.2 Casos de uso

Como ya se mencionó en la sección 6.1, las actividades de desarrollo bajo esta metodología están dirigidas por casos de uso. Pero ¿en qué consiste exactamente un caso de uso?

Un caso de uso especifica el comportamiento de un sistema o de una parte del mismo, y es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor.

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo. Los casos de uso proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema. Además, los casos de uso ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo.¹

6.1.3 Diagrama de casos de uso

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. En la sección 7.5 se muestran los Diagramas de casos de uso para el sistema SIMUPAT.

• ¹ RUMBAUGH, James; JACOBSON, Ivar y BOOCH, Grady. El Lenguaje Unificado de Modelado, Addison Wesley, Madrid, 1999.

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

Actores: Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema. Se representa mediante una figura humana dibujada con líneas. Esta representación sirve tanto para actores que son personas como para otro tipo de actores.

Casos de Uso: Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

Relaciones entre casos de uso: Un caso de uso, en principio, debería describir una tarea que tiene un sentido completo para el usuario. Sin embargo, hay ocasiones en las que es útil describir una interacción con un alcance menor como caso de uso. La razón para utilizar estos casos de uso no completos en algunos casos, es mejorar la comunicación con el equipo de desarrollo. Para el caso de que queramos utilizar estos casos de uso más pequeños, las relaciones entre estos y los casos de uso ordinarios pueden ser de los siguientes tres tipos:

- **Inclusión (<<include>>):** Un caso de uso base incorpora explícitamente a otro caso de uso en un lugar especificado en dicho caso base. Se suele utilizar para encapsular un comportamiento parcial común a varios casos de uso. En la figura 7.2 de la sección 7.5.1 se muestra cómo el caso de uso Buscar Pieza puede incluir el comportamiento del caso de uso Listar Piezas.
- **Extensión (<<extend>>):** Cuando un caso de uso base tiene ciertos puntos (puntos de extensión) en los cuales, dependiendo de ciertos criterios, se va a realizar una interacción adicional. El caso de uso que extiende describe un comportamiento opcional del sistema (a diferencia de la relación incluye que se da siempre que se realiza la interacción descrita). En la

figura 7.2 de la sección 7.5.1 se muestra como el caso de uso Listar Pieza permite adicionalmente realizar el caso de uso Ver Detalles de la Pieza.

Ambos tipos de relación se representan como una flecha con línea discontinua etiquetada con el estereotipo correspondiente (<<include>> o <<extend>>), de tal forma que la flecha indique el sentido en el que debe leerse la etiqueta, teniendo en cuenta que dicha flecha representa una relación de dependencia.

- **Generalización:** Cuando un caso de uso definido de forma abstracta se particulariza por medio de otro caso de uso más específico. Se representa por una línea continua entre los dos casos de uso, con el triángulo que simboliza generalización pegado al extremo del caso de uso más general. El caso de uso hijo hereda las asociaciones y características del caso de uso padre. El caso de uso padre se trata de un caso de uso abstracto, que no está definido completamente.

6.2 EL PLAN DE TRABAJO

El plan del proyecto se estructuró de acuerdo con las 3 primeras fases del Proceso Unificado de Rational: Inicio, Elaboración y Construcción, cada una de ellas con sus respectivos flujos de trabajo. La fase de Transición no se contempló pues depende en gran medida del proceso de reestructuración del museo.

Cada actividad del Proceso Unificado de Rational lleva algunos artefactos asociados, bien sean requeridos como entradas, bien sean generados como salidas. Los modelos son el tipo de artefacto más importante del RUP. Un modelo es una simplificación de la realidad, creada para comprender mejor el sistema. Los siguientes son algunos de los modelos mas utilizados por el RUP.

- Modelo del negocio: Establece una abstracción de la organización.
- Modelo de casos de uso: Establece los requisitos funcionales del sistema.
- Modelo de diseño: Establece el vocabulario del problema y su solución.

- Modelo de despliegue: Establece la topología del hardware sobre la cual se ejecutará el sistema.
- Modelo de implementación: Establece las partes que se utilizarán para ensamblar y hacer disponible el sistema físico.

Una vista es una proyección de un modelo. En el RUP, la arquitectura de un sistema se captura en forma de vistas que interactúan entre si: la vista de diseño, la vista de procesos la vista de despliegue, la vista de implementación y la vista de casos de uso.

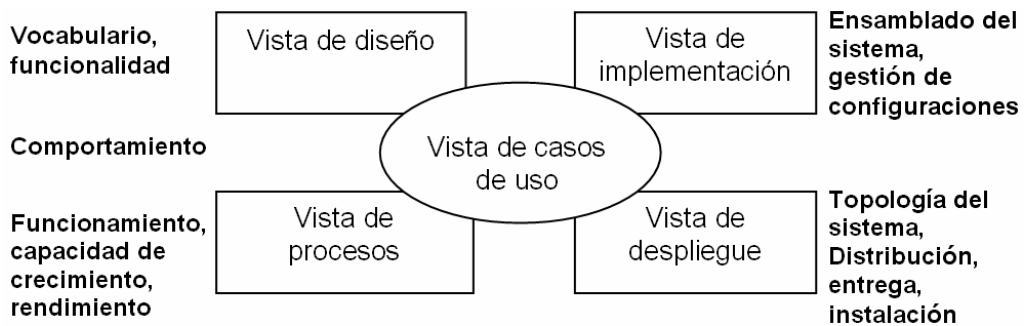


Figura 6. 2. Modelado de la arquitectura de un sistema.

A continuación se especifican los productos y los hitos generados en cada una de las fases del proceso.

Tabla 6.1. Tabla de productos e hitos generados en cada una de las fases del RUP.

Fase	Productos
Inicio	Hito: <i>Objetivos del ciclo de vida</i>
	<ul style="list-style-type: none"> • Un documento que contiene: Objetivos, requerimientos generales, características principales y restricciones. • Desarrollo de 10 a 20% de los casos de uso. • Glosario. • Contexto del negocio, criterios de éxito, pronóstico financiero. • Análisis inicial de riesgos. • Esquema inicial de la arquitectura

	<ul style="list-style-type: none"> • borrador del plan de proyecto.
Elaboración	<p style="text-align: center;">Hito: <i>Arquitectura estable</i></p> <ul style="list-style-type: none"> • Modelo de casos de uso 80% completo con descripciones detalladas, otros requerimientos no funcionales o no asociados a casos de uso. • Descripción de la arquitectura del software, vistas de los modelos de casos de uso, diseño y despliegue. • Un prototipo. • Plan de proyecto.
Construcción	<p style="text-align: center;">Hito: <i>Funcionalidad operativa inicial</i></p> <ul style="list-style-type: none"> • Producto de software integrado y corriendo en la plataforma adecuada. • Modelos del sistema completos. • Manuales de usuario, técnicos y administrativos
Transición	<p style="text-align: center;">Hito: <i>Versión de producto</i></p> <ul style="list-style-type: none"> • Pruebas Beta para validar el producto con las expectativas del cliente. • Entrenamiento de usuarios. • Distribución del producto.

6.3 CONCEPTO INICIAL DE LA ARQUITECTURA DE LA APLICACIÓN

La aplicación está soportada por una plataforma operativa como se muestra en la figura 6.3 conformada por el sistema operativo, el sistema manejador de base de datos, el servidor Web y los navegadores de Internet. Sobre dicha plataforma operativa se encuentran interactuando las tres capas del modelo basado en Web a través de interfaces seguras para proteger la aplicación.



Figura 6.3. Arquitectura global de la aplicación.

La capa de las reglas del negocio se encuentra dividida en subsistemas que poseen funciones bien definidas así:

Subsistema de administración de piezas: es el encargado de manejar la información relacionada con los especímenes anatomopatológicos del museo como fotografías, diagnósticos, historia clínica del donante y palabras clave para la facilitar la búsqueda. Permite agregar, modificar y eliminar piezas.

Subsistema de mantenimiento de inventario: proporciona funciones para apoyar el mantenimiento físico del inventario como impresión de rótulos e informes.

Subsistema de consulta de material: es el encargado de apoyar y facilitar las consultas del usuario Web.

Subsistema de administración de prácticas: su función principal es agilizar y facilitar la realización de las diferentes actividades y clases prácticas que ofrece el Departamento de Patología.

Subsistema de administración de usuarios: permite la creación, modificación y eliminación de usuarios y perfiles de usuario.

Subsistema de administración de software: es el encargado de la instalación y desinstalación de la aplicación, además ofrece funciones de respaldo y copias de seguridad.

6.4 ITERACIONES DE LA FASE DE CONSTRUCCIÓN

Durante la fase de construcción se realizaron 5 iteraciones en las cuales se ejecutaran los flujos de trabajo de requerimientos, análisis y diseño, implementación, prueba, administración de cambios y configuraciones y administración del proyecto. Para cada iteración se asignaron uno o dos subsistemas dependiendo de su complejidad así:

Iteración 1: subsistema de administración de piezas. Y subsistema de consulta de material.

Iteración 2: subsistema de mantenimiento de inventario.

Iteración 3: subsistema de administración de prácticas.

Iteración 4: subsistema de administración de usuarios y mecanismos de seguridad.

Iteración 5: subsistema de administración de software.

6.5 CRONOGRAMA DE ACTIVIDADES PARA UNA ITERACIÓN

El siguiente cronograma fue el que se elaboró inicialmente para cada una de las iteraciones de la fase de construcción, como se podrá observar, cada iteración está conformada por tres semanas de trabajo de la siguiente manera:

- Semana 1.
 - Miércoles
 - Elaborar el diagrama de casos de uso.
 - Elaborar la especificación de Requisitos Funcionales.
 - Jueves
 - Elaborar diagrama de clases para las reglas del negocio.
 - Elaborar Diagrama entidad-relación.
 - Reunión del comité de sistemas de información.

- Revisión de la especificación de requerimientos.
- Viernes
 - Corregir diagramas.
 - Corregir requerimientos.
 - Corregir código.
 - Hacer pruebas.
- Lunes
 - Elaborar el modelo de implementación.
 - Desarrollar las clases de reglas del negocio.
 - Desarrollar la base de datos.
 - Hacer pruebas
- Martes
 - Desarrollar las clases de reglas del negocio.
 - Desarrollar la base de datos.
 - Hacer pruebas
- Semana 2.
 - Miércoles
 - Desarrollar clases de reglas del negocio.
 - Desarrollar las interfaces entre la base de datos y las clases de reglas del negocio.
 - Hacer pruebas.
 - Jueves

- Desarrollar las interfaces entre las bases de datos y las clases de reglas del negocio.
 - Hacer pruebas.
 - Reunión del comité de sistemas de información.
 - Viernes
 - Corregir diagramas.
 - Corregir requerimientos.
 - Corregir código.
 - Hacer pruebas.
 - Lunes
 - Elaborar el diagrama de clases para la interfaz de usuario.
 - Desarrollar las clases de interfaz de usuario.
 - Hacer pruebas.
 - Martes
 - Desarrollar las clases de interfaz de usuario.
 - Desarrollar las interfaces entre las clases de reglas del negocio y las clases de interfaz de usuario.
 - Elaborar las alternativas de la interfaz gráfica de usuario.
 - Hacer pruebas.
- Semana 3.
 - Miércoles
 - Elaborar las alternativas de la interfaz gráfica de usuario.
 - Implementar el prototipo funcional.

- Hacer pruebas
- Jueves
- Elaborar las alternativas de la interfaz gráfica de usuario.
- Implementar el prototipo funcional.
- Hacer pruebas.
- Reunión del Comité de Sistemas de Información.
 - Selección, por parte del comité, de una alternativa de interfaz gráfica de usuario.
 - Revisión del prototipo funcional.
- Viernes
 - Corregir diagramas.
 - Corregir requerimientos.
 - Corregir código.
 - Hacer pruebas.
- Lunes
 - Elaborar manuales de usuario, técnicos y administrativos.
 - Elaborar la ayuda en línea.
- Martes
 - Elaborar la documentación del proyecto.
 - Gestión del proyecto.
 - Mostrar el avance del proyecto al director.

Debido a la inexperiencia de los autores en el cálculo del tiempo requerido para cada actividad, no se pudo cumplir este cronograma aunque si se realizaron cada una de las actividades enunciadas anteriormente.

La primera iteración tomo más tiempo que las demás, debido a que los autores tuvieron que aprender a utilizar las herramientas y tecnología seleccionada. Además se realizaron algunas actividades adicionales como el diseño e implementación de las clases para el manejo de las excepciones, el diseño de la plantilla para la interfaz de presentación y el diseño general del modelo de implementación. Pero el factor que más influyó en el retraso fue el haber incluido demasiados casos de uso para una sola iteración, tomando incluso el doble del tiempo que se tenía presupuestado inicialmente. A partir de la segunda iteración el tiempo de desarrollo se incremento notablemente lo que permitió recuperar el tiempo invertido en la primera iteración y de esta forma se pudo cumplir con el plazo estipulado en el cronograma para la entrega del software.

A continuación se describirá el plan de trabajo real llevado a cabo por los autores durante la fase de construcción.

Actividades realizadas en cada iteración de la fase de construcción

- Actividad 1.
 - Elaborar el diagrama de casos de uso.
 - Elaborar la especificación de Requisitos Funcionales.

- Actividad 2.
 - Elaborar diagrama de clases para las reglas del negocio.
 - Elaborar diagrama de clases para interfaz con la base de datos.
 - Elaborar el diagrama de clases para la interfaz de usuario.
 - Elaborar Diagrama entidad-relación.
 - Elaborar las alternativas de la interfaz gráfica de usuario en papel (véase Anexo B).

- Actividad 3.
 - Reunión del comité de sistemas de información.
 - Revisión de la especificación de requerimientos y diseño preliminar.
 - Corregir diagramas.
 - Corregir requerimientos.
 - Gestión del proyecto (dividir el trabajo).
- Actividad 4.
 - Elaborar el modelo de implementación.
 - Desarrollar las clases de reglas del negocio.
 - Desarrollar la base de datos.
- Actividad 5.
 - Elaborar pruebas automáticas para las clases de reglas del negocio desarrolladas.
 - Corregir código defectuoso.
- Actividad 6.
 - Desarrollar las clases de interfaz entre las bases de datos y las clases de reglas del negocio.
- Actividad 7.
 - Elaborar pruebas automáticas para las clases de interfaz con la base de datos desarrolladas.
 - Corregir código defectuoso.
- Actividad 8.
 - Desarrollar las clases de interfaz de usuario.
- Actividad 9.

- Elaborar manual del programador.
- Actividad 10.
 - Hacer pruebas de todos los elementos desarrollados a través de la interfaz grafica de usuario.
 - Corregir código defectuoso.
- Actividad 11.
 - Reunión del comité de sistemas de información para revisar el prototipo funcional.

Actividades realizadas al final de la fase de construcción

- Elaborar manuales de usuario.
- Elaborar la documentación del proyecto.
- Probar el sistema en su totalidad (cada uno de los caso de uso).
- Elaborar el acta de entrega.

7 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE PARA SIMUPAT

El presente documento concerniente a la Especificación de requisitos Software (ERS) del Sistema de Información del Museo de Patología SIMUPAT, se ha estructurado siguiendo las recomendaciones formuladas por el estándar IEEE¹ y utilizando la metodología para elicitación de requisitos software propuesta por Amador Durán Toro y Beatriz Bernárdez Jiménez². Fue realizado con base en la información recolectada en las reuniones periódicas llevadas a cabo por el comité de sistemas de información del museo.

7.1 INTRODUCCIÓN

7.1.1 Propósito de la ERS

El propósito del siguiente documento es la creación de una guía de especificación requisitos completa y precisa, para que el equipo de Diseño e implementación tenga parámetros claros sobre el Sistema que se va a desarrollar. Además, se utilizará para determinar el avance del proyecto y como documento de respaldo en el momento de entregar el producto terminado.

El documento va dirigido al comité de sistemas de información del museo y alcanzará su aprobación hasta que todos los integrantes del comité lleguen a un consenso sobre su contenido.

7.1.2 Participantes

Ver la sección 3.3.

7.1.3 Definición del problema

Ver la sección 3.3.

¹ IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830–1998, Institute of Electrical and Electronics Engineers, 1993.

² DURAN TORO, Amador y BERNÁRDEZ JIMÉNEZ, Beatriz. Metodología para elicitación de requisitos software, versión 2.1. Universidad de Sevilla. Departamento de Lenguajes y Sistemas Informáticos, facultad de Informática y Estadística. Sevilla, España, Octubre de 2000.

7.1.4 Alcance del sistema

El objetivo principal de SIMUPAT es contribuir a la organización del museo de piezas macroscópicas perteneciente al departamento de patología UIS y publicar material didáctico de consulta a través de Internet para los estudiantes de la facultad de salud. Aunque se proyecta como una herramienta de consulta para la comunidad médica internacional interesada en el tema, el enfoque de SIMUPAT no es educativo sino de organización y mantenimiento de las existencias del museo. Lo ideal sería que en el futuro se aproveche el material didáctico existente en el museo y publicado por SIMUPAT para el desarrollo de proyectos de software educativo.

En cuanto a los beneficios esperados debido a la implantación de SIMUPAT se encuentran los siguientes:

- **Soporte administrativo:** Se reducirán considerablemente los costos de manejo del museo, manteniendo un control real sobre el mismo, lo cual hará que el personal (estudiantes, profesores y técnicos) que lo utilice pueda administrarlo de una forma más efectiva.
- **Publicación de material didáctico:** Se publicará material didáctico de gran utilidad para los estudiantes de la facultad de salud gracias al sistema de consulta en línea que estará apoyado por una galería digital de fotografías, contribuyendo de esta forma con la misión institucional de la Universidad Industrial de Santander, en lo referente a las premisas de formación de personas de alta calidad profesional y su propósito de generación y adecuación de conocimiento.
- **Agilidad en la búsqueda de material:** El personal técnico administrará de manera más fácil, los especímenes que conforman la colección del museo gracias a los reportes de inventario, implantación de rótulos con código de barras y demás características de manejo de inventario que poseerá el sistema.
- **Mejores prácticas de clase:** Los profesores podrán diseñar nuevas prácticas de clase que aprovechen mejor los recursos disponibles pues se conocerá exactamente de que dispone el museo.

7.2 DESCRIPCIÓN GLOBAL DEL PRODUCTO A DESARROLLAR

7.2.1 Perspectiva del producto

SIMUPAT es un producto independiente y completamente autónomo, es decir, no hace parte de ningún sistema mayor ni utiliza funcionalidades de sistemas externos, excepto por componentes complementarios como librerías o clases de presentación que contribuyan a un desarrollo más rápido del sistema.

Entre los principales usuarios de SIMUPAT se encuentran los profesores del departamento de patología, los estudiantes de pregrado y postgrado de la facultad de salud, el personal técnico del museo y en general cualquier persona perteneciente a la comunidad medica internacional interesada en el tema, que cuente con una conexión a Internet. Por esta razón la herramienta debería ofrecer una interfaz tanto en español como en ingles para que personas de diferentes nacionalidades puedan utilizarla, y debe ser sencilla e intuitiva para que el usuario pueda familiarizarse con ella fácilmente.

La Herramienta SIMUPAT será una aplicación basada en Web con arquitectura de tres capas (véase la sección 8.4) y por lo tanto necesita para su correcto funcionamiento contar con un servidor Web y un servidor de datos.

7.2.2 Funciones del producto

El Sistema SIMUPAT estará capacitado para ejecutar las siguientes funciones:

- Controlar, en tiempo real, el nivel de existencia (número de ejemplares) de cada una de las piezas pertenecientes al museo.
- Mantener un registro exacto de la localización física de cada ejemplar en resina de la pieza, de tal forma que su búsqueda dentro de las estanterías del museo sea inmediata.
- Alertar, con la debida anticipación, al técnico sobre pedidos de bloques para que éste lleve a cabo los procedimientos necesarios, de manera que estén disponibles cuando se lleven a cabo las prácticas de clase.
- Organizar las prácticas de clase.

- Publicar en Internet, mediante un sistema de consulta en línea, la información morfológica, clínica y fotográfica disponible para cada pieza.
- Mostrar reportes sobre el nivel de existencias del museo, estado actual de cada bloque de resina, pérdidas o faltantes, bloques deteriorados.
- Imprimir reportes, catálogos y rótulos de bloques.
- Mostrar ayuda para todas las funciones del producto.

7.2.3 Características generales del usuario

Los usuarios en general deberán tener conocimientos básicos sobre anatomía y patología para utilizar correctamente la aplicación. Aun así, si un usuario quisiera ampliar la funcionalidad del producto le sería de gran ayuda tener conocimientos avanzados en patología y conocer la CIE-10.

7.2.4 Suposiciones y dependencias

La presente Especificación de Requerimientos de Software, fue escrita asumiendo los siguientes puntos:

- **Estabilidad de los requisitos:** Se supone que una vez que los Integrantes del CSI lleguen a un consenso sobre los requisitos software para el sistema SIMUPAT, el presente documento alcanzara su aprobación y a partir de ese momento dichos requisitos serán invariables, de manera que sí se propone algún cambio, deberá quedar consignado en una futura versión del presente documento, la cual tendrá que ser evaluada por el CSI antes de dar su aprobación.
- **Disponibilidad de un servidor:** La aplicación que se va a desarrollar necesita de un servidor, el cual se considera disponible, que además de permitir la publicación de sitios Web, sirva como alojamiento para la base de datos y la galería de fotografías del Museo. Se supone a este servidor correctamente administrado, con políticas claras sobre seguridad y copias de respaldo.
- **Disponibilidad del hardware de control:** Se refiere a que los lectores de código de barras, laminadoras plásticas, impresoras de código de barras, equipos con acceso a red y suministros para los mismos, se encuentran disponibles.

- **Mantenimiento periódico del inventario físico:** Se supone que la Administración del Museo, adoptará políticas que hagan mantenimiento al inventario físico, tales como reemplazar a tiempo rótulos deteriorados, reportar a tiempo cambios en el estado de preservación de los bloques de resina, etc.
- **Operabilidad de la red UIS:** Se asume que la red UIS estará operativa permanentemente y que cualquier percance se solucione en un término no mayor a 24 horas.
- **Continuidad en la administración:** Se supone que se nombrará un administrador del sistema y que éste llevará a cabo un mantenimiento periódico del mismo.
- **Reglamentación interna:** Se supone que la Administración del Museo, velará por el estricto cumplimiento de las normas referentes al manejo de los especímenes del museo y que dicha reglamentación estará consignada en un Manual de Procedimientos.
- **Capacitación y buena fe:** Se asume que el personal del Museo cursará la capacitación pertinente en el manejo de Sistema y que no empleará medidas que atenten deliberadamente contra el sistema (sabotaje).
- **Licencias:** Se supone que la Administración del Museo tiene las licencias necesarias del software que estará instalado en los equipos terminales, como mínimo el software instalado será un Sistema Operativo y un Navegador de Internet.

7.3 OBJETIVOS DEL SISTEMA, SEGÚN ESTA ESPECIFICACIÓN DE REQUERIMIENTOS

A continuación se enuncian los objetivos del sistema, contemplados en esta especificación de requerimientos de software. **No deben confundirse** estos objetivos con los objetivos específicos del proyecto (véase la sección 1.2) aunque estén íntimamente relacionados.

OBJ-01	Administrar el material
Versión	1.0
Descripción	El sistema deberá permitir crear, eliminar y modificar piezas y casos clínicos

Importancia	Esencial
Estado	Estable

OBJ-02	Publicar el material
Versión	1.0
Descripción	El sistema deberá permitir publicar información relativa a las piezas y casos clínicos pertenecientes al museo.
Importancia	Esencial
Estado	Estable

OBJ-03	Mantener el inventario físico
Versión	1.0
Descripción	El sistema deberá permitir hacer mantenimiento al inventario físico, en concreto: controlar las existencias de bloques, imprimir rótulos, realizar informes e imprimir catálogos.
Importancia	Esencial
Estado	Estable

OBJ-04	Administrar usuarios
Versión	1.0
Descripción	El sistema deberá permitir crear, modificar y eliminar usuarios, en concreto: profesores, técnicos, administradores del museo y administradores del software.

Importancia	Esencial
Estado	Estable

OBJ-05	Administrar las prácticas de clase
Versión	1.0
Descripción	El sistema deberá permitir crear, modificar y eliminar la información relevante a las prácticas de clase y dar soporte a los procesos de prestar y guardar bloques para su uso en dichas prácticas o como repaso a los estudiantes.
Importancia	Estable
Estado	Estable

OBJ-07	Proporcionar ayuda
Versión	1.0
Descripción	El sistema deberá proporcionar ayuda de todas las funciones a las que tenga acceso el usuario.
Importancia	Condicional
Estado	Estable

7.4 REQUERIMIENTOS DE ALMACENAMIENTO DE INFORMACIÓN

RI-01	Información sobre piezas
Versión	1.0

Objetivos asociados	OBJ-01 Administrar el material OBJ-02 Publicar el material
Requisitos asociados	RI-02 Información sobre diagnósticos RI-03 Información sobre bloques de resina.
Descripción	El sistema deberá almacenar la información correspondiente a las piezas, en concreto:
Datos específicos	<ul style="list-style-type: none"> • Código de la pieza (CIE10). • Descripción macroscópica (en español y en inglés). • Fotografías de la pieza (Macro y Micro) con comentarios en español y en inglés. • Diagnóstico principal. • Diagnósticos secundarios. • Palabras clave para optimizar la búsqueda (mínimo 3, máximo 10, en español y en inglés) • Resumen de la historia clínica (opcional, en español y en inglés). • Fotografías de la historia clínica (opcional). • Bloques de Resina Asociados.
Intervalo temporal	Presente

RI-02	Información sobre diagnósticos
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material

	OBJ-02 Publicar el material
Requisitos asociados	
Descripción	El sistema deberá almacenar la información correspondiente a los diagnósticos, en concreto:
Datos específicos	<ul style="list-style-type: none"> • Código del diagnóstico (CIE10). • Nombre del diagnóstico (en español y en inglés). • Nombre del órgano en el cual el diagnóstico se presenta, en español y en inglés.
Intervalo temporal	Presente

RI-03	Información sobre bloques de resina
Versión	1.0
Objetivos asociados	<p>OBJ-01 Administrar el material</p> <p>OBJ-02 Publicar el material</p>
Requisitos asociados	RI-01 Información sobre piezas
Descripción	El sistema deberá almacenar la información correspondiente a los bloques de resina que contienen las piezas, en concreto:

Datos específicos	<ul style="list-style-type: none"> • Código de la pieza contenida en el bloque. • Código del bloque de resina, construido con caracteres compatibles para código de barras. • Fecha de ingreso del bloque. • Fecha de egreso o modificación del bloque. • Estado del bloque: <ul style="list-style-type: none"> ▫ Prestado (dentro de las instalaciones) ▫ Guardado en estanterías. ▫ Averiado ▫ En mantenimiento ▫ Extraviado ▫ Desechado
Intervalo temporal	Pasado y presente

RI-04	Información sobre usuarios autenticados
Versión	1.0
Objetivos asociados	OBJ-04 Administrar los usuarios
Requisitos asociados	
Descripción	El sistema deberá almacenar la información correspondiente a los usuarios, en concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre y apellidos • Dirección de correo electrónico.

	<ul style="list-style-type: none"> • Nombre de usuario y contraseña. • Tipo de usuario, en concreto: <ul style="list-style-type: none"> ▫ Administrador del sistema ▫ Administrador del museo ▫ Técnico ▫ Residente
Intervalo temporal	Presente

RI-05	Información sobre prácticas
Versión	1.0
Objetivos asociados	OBJ-05 Administrar prácticas de clase.
Requisitos asociados	RI-01 Información sobre piezas
Descripción	El sistema deberá almacenar la información correspondiente a las prácticas dictadas por el Departamento de Patología, que hacen uso de los especímenes almacenados en el museo. Debe administrar la siguiente información:
Datos específicos	<ul style="list-style-type: none"> • Nombre de la práctica. • Tipo de la práctica (MACRO-1, MACRO-2, MICRO). • Lista de piezas que se van a estudiar en esta práctica. • Semana de realización de la práctica, respecto del calendario académico determinado por el Departamento de Patología.

Intervalo temporal	Presente
--------------------	----------

7.5 REQUERIMIENTOS FUNCIONALES

7.5.1 Diagramas de casos de uso

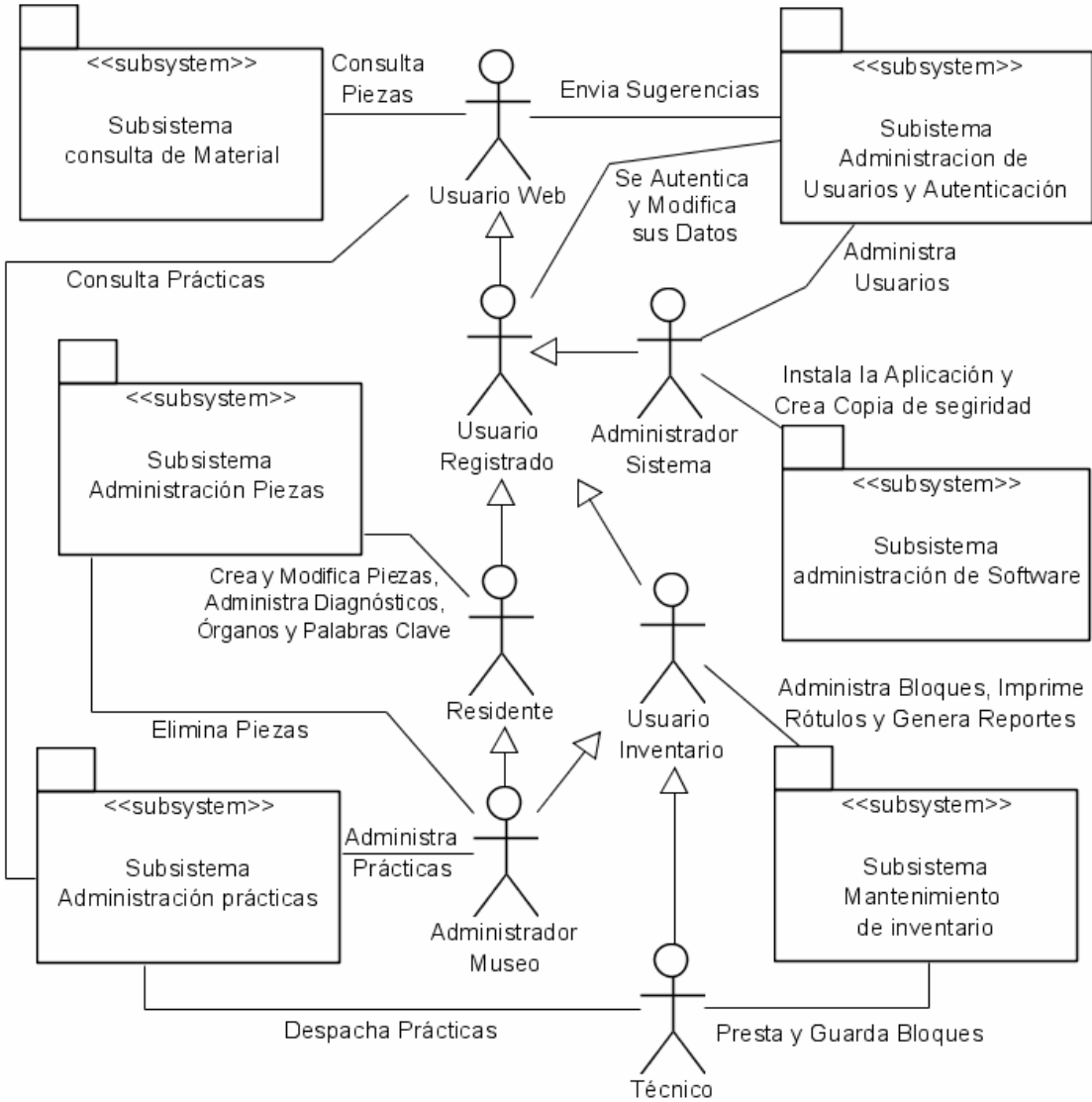


Figura 7.1. Diagrama de Subsistemas que muestra la interacción entre los diferentes actores de SIMUPAT y los subsistemas (módulos) que lo componen.

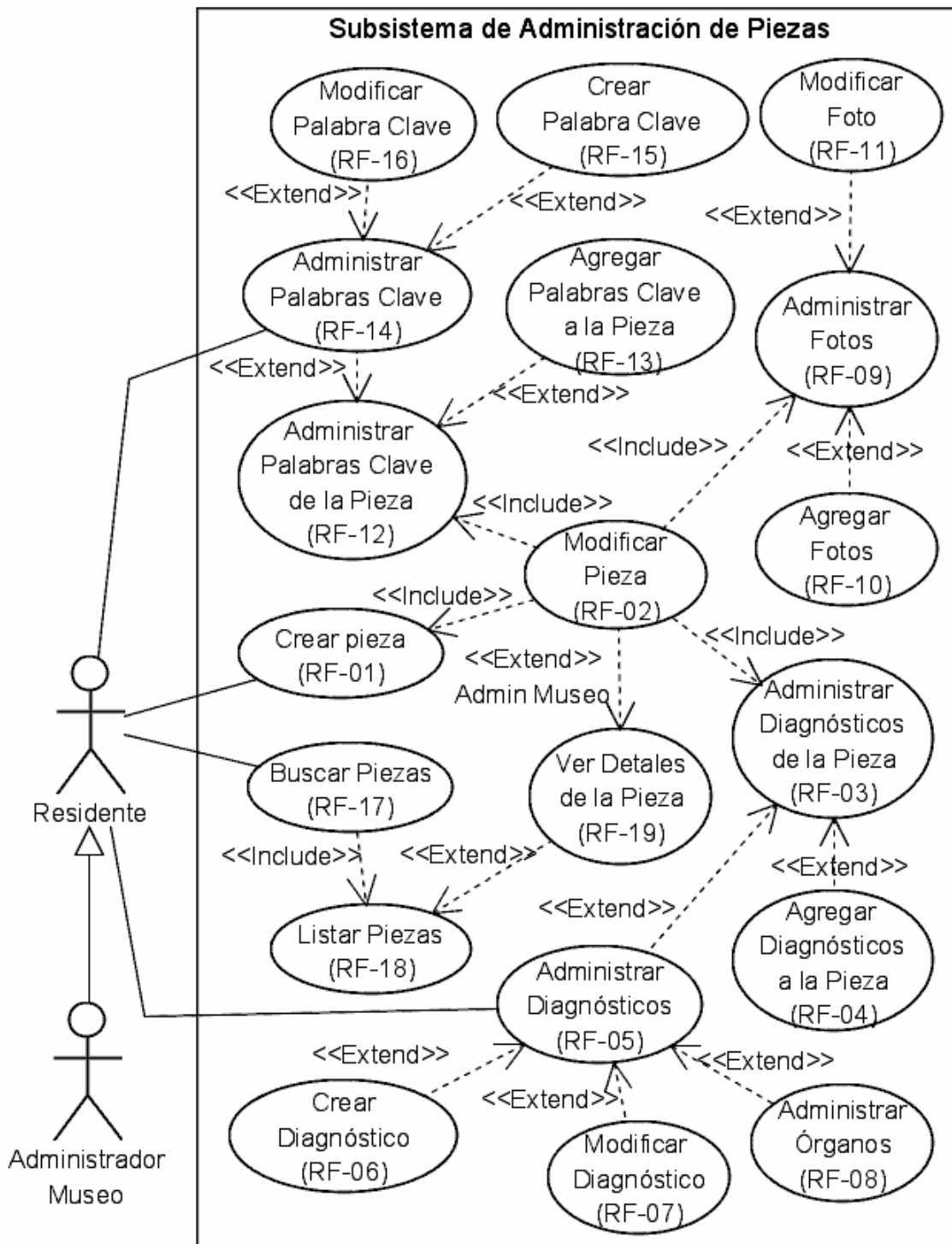


Figura 7.2. Diagrama de casos de uso del módulo (subsistema) de administración de piezas.

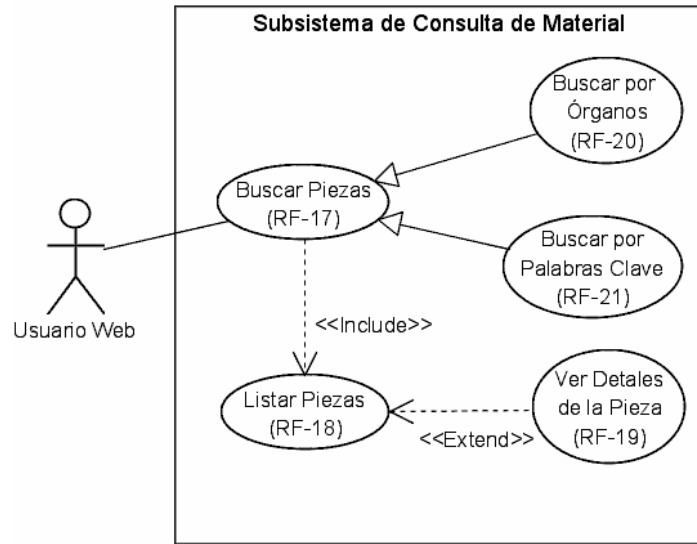


Figura 7.3. Diagrama de casos de uso para el módulo de consulta.

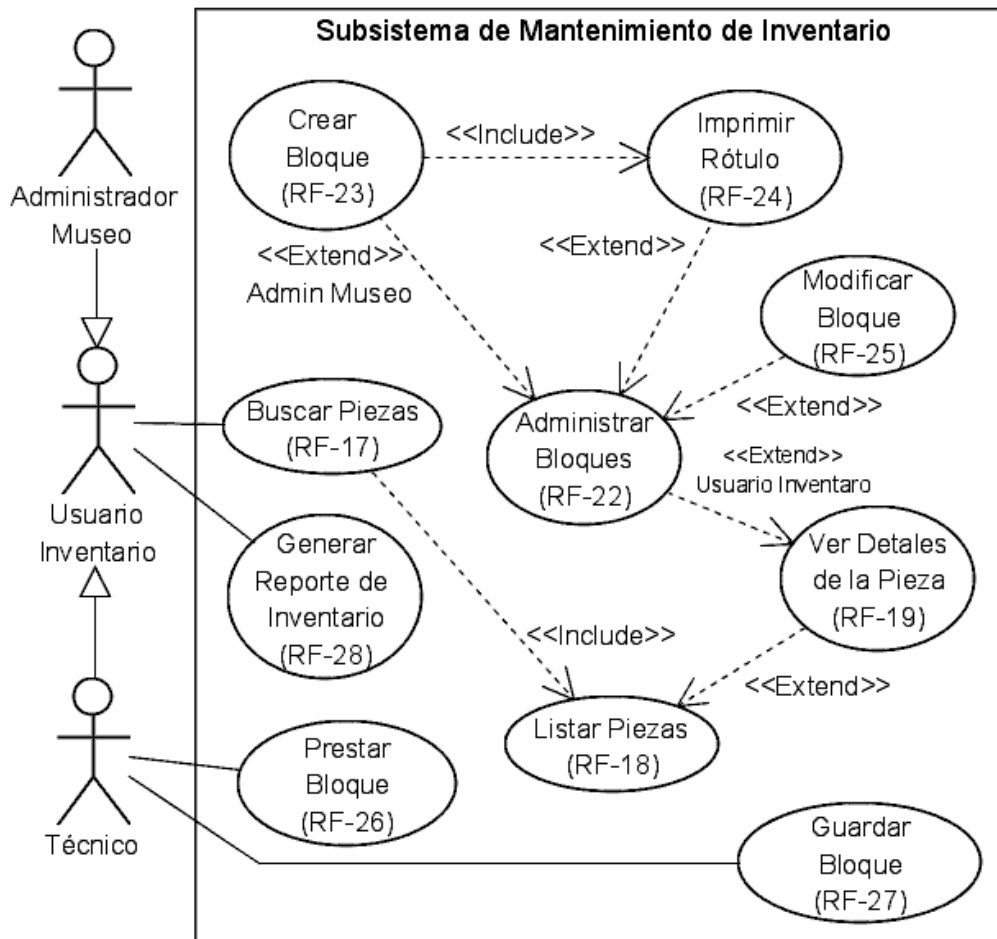


Figura 7.4. Diagrama de casos de uso para el módulo de mantenimiento de inventario.

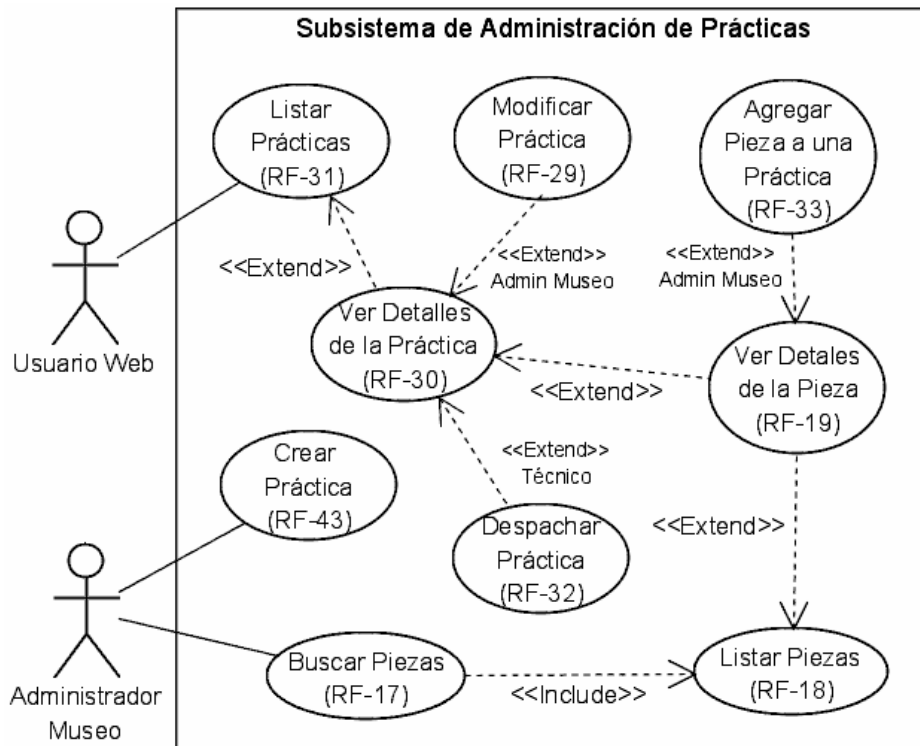


Figura 7.5. Diagrama de casos de uso del módulo (subsistema) de administración de prácticas.

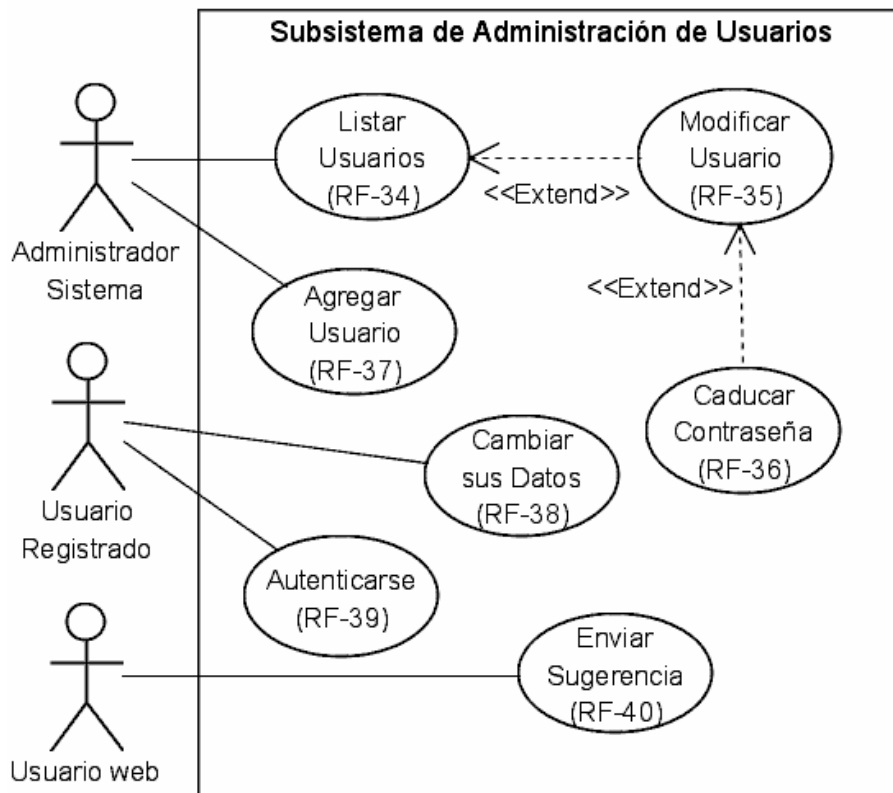


Figura 7.6. Diagrama de casos de uso del módulo (subsistema) de administración de usuarios.

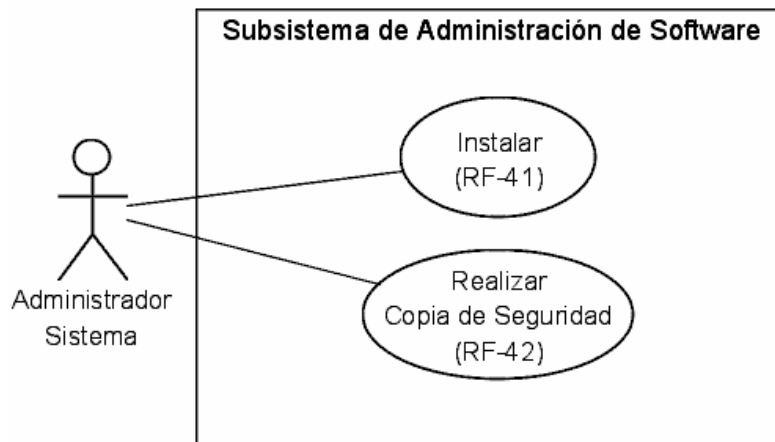


Figura 7.7. Diagrama de casos de uso para el subsistema de administración de software.

7.5.2 Definición de actores

ACT-01	Usuario Web
Versión	1.0
Descripción	Este actor representa a los usuarios que consultan la colección a través de Internet, es decir, todos los usuarios.

ACT-02	Residente
Versión	1.0
Descripción	Este actor representa a los residentes del Departamento de patología.

ACT-03	Técnico
Versión	0.0
Descripción	Este actor representa a los técnicos del Museo de patología.

ACT-04	Administrador del museo
Versión	0.0
Descripción	Este actor representa al personal encargado de la administración del material del museo.

ACT-05	Administrador del sistema
Versión	0.0
Descripción	Este actor representa a las personas encargadas de administrar la información de los usuarios del sistema y de hacer las labores de mantenimiento y copia de seguridad.

7.5.3 Facultades de usuarios

Esta sección muestra cuadros asociativos de requerimientos funcionales y facultades de usuario cuyo propósito es exponer una visión condensada de las facultades a las que tiene derecho cada tipo de usuario en el Sistema. Para todos los usuarios, excepto el Usuario Web, se requiere autenticación (véase la sección 8.4.6.1).

7.5.3.1 Tipos de facultades

Los tipos de facultades o derechos contemplados en este cuadro son los siguientes:

- **Total (✓):** El usuario tiene acceso a todas las características y funciones detalladas en el requerimiento funcional.
- **Limitada (☑):** El usuario sólo tiene derecho a algunas de las características y funciones detalladas en el requerimiento funcional
- **Negada (✗):** El usuario no puede acceder a la funcionalidad de este requerimiento.

Tabla 7.1. Resumen de los permisos de usuario para los requerimientos del RF-01 al RF-20.

Identificación del requerimiento.	Usuario Web	Residen	Técnico	Admin. Museo	Admin. Sistema
RF-01 Crear Pieza	x	✓	x	✓	x
RF-02 Modificar Pieza	x	<input checked="" type="checkbox"/>	x	✓	x
RF-03 Administrar diagnósticos de la pieza.	x	✓	x	✓	x
RF-04 Agregar diagnósticos a la pieza.	x	✓	x	✓	x
RF-05 Administrar diagnósticos.	x	✓	x	✓	x
RF-06 Crear diagnóstico.	x	✓	x	✓	x
RF-07 Modificar diagnóstico.	x	✓	x	✓	x
RF-08 Administrar órganos.	x	✓	x	✓	x
RF-09 Administrar fotografías.	x	✓	x	✓	x
RF-10 Agregar fotografías.	x	✓	x	✓	x
RF-11 Modificar fotografía.	x	✓	x	✓	x
RF-12 Administrar palabras clave de la pieza.	x	✓	x	✓	x
RF-13 Agregar palabras clave a la Pieza.	x	✓	x	✓	x
RF-14 Administrar palabras clave	x	✓	x	✓	x
RF-15 Crear palabra clave.	x	✓	x	✓	x
RF-16 Modificar palabra clave.	x	✓	x	✓	x

RF-17 Buscar piezas	✓	✓	✓	✓	✓
RF-18 Listar piezas.	✓	✓	✓	✓	✓
RF-19 Ver detalles de la pieza.	✓	✓	✓	✓	✓
RF-20 Buscar por órganos	✓	✓	✓	✓	✓

Tabla 7.2. Resumen de los permisos de usuario para los requerimientos del RF-21 al RF-40.

Identificación del requerimiento.	Usuario Web	Residen	Técnico	Admin. Museo	Admin. Sistema
RF-21 Buscar por palabras clave	✓	✓	✓	✓	✓
RF-22 Administrar bloques.	x	x	x	✓	x
RF-23 Crear bloque.	x	x	x	✓	x
RF-24 Imprimir rótulo.	x	x	✓	✓	x
RF-25 Modificar bloque de resina.	x	x	<input checked="" type="checkbox"/>	✓	x
RF-26 Prestar bloque.	x	x	✓	x	x
RF-27 Guardar bloque.	x	x	✓	x	x
RF-28 Generar reporte de inventario.	x	x	✓	✓	x
RF-29 Modificar práctica.	x	x	x	✓	x
RF-30 Ver detalles de la práctica.	✓	✓	✓	✓	✓
RF-31 Listar prácticas.	✓	✓	✓	✓	✓

RF-32 Despachar práctica.	x	x	✓	x	x
RF-33 Agregar pieza a una práctica.	x	x	x	✓	x
RF-34 Listar usuarios.	x	x	x	x	✓
RF-35 Modificar usuario.	x	x	x	x	✓
RF-36 Caducar contraseña.	x	x	x	x	✓
RF-37 Agregar usuario.	x	x	x	x	✓
RF-38 Cambiar sus datos.	x	✓	✓	✓	✓
RF-39 Autenticarse.	x	✓	✓	✓	✓
RF-40 Enviar sugerencia.	✓	✓	✓	✓	✓

Tabla 7.3. Resumen de los permisos de usuario para los requerimientos del RF-41 al RF-43.

Identificación del requerimiento.	Usuario Web	Residen	Técnico	Admin. Museo	Admin. Sistema
RF-41 Instalar	x	x	x	x	✓
RF-42 Realizar copia de seguridad.	x	x	x	x	✓
RF-43 Crear práctica.	x	x	x	✓	x

7.5.4 Tablas asociadas a los casos de uso

RF-01	Crear pieza
Versión	1.0

Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-01 Información sobre piezas RF-02 Modificar pieza	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo solicite la creación de una nueva pieza:	
Precondición	El administrador del museo dispone de las fotografías macro de la pieza. El administrador del museo se encuentra en el menú principal.	
Secuencia normal	Paso	Acción
	1	El administrador del museo solicita al sistema comenzar el proceso de creación de una nueva pieza.
	2	Se realiza el caso de uso RF-02 Modificar pieza, partiendo de una pieza vacía y a continuación este caso de uso continúa.
	3	El sistema pregunta al usuario si desea crear otra pieza. Si el usuario responde que si, el caso de uso continúa en el paso 2. Si el usuario responde que no, el caso de uso termina.
Poscondición	El sistema muestra el menú principal	
Excepciones	Ninguna	

RF-02	Modificar pieza
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material

Requisitos asociados	RI-01 Información sobre piezas RF-03 Administrar diagnósticos de la pieza RF-09 Administrar fotos RF-12 Administrar palabras clave de la pieza RF-01 Crear pieza. RF-19 Ver detalles de la pieza.	
Descripción	<p>El sistema debe proveer de una interfaz para la modificación de la información pertinente a cada pieza, como respuesta a diversas solicitudes de revisión, o por cualquier otra circunstancia que el Administrador contemple.</p> <p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso:</p>	
Precondición	<p>El administrador del museo puede acceder a este caso de uso de tres maneras diferentes:</p> <ol style="list-style-type: none"> 1. Observando los detalles de una pieza (RF-19). 2. Creando una pieza (RF-01). 3. Solicitando a través del menú principal la modificación de una pieza en particular. El sistema muestra la lista de todas las piezas para que el administrador del museo seleccione la que desea modificar. Cada ítem de la lista consta de código CIE10, diagnóstico y órgano. 	
Secuencia normal	Paso	Acción

	1	<p>El sistema muestra los siguientes datos utilizando elementos gráficos para la edición de su contenido:</p> <ol style="list-style-type: none"> 1 Cuadros de texto para la descripción macroscópica en español y en inglés. 2 Cuadros de texto para el resumen de historia clínica en español y en inglés. 3 RF-03 Administrar diagnósticos de la pieza: controles que permiten agregar y eliminar diagnósticos de la pieza 4 RF-12 Administrar palabras clave de la pieza: controles que permiten agregar y eliminar palabras clave de la pieza 5 RF-09 Administrar fotos: controles que permiten agregar, modificar y eliminar fotos de la pieza. <p>Además el sistema muestra un botón que permitirá abortar este caso de uso.</p> <p>Si se trata de una pieza nueva todas las listas y campos de texto aparecerán vacíos.</p>
	2	<p>El administrador del museo proporciona los datos requeridos, realiza los cambios que considere pertinentes y solicita al sistema que los almacene</p>
	3	<p>El sistema presenta al usuario la información completa de la pieza, con los cambios resaltados y pregunta al Usuario si Acepta o Rechaza los cambios.</p>

	4	El administrador del museo confirma que los datos de la pieza son correctos.
	5	El sistema informa al administrador que el proceso ha terminado con éxito.
Poscondición	El sistema muestra los detalles de la pieza RF-19	
Excepciones	Paso	Acción
	1	Si el administrador del museo solicita cancelar la operación, el sistema cancela y a continuación este caso de uso termina
	3	Si faltan datos el sistema alerta al administrador del museo permitiéndole completar los datos. El sistema considera que faltan datos cuando: <ul style="list-style-type: none"> - Han quedado campos vacíos o con información inválida - No se ha agregado un diagnóstico principal - No se han agregado fotos macro - Se han agregado menos de 3 palabras clave

RF-03	Administrar diagnósticos de la pieza
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material
Requisitos asociados	RI-01 Información sobre piezas RF-02 Modificar pieza RF-04 Agregar diagnósticos a la pieza

	RF-05 Administrar diagnósticos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso:	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-02 Modificar pieza	
Secuencia normal	Paso	Acción
	1	<p>El sistema muestra la lista de los diagnósticos de la pieza. Cada ítem de la lista consta de: Código CIE10, órgano, nombre del diagnóstico, un botón para seleccionarlo y un botón para eliminarlo.</p> <p>Al final de la lista se muestra: Un botón para eliminar los diagnósticos seleccionados un botón para Agregar diagnósticos a la lista y un botón para acceder a las funciones de administración de diagnósticos.</p>
Poscondición	El administrador del museo continúa realizando el caso de uso RF-02 Modificar pieza	
Excepciones	Paso	Acción
	1	Si el administrador del museo elige agregar un diagnóstico, se realiza el caso de uso RF-04 Agregar diagnósticos a la pieza y a continuación este caso de uso continúa
	1	Si el administrador del museo desea hacer uso de las funciones de administración de diagnósticos ya sea para corregir, crear o eliminar un diagnóstico, se realiza el caso de uso RF-05 Administrar diagnósticos y a continuación este caso de uso continúa

	1	Si el administrador del museo elige eliminar, los diagnósticos seleccionados se eliminarán de la lista de diagnósticos de la pieza. A continuación este caso de uso continúa
--	---	--

RF-04	Agregar diagnósticos a la pieza	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-01 Información sobre piezas RF-03 Administrar diagnósticos de la pieza	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo agregue diagnósticos a una pieza.	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-03 Administrar diagnósticos de la pieza. El administrador del museo conoce los diagnósticos que debe tener la pieza.	
Secuencia normal	Paso	Acción
	1	El sistema muestra la lista de los diagnósticos existentes. Cada ítem de la lista consta de: Código CIE10, órgano, nombre del diagnóstico y un botón para seleccionarlo. Al final de la lista se muestra un botón para agregar los diagnósticos seleccionados a la lista de diagnósticos de la pieza.

	2	El administrador del museo selecciona los diagnósticos que desea agregar y solicita al sistema que los agregue a la lista de diagnósticos de la pieza
	3	El sistema agrega los diagnósticos seleccionados a la lista de diagnósticos de la pieza.
Poscondición	El administrador del museo continúa realizando el caso de uso RF-02 Modificar pieza.	
Excepciones	Ninguna	

RF-05	Administrar diagnósticos
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material
Requisitos asociados	RI-02 Información sobre diagnósticos RF-06 Crear diagnóstico. RF-07 Modificar diagnóstico. RF-08 Administrar órganos. RF-03 Administrar diagnósticos de la pieza
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo desee crear nuevos diagnósticos o modificar los diagnósticos existentes.
Precondición	El administrador del museo puede acceder a este caso de uso de dos maneras diferentes: 1. Cuando se encuentra realizando el caso de uso RF-03

	<p>Administrar diagnósticos de la pieza.</p> <p>2. cuando lo Solicita a través del menú principal.</p>	
Secuencia normal	Paso	Acción
	1	<p>El sistema muestra la lista de los diagnósticos existentes. Cada ítem de la lista consta de: Código CIE10, órgano, nombre del diagnóstico en español y en inglés, un botón para seleccionarlo y un botón para modificarlo.</p> <p>Al final de la lista se muestra un botón para eliminar los diagnósticos seleccionados y un botón para crear un nuevo diagnóstico.</p>
	2	<p>Si el administrador del museo elige crear un nuevo diagnóstico, se realiza el caso de uso RF-06 Crear diagnóstico y a continuación este caso de uso continúa</p>
Poscondición	<p>El administrador del museo continúa realizando el caso de uso RF-03 Administrar diagnósticos de la pieza o vuelve al menú principal, dependiendo de la precondición que lo originó.</p>	
Excepciones	Paso	Acción
	1	<p>Si el administrador del museo elige eliminar, el sistema solicita la confirmación de esta acción. Si el administrador confirma, los diagnósticos seleccionados se eliminarán sólo si no existen piezas que contengan dichos diagnósticos.</p>
	1	<p>Si el administrador del museo elige modificar, se realiza el caso de uso RF-07 Modificar diagnóstico y a continuación este caso de uso continúa</p>

RF-06	Crear diagnóstico	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-02 Información sobre diagnósticos RF-05 Administrar diagnósticos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo solicite la creación de un nuevo diagnóstico.	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-05 Administrar diagnósticos.	
Secuencia normal	Paso	Acción
	1	El sistema solicita los siguientes datos: Código CIE10, órgano, Nombre del diagnóstico en español y en inglés.
	2	El administrador del museo proporciona los datos requeridos y solicita al sistema que los almacene
	3	El administrador del museo confirma que los datos suministrados son correctos.
	4	El sistema almacena los datos proporcionados y los agrega a la lista de diagnósticos
Poscondición	El administrador del museo continúa realizando el caso de uso RF-05 Administrar diagnósticos.	
Excepciones	Paso	Acción

	2	Si el sistema detecta que el nuevo diagnóstico ya existe, el sistema informa de la situación al administrador del museo permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa
	2	Si el administrador del museo solicita cancelar la operación, el sistema cancela y a continuación este caso de uso termina

RF-07	Modificar diagnóstico	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-02 Información sobre diagnósticos RF-05 Administrar diagnósticos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo solicite la modificación de un diagnóstico que se desea corregir.	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-05 Administrar diagnósticos.	
Secuencia normal	Paso	Acción
	1	El sistema muestra los datos actuales y solicita los nuevos: Código CIE10, Nombre del diagnóstico en español y en inglés.
	2	El administrador del museo proporciona los datos requeridos y solicita al sistema que los almacene

	3	El administrador del museo confirma que los datos suministrados son correctos.
	4	El sistema almacena los datos proporcionados y actualiza la lista de diagnósticos
Poscondición	El administrador del museo continúa realizando el caso de uso RF-05 Administrar diagnósticos.	
Excepciones	Paso	Acción
	2	Si el sistema detecta que los nuevos datos del diagnóstico corresponden a un diagnóstico que ya existe, el sistema informa de la situación al administrador permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa
	1	Si el administrador del museo solicita cancelar la operación, el sistema cancela y a continuación este caso de uso termina

RF-08	Administrar órganos
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material
Requisitos asociados	RI-02 Información sobre diagnósticos. RF-05 Administrar diagnósticos.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso cuando el administrador del museo desee crear, modificar o eliminar órganos.
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-

	05 Administrar diagnósticos.	
Secuencia normal	Paso	Acción
	1	<p>El sistema muestra la lista de órganos existentes. Cada ítem de la lista consta de: nombre del órgano en español y en inglés, un botón para seleccionarlo y un botón para modificarlo.</p> <p>Al final de la lista se muestra un botón para eliminar los órganos seleccionados y un botón para crear un nuevo órgano.</p>
Poscondición	El administrador del museo continúa realizando el caso de uso RF-02 Modificar pieza	
Excepciones	Paso	Acción
	1	<p>Si el administrador del museo elige crear o modificar un órgano:</p> <ul style="list-style-type: none"> - El sistema solicita el nombre del órgano en español y en inglés. - El sistema verifica que el órgano no exista. - El administrador del museo confirma que los datos suministrados son correctos. - El sistema almacena los datos proporcionados y actualiza la lista de órganos. <p>A continuación este caso de uso continúa</p>
	1	<p>Si el administrador del museo elige eliminar, el sistema solicita la confirmación de esta acción. Si el administrador confirma, los órganos seleccionados se eliminarán sólo si no existen piezas que contengan dichos órganos y a continuación este caso de uso continúa</p>

RF-09	Administrar fotos	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-01 Información sobre piezas RF-10 Agregar fotos RF-02 Modificar Pieza. RF-11 Modificar foto	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso ofreciendo al administrador del museo la posibilidad de agregar, modificar o eliminar fotos ya sean macro, micro o asociadas a la historia clínica de la pieza.	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-02 Modificar pieza.	
Secuencia normal	Paso	Acción
	1	El sistema muestra la lista de fotos de la pieza. Cada ítem de la lista consta de: Miniatura de la foto, Descripción de la foto y un botón para seleccionarla. Al final de la lista se muestra: Un botón para eliminar las fotografías seleccionadas y un botón para Agregar fotografías a la lista.
Poscondición	El administrador del museo continúa realizando el caso de uso RF-02 Modificar pieza	

Excepciones	Paso	Acción
	1	Si el administrador del museo elige agregar una fotografía, se realiza el caso de uso RF-10 Agregar fotos y a continuación este caso de uso continúa
	1	Si el administrador del museo elige modificar una fotografía, se realiza el caso de uso RF-11 Modificar foto y a continuación este caso de uso continúa
	1	Si el administrador del museo elige eliminar, las fotos seleccionadas se eliminarán de la lista de fotos de la pieza. A continuación este caso de uso continúa

RF-10	Agregar fotos
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material
Requisitos asociados	RI-01 Información sobre piezas RF-09 Administrar fotos
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo agregue fotografías a una pieza.
Precondición	El administrador del museo se encuentra en el caso de uso RF-09 Administrar fotos. El administrador del museo conoce el nombre de las fotos y su ubicación.

Secuencia normal	Paso	Acción
	1	El sistema solicita los siguientes datos: Ruta del archivo, descripción de la foto en español y en inglés (opcional).
	2	El administrador del museo proporciona los datos requeridos y solicita al sistema que los almacene
	3	El administrador del museo confirma que los datos suministrados son correctos.
	4	El sistema almacena los datos proporcionados y actualiza la lista de fotos de la pieza
Poscondición	El administrador del museo continúa realizando el caso de uso RF-09 Administrar fotos.	
Excepciones	Paso	Acción
	1	Si el administrador del museo solicita cancelar la operación, el sistema cancela y a continuación este caso de uso termina

RF-11	Modificar foto
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material
Requisitos asociados	RI-01 Información sobre piezas RF-09 Administrar fotos
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo modifique las fotografías de una pieza.

Precondición	<p>El administrador del museo se encuentra realizando el caso de uso RF-09 Administrar fotos.</p> <p>El administrador del museo conoce el nombre de las fotos y su ubicación.</p>	
Secuencia normal	Paso	Acción
	1	El sistema muestra los datos actuales y solicita los nuevos: Ruta del archivo, descripción de la foto en español y en inglés (opcional).
	2	El administrador del museo proporciona los datos requeridos y solicita al sistema que los almacene
	3	El administrador del museo confirma que los datos suministrados son correctos.
	4	El sistema almacena los datos proporcionados y actualiza la lista de fotos de la pieza
Poscondición	<p>El administrador del museo continúa realizando el caso de uso RF-09 Administrar fotos.</p>	
Excepciones	Paso	Acción
	1	Si el administrador del museo solicita cancelar la operación, el sistema cancela y a continuación este caso de uso termina

RF-12	Administrar palabras clave de la pieza
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material

Requisitos asociados	RI-01 Información sobre piezas RF-02 Modificar pieza RF-13 Agregar palabras clave a la pieza RF-14 Administrar palabras clave	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso:	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-02 Modificar pieza	
Secuencia normal	Paso	Acción
	1	El sistema muestra la lista de las palabras clave de la pieza. Cada ítem de la lista consta de: palabra clave, un botón para seleccionarla y un botón para eliminarla. Al final de la lista se muestra: Un botón para eliminar las palabras clave seleccionadas un botón para Agregar palabras clave a la lista y un botón para acceder a las funciones de administración de palabras clave.
Poscondición	El administrador del museo continúa realizando el caso de uso RF-02 Modificar pieza	
Excepciones	Paso	Acción
	1	Si el administrador del museo elige agregar una palabra clave, se realiza el caso de uso RF-13 Agregar palabras clave a la pieza y a continuación este caso de uso continúa

	1	Si el administrador del museo desea hacer uso de las funciones de administración de palabras clave ya sea para corregir, crear o eliminar una palabra clave, se realiza el caso de uso RF-14 Administrar palabras clave y a continuación este caso de uso continúa
	1	Si el administrador del museo elige eliminar, las palabras clave seleccionadas se eliminarán de la lista de palabras clave de la pieza. A continuación este caso de uso continúa

RF-13	Agregar palabras clave a la pieza	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-01 Información sobre piezas RF-12 Administrar palabras clave de la pieza	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo agregue diagnósticos a una pieza.	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-12 Administrar palabras clave de la pieza. El administrador del museo conoce las palabras clave que debe tener la pieza.	
Secuencia normal	Paso	Acción

	1	El sistema muestra la lista de las palabras clave existentes. Cada ítem de la lista consta de: Palabra clave y un botón para seleccionarla. Al final de la lista se muestra un botón para agregar las palabras clave seleccionados a la lista de palabras clave de la pieza.
	2	El administrador del museo selecciona las palabras clave que desea agregar y solicita al sistema que los agregue a la lista de palabras clave de la pieza
	3	El sistema agrega las palabras clave seleccionados a la lista de palabras clave de la pieza.
Poscondición	El administrador del museo continúa realizando el caso de uso RF-02 Modificar pieza.	
Excepciones	Ninguna	

RF-14	Administrar palabras clave
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material
Requisitos asociados	RI-01 Información sobre piezas RF-15 Crear palabra clave. RF-16 Modificar palabra clave. RF-12 Administrar palabras clave de la pieza.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo desee crear nuevas

	palabras clave o modificar las palabras clave existentes	
Precondición	<p>El administrador del museo puede acceder a este caso de uso de dos maneras diferentes:</p> <ol style="list-style-type: none"> 3. cuando se encuentra realizando el caso de uso RF-12 Administrar palabras clave de la pieza. 4. cuando lo Solicita a través del menú principal. 	
Secuencia normal	Paso	Acción
	1	<p>El sistema muestra la lista de las palabras clave existentes. Cada ítem de la lista consta de: Palabra clave en español y en inglés, un botón para seleccionarla y un botón para modificarla.</p> <p>Al final de la lista se muestra un botón para eliminar las palabras clave seleccionadas y un botón para crear una nueva palabra clave.</p>
	2	<p>Si el administrador del museo elige crear una nueva palabra clave, se realiza el caso de uso RF-15 Crear palabra clave y a continuación este caso de uso continúa</p>
Poscondición	<p>El administrador del museo continúa realizando el caso de uso RF-12 Administrar palabras clave de la pieza o vuelve al menú principal, dependiendo de la precondición que lo originó.</p>	
Excepciones	Paso	Acción
	1	<p>Si el administrador del museo elige eliminar, el sistema solicita la confirmación de esta acción. Si el administrador confirma, las palabras clave seleccionadas se eliminarán sólo si no existen piezas que contengan dichas palabras clave.</p>

	1	Si el administrador del museo elige modificar, se realiza el caso de uso RF-16 Modificar palabra clave y a continuación este caso de uso continúa
--	---	---

RF-15	Crear palabra clave	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-01 Información sobre piezas RF-14 Administrar palabras clave	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo solicite la creación de una nueva palabra clave.	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-14 Administrar palabras clave.	
Secuencia normal	Paso	Acción
	1	El sistema solicita los siguientes datos: Palabra clave en español y en inglés.
	2	El administrador del museo proporciona los datos requeridos y solicita al sistema que los almacene
	3	El administrador del museo confirma que los datos suministrados son correctos.
	4	El sistema almacena los datos proporcionados y los agrega a la lista de palabras clave

Poscondición	El administrador del museo continúa realizando el caso de uso RF-14 Administrar palabras clave.	
Excepciones	Paso	Acción
	2	Si el sistema detecta que la nueva palabra clave ya existe, el sistema informa de la situación al administrador del museo permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa
	2	Si el administrador del museo solicita cancelar la operación, el sistema cancela y a continuación este caso de uso termina

RF-16	Modificar palabra clave	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material	
Requisitos asociados	RI-01 Información sobre piezas RF-14 Administrar palabras clave	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del museo solicite la modificación de una palabra clave que se desea corregir	
Precondición	El administrador del museo se encuentra realizando el caso de uso RF-14 Administrar palabras clave.	
Secuencia normal	Paso	Acción
	1	El sistema muestra los datos actuales y solicita los nuevos: palabra clave en español y en inglés.

	2	El administrador del museo proporciona los datos requeridos y solicita al sistema que los almacene
	3	El administrador del museo confirma que los datos suministrados son correctos.
	4	El sistema almacena los datos proporcionados y actualiza la lista de palabras clave
Poscondición	El administrador del museo continúa realizando el caso de uso RF-14 Administrar palabras clave.	
Excepciones	Paso	Acción
	2	Si el sistema detecta que los nuevos datos de la palabra clave corresponden a una palabras clave que ya existe, el sistema informa de la situación al administrador permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa
	1	Si el administrador del museo solicita cancelar la operación, el sistema cancela y a continuación este caso de uso termina

RF-17	Buscar piezas
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material OBJ-02 Publicar el material
Requisitos asociados	RI-01 Información sobre piezas RF-18 Listar piezas.

	RF-20 Buscar por órganos.	
	RF-21 Buscar por palabras clave.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El usuario solicita al sistema comenzar el proceso de búsqueda de piezas.
	2	El sistema solicita al usuario introducir las palabras para realizar la consulta, estas palabras pueden ser: Códigos CIE10, diagnósticos y palabras clave, en el caso de que busque por palabras clave (RF-21). En el caso de la búsqueda ramificada (RF-20) se supone que el usuario ha especificado el diagnóstico que se usará como criterio de búsqueda.
	3	El usuario proporciona los datos requeridos y solicita al sistema que busque todas las piezas que cumplan con uno o mas de las palabras especificadas
	4	Se realiza el caso de uso RF-18 Listar piezas
Poscondición	El usuario continúa realizando este caso de uso.	
Excepciones	Paso	Acción
	1	Si el usuario ingresa al sistema se inicia automáticamente el programa de búsqueda de piezas porque es el caso de uso predeterminado para el inicio del sistema.

	3	Si el sistema no encuentra ninguna pieza que cumpla con los parámetros de búsqueda, el sistema informa de la situación al usuario, a continuación este caso de uso continúa
--	---	---

RF-18	Listar piezas	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material OBJ-02 Publicar el material	
Requisitos asociados	RI-01 Información sobre piezas RF-17 Buscar piezas RF-19 Ver detalles de la pieza.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
Precondición	El sistema encontró piezas que cumplen con los parámetros de búsqueda especificados por el usuario	
Secuencia normal	Paso	Acción
	1	El sistema muestra una lista de piezas. Cada ítem de la lista costa de: Código de la pieza, órgano, nombre del diagnóstico principal y resumen de la descripción macroscópica, adicionalmente habrá un botón para ver los detalles de la pieza. Los parámetros para realizar el ordenamiento son: 1. Mayor número de coincidencias de parámetros de búsqueda 2. Orden descendente en el código CIE-10.

	2	Si la lista de resultados es muy larga se muestran los resultados en varias páginas
Poscondición	El usuario continúa realizando el caso de uso RF-17 Buscar piezas.	
Excepciones	Paso	Acción
	1	Existe la opción de mostrar una vista de miniaturas de las piezas que coincidan con el criterio de búsqueda. La miniatura de cada pieza se obtendrá de la primera fotografía asociada a dicha pieza. Debajo de cada miniatura estará el nombre del órgano y el diagnóstico principal. Además se suministra un botón para ver los detalles de la pieza (RF-19).

RF-19	Ver detalles de la pieza
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material OBJ-02 Publicar el material
Requisitos asociados	RI-01 Información sobre piezas RF-18 Listar piezas. RF-02 Modificar pieza. RF-22 Administrar bloques. RF-30 Ver detalles de la práctica. RF-33 Agregar pieza a una práctica.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso

Precondición	El usuario se encuentra realizando el caso de uso RF-18 listar piezas.	
Secuencia normal	Paso	Acción
	1	<p>El sistema muestra los siguientes datos:</p> <p>Nombre del órgano</p> <p>CIE10 y nombre del diagnóstico principal</p> <p>CIE10 y nombre de los diagnósticos secundarios</p> <p>Descripción macroscópica</p> <p>Lista de fotos macro de la pieza con su descripción</p> <p>Lista de fotos micro de la pieza con su descripción</p> <p>Lista de palabras clave de la pieza</p> <p>Resumen de Historia Clínica con su lista de fotos.</p> <p>Además se muestra un botón para regresar a la lista de piezas</p>
Poscondición	Ninguna.	
Excepciones	Paso	Acción
	1	<p>Si el usuario es un administrador del museo, se muestran controles para acceder a diferentes labores de administración de la pieza así:</p> <p>Si elige “modificar” se realiza el caso de uso RF-02 Modificar pieza.</p> <p>Si elige “administrar bloques” se realiza el caso de uso RF-22 Administrar Bloques.</p> <p>Si elige “Agregar pieza a una práctica” se realiza el caso de uso RF-33.</p>

RF-20	Buscar por órganos	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material. OBJ-02 Publicar el material.	
Requisitos asociados	RI-01 Información sobre piezas. RF-17 Buscar piezas.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario desea buscar una determinada pieza en el sistema, comenzando de forma ramificada desde el órgano asociado a dicha pieza.	
Precondición	El usuario se encuentra en la página de consulta ramificada de piezas.	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario los órganos para los cuales hay al menos una pieza asociada.
	2	El usuario selecciona el órgano de su interés.
	3	El sistema muestra los diagnósticos, relevantes al órgano seleccionado por el usuario, asociados al menos a una pieza.
	4	El usuario selecciona el diagnóstico de su interés
	5	El sistema realiza el caso de uso RF-17 para encontrar las piezas asociadas al diagnóstico elegido y a continuación este caso de uso termina,
Poscondición	El usuario observa una lista de piezas que concuerdan con su	

	búsqueda, producto de la ejecución del caso de uso RF-17	
Excepciones	Paso	Acción
	1	Si el usuario accede a cualquier otra función del sistema, este caso de uso termina.

RF-21	Buscar por palabras clave	
Versión	0.1	
Objetivos asociados	OBJ-01 Administrar el material. OBJ-02 Publicar el material.	
Requisitos asociados	RI-01 Información sobre piezas RF-17 Buscar piezas.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera consultar las piezas disponibles para un conjunto de palabras que el suministra al sistema, de manera análoga a las páginas de búsqueda en Internet.	
Precondición	El usuario ingresa al programa de consulta de material.	
Secuencia normal	Paso	Acción
	1	El sistema muestra una caja de texto para que el usuario introduzca sus datos y un botón para comenzar la consulta.

	2	El sistema busca todas las posibles coincidencias de las palabras suministradas por el usuario con la información asociada a las piezas almacenadas en el sistema. Dichas coincidencias pueden ser palabras completas, partes de una o más palabras, sin importar mayúsculas y minúsculas.
	3	El sistema ejecuta el caso de uso RF-17 para hacer efectiva la consulta y mostrar al usuario los resultados. A continuación este caso de uso termina.
Poscondición	El usuario observa los resultados de su consulta.	
Excepciones	Paso	Acción
	1	Si el usuario ordena una consulta vacía o que contiene código malicioso que podría afectar al sistema, este caso de uso elimina el código malicioso en lo posible y continúa; o aborta este caso de uso.

RF-22	Administrar bloques de resina
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material. OBJ-03 Mantener el inventario físico.
Requisitos asociados	RI-03 Información sobre bloques de resina. RF-19 ver detalles de la pieza. RF-23 Crear bloque de resina. RF-24 Imprimir rótulo del bloque.

	RF-25 Modificar bloque de resina.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario desea efectuar labores de administración de los bloques de resina en donde están almacenados los ejemplares de una pieza determinada.	
Precondición	El usuario se encuentra realizando el caso de uso RF-19 Ver detalles de la pieza, en donde el sistema da la posibilidad de activar este caso de uso (por ejemplo, con un botón).	
Secuencia normal	Paso	Acción
	1	El usuario, luego de activar este caso de uso, observa un listado de los bloques disponibles para la pieza.
	2	Para cada ítem de la lista, es decir un bloque de la pieza, se muestra un botón para activar los casos de uso RF-24 Imprimir rótulo del bloque y RF-25 Modificar bloque de resina. Adicionalmente se muestra un botón para entrar al caso de uso RF-23 Crear bloque de resina.
	3	Este caso de uso termina.
Poscondición	El usuario entra a uno de estos casos de uso: RF-23 Crear bloque de resina, RF-24 Imprimir rótulo del bloque, RF-25 Modificar bloque de resina.	
Excepciones	Paso	Acción
	1	Si el usuario activa cualquier otra función del sistema, este caso de uso termina.

RF-23	Crear bloque de resina.	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material. OBJ-03 Mantener el inventario físico	
Requisitos asociados	RI-03 Información sobre bloques de resina. RF-22 Administrar bloques de resina. RF-24 Imprimir rótulo del bloque.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario desea registrar en el sistema un nuevo bloque de resina para una pieza determinada.	
Precondición	El usuario se encuentra realizando el caso de uso RF-22 Administrar bloques de resina, y ha activado este caso de uso.	
Secuencia normal	Paso	Acción
	1	El sistema pregunta al usuario si realmente desea agregar un nuevo bloque a la pieza.
	2	El usuario confirma su deseo de registrar un bloque nuevo.
	3	El sistema registra el bloque y ejecuta RF-24 para que el usuario imprima el rótulo que identificará al nuevo bloque dentro de las instalaciones del museo.
	4	El sistema muestra al usuario el resultado de su operación y a continuación este caso de uso termina, una vez que RF-24 (Imprimir rótulo del bloque) ha concluido.
Poscondición	El usuario se encuentra en el caso de uso RF-24	

Excepciones	Paso	Acción
	3	Si el usuario no confirma la operación, este caso de uso termina.

RF-24	Imprimir rótulo del bloque	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material. OBJ-03 Mantener el inventario físico	
Requisitos asociados	RI-03 Información sobre bloques de resina. RF-23 Crear bloque. RF-22 Administrar bloques de resina.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario debe imprimir el rótulo para un bloque de resina nuevo, o si desea imprimir uno de reemplazo para un bloque de resina cuyo rótulo esté deteriorado.	
Precondición	El usuario se encuentra en el caso de uso RF-22 Administrar bloques, y habiendo seleccionado uno de ellos activa este caso de uso; o en el caso de uso RF-23 Crear bloque de resina, el sistema le solicita que procesa con la impresión del rótulo.	
Secuencia normal	Paso	Acción
	1	El sistema muestra una nueva ventana del explorador con la imagen del rótulo del bloque.

	2	El usuario ordena al navegador de Internet que imprima el rótulo generado por el sistema.
	3	El usuario pega el rótulo impreso al bloque.
Poscondición	El rótulo impreso queda adherido al bloque y el usuario vuelve al caso de uso RF-22.	
Excepciones	Paso	Acción
	1	El usuario puede no imprimir el bloque, así el sistema lo haya generado. Como el sistema no tiene potestad para manejar el sistema de impresión, tal tarea es responsabilidad exclusiva de la persona encargada.

RF-25	Modificar bloque de resina
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material. OBJ-03 Mantener el inventario físico
Requisitos asociados	RI-03 Información sobre bloques de resina. RF-22 Administrar bloques de resina.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera cambiar el estado en el cual se encuentra un determinado bloque de resina o si desea registrarlo como desechado.
Precondición	El usuario está en el caso de uso RF-22 Administrar bloques de resina y decide ejecutar este caso de uso, habiendo seleccionado un bloque de

	la lista de bloques mostrada por dicho caso de uso (RF-22).	
Secuencia normal	Paso	Acción
	1	El sistema muestra la información actual del bloque seleccionado.
	2	El usuario cambia la información del bloque (concretamente el estado del mismo) según prefiera.
	3	El sistema pregunta al usuario si realmente desea cambiar los datos de dicho bloque
	4	El usuario confirma la operación de cambio de estado del bloque.
	5	El sistema cambia el estado del bloque y actualiza la fecha de modificación del mismo a la fecha actual. Posteriormente el sistema muestra el resultado de la operación al usuario. A continuación, este caso de uso termina.
Poscondición	El usuario regresa al caso de uso RF-22 Administrar bloques.	
Excepciones	Paso	Acción
	3	Si el usuario no da su confirmación de seguir con la operación, activa cualquier otra función del sistema, este caso de uso termina.
	2	Si el usuario no es un administrador del museo, no podrá marcar un bloque como "Eliminado".
	2	Si el estado actual del bloque es "Eliminado", este caso de uso muestra un mensaje de error y termina.

RF-26	Prestar bloque de resina	
Versión	0.1	
Objetivos asociados	<p>OBJ-01 Administrar el material.</p> <p>OBJ-02 Publicar el material.</p> <p>OBJ-03 Mantener el inventario físico</p>	
Requisitos asociados	RI-03 Información sobre bloques de resina.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera registrar en el sistema la salida de un bloque de resina de las instalaciones del museo para su uso en el aula de clase.	
Precondición	El usuario ingresa al programa de préstamo, tiene el bloque de resina a su alcance y observa el rótulo respectivo para dicho bloque.	
Secuencia normal	Paso	Acción
	1	El sistema muestra una caja de texto para digitar el código del bloque, un botón para registrar los valores que introduce el usuario, y un botón para confirmar la salida del bloque.
	2	El usuario introduce el código del bloque, ya sea mediante el teclado o deslizando el lector de código de barras sobre el rótulo del bloque, y pulsa (sólo si usa el teclado) el botón para registrar el código digitado.
	3	El sistema pregunta al usuario si desea registrar el bloque como prestado.
	4	El usuario confirma la salida del bloque pulsando el botón respectivo.

	5	El sistema procesa la petición del usuario, notifica al usuario del éxito de la operación y espera que otro bloque sea procesado, regresando al paso número uno.
Poscondición	Este caso de uso vuelve a comenzar desde el paso 1.	
Excepciones	Paso	Acción
	-	Si en cualquier momento el usuario activa otra función del sistema, este caso de uso termina.
	3	Si el sistema detecta un error en el código del bloque, o si el bloque no posee un estado de "Guardado", se genera un mensaje de error y este caso de uso vuelve a comenzar.
	4	Si el usuario no da su confirmación, este caso de uso comienza de nuevo en el paso número uno.

RF-27	Guardar bloque de resina
Versión	0.1
Objetivos asociados	OBJ-01 Administrar el material. OBJ-02 Publicar el material. OBJ-03 Mantener el inventario físico
Requisitos asociados	RI-03 Información sobre bloques de resina.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera registrar en el sistema la entrada de un bloque de resina desde el aula de clase hacia las instalaciones del museo.

Precondición	El usuario ingresa al programa de entrega, tiene el bloque de resina a su alcance y observa el rótulo respectivo para dicho bloque.	
Secuencia normal	Paso	Acción
	1	El sistema muestra una caja de texto para digitar el código del bloque, un botón para registrar los valores que introduce el usuario, y un botón para confirmar la entrada del bloque.
	2	El usuario introduce el código del bloque, ya sea mediante el teclado o deslizando el lector de código de barras sobre el rótulo del bloque, y pulsa (sólo si usa el teclado) el botón para registrar el código digitado.
	3	El sistema pregunta al usuario si desea registrar el bloque como guardado.
	4	El usuario confirma la entrada del bloque pulsando el botón respectivo.
	5	El sistema procesa la petición del usuario, notifica al usuario del éxito de la operación y espera que otro bloque sea procesado, regresando al paso número uno.
Poscondición	Este caso de uso vuelve a comenzar desde el paso 1.	
Excepciones	Paso	Acción
	-	Si en cualquier momento el usuario activa otra función del sistema, este caso de uso termina.

	-	Si el usuario detecta que el bloque ha sido averiado o si fue extraviado, debe cancelar inmediatamente este caso de uso y notificar al administrador del museo. Ello implica que este caso de uso termina.
	3	Si el sistema detecta un error en el código del bloque, o si el bloque no posee un estado de "Prestado", se genera un mensaje de error y este caso de uso vuelve a comenzar.
	4	Si el usuario no da su confirmación, este caso de uso comienza de nuevo en el paso número uno.

RF-28	Generar reporte de inventario	
Versión	0.1	
Objetivos asociados	OBJ-01 Administrar el material. OBJ-03 Mantener el inventario físico	
Requisitos asociados	RI-03 Información sobre bloques de resina.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiera que el sistema muestre en detalle estadísticas sobre el estado del inventario del museo.	
Precondición	El usuario ingresa al programa de generación de reportes de inventario.	
Secuencia normal	Paso	Acción

	1	<p>El sistema muestra al usuario los diferentes criterios que pueden ser utilizados para generar el reporte, dichos criterios son:</p> <ul style="list-style-type: none"> • Fecha de ingreso del bloque. • Fecha de última modificación. • Estado del bloque.
	2	<p>El usuario da al sistema los criterios que desea para elaborar el reporte, puede escogerse cualquier combinación de criterios y escoger libremente el intervalo de fechas que se tendrán en cuenta. (Las fechas según el calendario Gregoriano)</p>
	3	<p>El sistema genera el reporte pedido por el usuario.</p>
	4	<p>El sistema muestra al usuario el reporte generado; dicho reporte contendrá la siguiente información:</p> <ul style="list-style-type: none"> • Número de bloques guardados en inventario. • Número de bloques prestados. • Número de bloques en mantenimiento. • Número de bloques desechados. • Número de bloques averiados. • Número de bloques extraviados.

		Adicionalmente el sistema muestra una lista de los bloques que cumplen con los criterios de generación del reporte. Cada ítem de la lista tendrá el código del bloque, su estado actual, la fecha de ingreso y la fecha de la última modificación. Por último pregunta al usuario si desea imprimir el informe.
	5	Si el usuario decide imprimir el informe, el sistema muestra en una ventana nueva una versión imprimible del informe terminando este caso de uso. Si el usuario no desea imprimir el reporte, este caso de uso termina.
Poscondición	Este caso de uso vuelve a comenzar desde el paso 1.	
Excepciones	Paso	Acción
	2	Si no se da ningún criterio al sistema, se muestra un mensaje de error y se vuelve al paso número 1.

RF-29	Modificar práctica
Versión	1.0
Objetivos asociados	OBJ-02 Publicar el material. OBJ-05 Administrar las prácticas de clase.
Requisitos asociados	RI-05 Información sobre prácticas. RF-30 Ver detalles de la práctica.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiere cambiar la información de una

	práctica en el sistema.	
Precondición	El usuario se encuentra en el caso de uso RF-30 Ver detalles de la práctica.	
Secuencia normal	Paso	Acción
	1	El sistema muestra la información actual de la práctica, junto con una interfaz gráfica que permita la introducción de los nuevos datos de la misma. Nota: Esta interfaz permite el cambio de toda la información de la práctica, excepto por agregar piezas a la misma (ver RF-33 Agregar pieza a una práctica). Remover piezas de la práctica si es posible en este caso de uso.
	2	El usuario diligencia el formulario y envía los datos al sistema.
	3	El sistema pregunta si los datos suministrados son correctos.
	4	El usuario confirma que los datos son correctos.
	5	El sistema actualiza la información de la práctica y notifica al usuario del éxito de la operación. A continuación este caso de uso termina.
Poscondición	El usuario observa la pantalla generada por el sistema en donde se notifica el estado de la modificación de la práctica.	
Excepciones	Paso	Acción
	3	Si el usuario no confirma los datos, este caso de uso vuelve al paso 1. Si decide cancelar la operación, este caso de uso termina.

RF-30	Ver detalles de la práctica	
Versión	1.0	
Objetivos asociados	OBJ-02 Publicar el material. OBJ-05 Administrar las prácticas de clase.	
Requisitos asociados	RI-05 Información sobre prácticas. RF-31 Listar prácticas. RF-29 Modificar práctica. RF-19 Ver detalles de la pieza. RF-32 Despachar práctica.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiere observar la información de una práctica en particular.	
Precondición	El usuario se encuentra viendo una lista de prácticas, es decir en el caso de uso RF-31 Listar prácticas, y ordena al sistema ver los detalles de una de las prácticas que conforman la lista.	
Secuencia normal	Paso	Acción
	1	El sistema muestra toda la información disponible sobre la práctica, junto con un listado de piezas, similar al mostrado por el caso de uso RF-18 Listar piezas, que están asociadas a la misma. Este caso de uso termina
Poscondición	El usuario observa los detalles de la práctica de su interés-	
Excepciones	Paso	Acción

	1	Este caso de uso muestra controles adicionales a la información de la práctica para ejecutar los casos de uso RF-29 Modificar Práctica, RF-19 Ver detalles de la pieza y RF-32 Despechar práctica, según el tipo de usuario que esté ejecutando este caso de uso.
--	---	---

RF-31	Listar prácticas	
Versión	1.0	
Objetivos asociados	OBJ-02 Publicar el material. OBJ-05 Administrar las prácticas de clase.	
Requisitos asociados	RI-05 Información sobre prácticas. RF-30 Ver detalles de la práctica.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiere observar la lista de prácticas que el personal docente del departamento de Patología va a realizar durante el periodo académico.	
Precondición	El usuario se ordena al sistema listar las prácticas disponibles.	
Secuencia normal	Paso	Acción
	1	El sistema muestra la lista de las prácticas disponibles en el sistema, para cada práctica de la lista se muestra un control para ver los detalles de dicha práctica.
Poscondición	El usuario observa la lista de prácticas disponibles.	

RF-32	Despachar práctica	
Versión	1.0	
Objetivos asociados	OBJ-01 Administrar el material.	
Requisitos asociados	RI-05 Información sobre prácticas. RI-03 Información sobre bloques de resina., RF-30 Ver detalles de la práctica.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiere registrar la salida/entrada de los bloques de resina que serán usados durante la práctica en el aula de clase	
Precondición	El usuario se encuentra en el caso de uso RF-30 Ver detalles de la práctica y tiene a mano los bloques cuya salida o entrada desea registrar.	
Secuencia normal	Paso	Acción
	1	El sistema muestra un listado de los bloques de resina que contienen ejemplares de las piezas asociadas a la práctica, junto con una caja de texto para que el usuario digite el código de cada bloque que vaya despachando y un botón para que el sistema procese el bloque cuyo código ha sido digitado.
	2	El usuario digita el código del bloque y pulsa el botón, o pasa el lector de código de barras sobre el bloque (en cuyo caso no hay necesidad de pulsar el botón).
	3	El sistema verifica que el bloque tenga el estado de "Prestado" o "Guardado" y cambia el estado del bloque dependiendo de ello.

	4	El sistema coloca en la lista de bloques la nueva información del bloque procesado.
	5	El caso de uso vuelve al paso número 2 hasta que todos los bloques hayan sido digitados o pasados por el lector.
	6	Al terminar de procesar los bloques, el usuario pulsa un botón en donde confirma el despacho de la práctica.
Poscondición	El sistema muestra un aviso al usuario notificándole que la práctica ha sido despachada.	
Excepciones	Paso	Acción
	6	Todos los bloques de la práctica deben quedar con estado Guardado o con Prestado, es decir, no pueden haber bloques prestados y guardados en el momento que el usuario ejecuta el paso 6. Si se da esta situación el sistema genera un mensaje de error y este caso de uso vuelve al paso 1.
	-	Si el usuario cancela la operación o activa otra función del sistema, este caso de uso ignora cualquier cambio realizado y este caso de uso termina.

RF-33	Agregar pieza a una práctica
Versión	1.0
Objetivos asociados	OBJ-01 Administrar el material.
Requisitos asociados	RI-05 Información sobre prácticas. RF-19 Ver detalles de la pieza.

Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiere agregar una pieza a una práctica, de tal forma que los ejemplares de dicha pieza (bloques) estén disponibles para la realización de la práctica.	
Precondición	El usuario se encuentra en el caso de uso RF-19 Ver detalles de la pieza.	
Secuencia normal	Paso	Acción
	1	Una vez que este caso de uso ha sido ejecutado, el sistema muestra un listado de las prácticas disponibles, junto con un botón para seleccionar cada práctica.
	2	El usuario selecciona la práctica a la cual desea agregar la pieza.
	3	El sistema agrega la pieza a la práctica.
	4	El sistema notifica al usuario del éxito de la operación.
Poscondición	El sistema muestra un aviso al usuario notificándole que la pieza ha sido asociada con éxito a la práctica seleccionada.	
Excepciones	Paso	Acción
	-	Si el usuario cancela la operación o activa otra función del sistema, este caso de uso ignora cualquier cambio realizado y este caso de uso termina.

RF-34	Listar usuarios
Versión	1.0

Objetivos asociados	OBJ-04 Administrar usuarios.	
Requisitos asociados	RI-04 Información sobre usuarios autenticados. RF-35 Modificar usuario.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el administrador quiera listar todos los usuarios autenticados que tienen acceso a las funciones privilegiadas del sistema, como administradores, residentes, o técnicos.	
Precondición	El administrador ordena al sistema mostrar los usuarios autenticados, desde el menú correspondiente.	
Secuencia normal	Paso	Acción
	1	Una vez que este caso de uso ha sido ejecutado, el sistema muestra un listado de los usuarios autenticados, junto con controles para modificar los datos de cada usuario o para eliminarlos.
Poscondición	El sistema muestra el listado de usuarios, esperando una nueva orden del administrador.	
Excepciones	Paso	Acción

RF-35	Modificar usuario
Versión	1.0
Objetivos asociados	OBJ-04 Administrar usuarios.
Requisitos asociados	RI-04 Información sobre usuarios autenticados.

	RF-34 Listar usuarios	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el administrador quiera cambiar los datos de un usuario determinado o eliminarlo.	
Precondición	El administrador se encuentra en el caso de uso RF-34 Listar usuarios y selecciona un usuario para cambiar sus datos.	
Secuencia normal	Paso	Acción
	1	El sistema muestra los datos actuales del usuario seleccionado junto con controles que permitan el cambio de dichos datos. Nota: Los controles de este caso de uso impedirán que sea visualizada o cambiada la contraseña de acceso de un usuario determinado. También habrá un control para eliminar el usuario.
	2	El administrador ingresa los nuevos datos y solicita al sistema que actualice la información del usuario. (Excepto si se ha eliminado el usuario)
	3	El sistema pide al administrador que confirme los datos, o la eliminación del usuario.
	4	El administrador confirma y la información del usuario es actualizada.
Poscondición	El sistema muestra el listado de usuarios, esperando una nueva orden del administrador.	
Excepciones	Paso	Acción

	-	Si el administrador decide cancelar este caso de uso, o ejecuta otra función del sistema, este caso de uso ignora los cambios efectuados y termina.
--	---	---

RF-36	Modificar usuario	
Versión	1.0	
Objetivos asociados	OBJ-04 Administrar usuarios.	
Requisitos asociados	RI-04 Información sobre usuarios autenticados. RF-35 Modificar usuario	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el administrador quiera caducar la contraseña de un usuario determinado, en caso que tal usuario haya perdido tal contraseña.	
Precondición	El administrador se encuentra en el caso de uso RF-35 Modificar usuario y ordena que la contraseña de dicho usuario caduque.	
	Paso	Acción
	1	El sistema pregunta al administrador si desea realmente caducar la contraseña del usuario.
	2	El administrador confirma la operación.
	3	El sistema caduca la contraseña y evita la autenticación del usuario afectado, hasta que éste no renueve su clave.

	4	El usuario afectado es notificado de la acción llevada a cabo por el administrador, y en caso que la clave antigua haya sido olvidada por el usuario, debe proveer de un mecanismo de autenticación alterno.
Poscondición	El sistema muestra el listado de usuarios, esperando una nueva orden del administrador.	
Excepciones	Paso	Acción
	-	Si el administrador decide cancelar este caso de uso, o ejecuta otra función del sistema, este caso de uso ignora los cambios efectuados y termina.

RF-37	Agregar usuario	
Versión	1.0	
Objetivos asociados	OBJ-04 Administrar usuarios.	
Requisitos asociados	RI-04 Información sobre usuarios autenticados.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el administrador quiera crear un usuario nuevo que tenga acceso a las funciones privilegiadas del sistema, según su rol dentro del Departamento de Patología.	
Precondición	El administrador ordena al sistema la ejecución de este caso de uso y posee los datos necesarios para la creación del usuario.	
	Paso	Acción
	1	El sistema muestra al administrador un formulario para el diligenciamiento de los datos del nuevo usuario.

	2	El administrador diligencia el formulario y pide al sistema la creación del usuario.
	3	El sistema pide al administrador que confirme la creación del usuario.
	4	El administrador confirma y el sistema crea un nuevo usuario con el rol especificado. El nuevo usuario es notificado.
Poscondición	El sistema muestra el listado de usuarios, esperando una nueva orden del administrador.	
Excepciones	Paso	Acción
	-	Si el administrador decide cancelar o no confirma la operación este caso de uso, o ejecuta otra función del sistema, este caso de uso ignora los cambios efectuados y termina.

RF-38	Cambiar sus datos	
Versión	1.0	
Objetivos asociados	OBJ-04 Administrar usuarios.	
Requisitos asociados	RI-04 Información sobre usuarios autenticados.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario desea cambiar sus datos personales, como su clave.	
Precondición	El usuario ejecuta este caso de uso utilizando los controles apropiados para tal efecto.	
	Paso	Acción

	1	El sistema muestra al usuario sus datos y provee un formulario para el cambio de los mismos. Nota: El nombre de usuario no podrá ser cambiado.
	2	El usuario diligencia el formulario y pide al sistema la actualización de sus datos.
	3	El sistema pide al usuario que confirme la modificación de sus datos.
	4	El usuario confirma y el sistema muestra al usuario el resultado de la operación. Este caso de uso termina
Poscondición	El sistema retorna al estado en el cual el usuario estaba antes de la ejecución de este caso de uso.	
Excepciones	Paso	Acción
	-	Si el usuario decide cancelar o no confirma la operación este caso de uso, o ejecuta otra función del sistema, este caso de uso ignora los cambios efectuados y termina.

RF-39	Autenticarse
Versión	1.0
Objetivos asociados	OBJ-04 Administrar usuarios.
Requisitos asociados	RI-04 Información sobre usuarios autenticados.
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando cualquier usuario autentique su identidad ante el sistema para de este modo poder acceder a las funciones privilegiadas

	de éste.	
Precondición	El usuario intenta ejecutar un caso de uso que requiere la verificación de su identidad.	
Secuencia normal	Paso	Acción
	1	Una vez que este caso de uso ha sido ejecutado, el sistema muestra un formulario para que el usuario digite su nombre y su contraseña
	2	Si la contraseña es correcta, el sistema deja que el usuario acceda a los servicios a los cuales tiene acceso, según su rol.
Poscondición	El sistema deja al usuario en una página de bienvenida, y con acceso a las funciones propias de su rol dentro del sistema.	
Excepciones	Paso	Acción
	2	Si la contraseña es errónea, este caso de uso regresa al paso 1. Si el usuario no introduce la contraseña correcta, por tres veces consecutivas, este caso de uso termina.

RF-40	Enviar sugerencia
Versión	1.0
Objetivos asociados	OBJ-04 Administrar usuarios. OBJ-07 Proporcionar ayuda.
Requisitos asociados	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando cualquier usuario (autenticado o visitante) quiera

	dejar una sugerencia a los administradores.	
Precondición	El usuario entra al programa de envío de opiniones y sugerencias.	
Secuencia normal	Paso	Acción
	1	El sistema muestra un formulario en donde el usuario puede digitar el asunto y el contenido de la sugerencia.
	2	El sistema recoge la sugerencia y la envía por correo electrónico a los administradores.
Poscondición	El sistema informa al usuario que la sugerencia ha sido enviada.	
Excepciones	Paso	Acción

RF-41	Instalar	
Versión	1.0	
Objetivos asociados		
Requisitos asociados	RI-04 Información sobre usuarios autenticados.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando un empleado calificado instala el sistema para su funcionamiento.	
Precondición	La persona que va a efectuar la instalación posee una copia legítima de SIMUPAT y posee los recursos para instalarlo.	
Secuencia normal	Paso	Acción

	1	El usuario descomprime los archivos del sistema en un directorio accesible a la Web.
	2	El sistema pregunta si se está de acuerdo con los términos de la licencia de uso.
	3	El usuario acepta los términos y continúa la instalación
	4	El sistema pregunta al usuario, mediante un formulario, los datos necesarios para establecer conexión con un manejadote de base de datos.
	5	El usuario diligencia el formulario y el sistema intenta conectarse con el DBMS.
	6	El sistema crea las tablas necesarias en la base de datos para su funcionamiento.
	7	El sistema pregunta al usuario si se va a efectuar una instalación nueva o si se va a recuperar de una copia de seguridad
	8	Si el usuario escoge instalación nueva, se inicializan las tablas creadas y se avanza al paso 10.
	9	Si se escoge recuperar de una copia de seguridad, el sistema solicita los archivos que conforman dicha copia y restaura el contenido de la base de datos.
	10	El sistema solicita datos para la creación del usuario Administrador del Sistema.
	11	El usuario introduce los datos y el usuario Administrador es creado.

Poscondición	El sistema muestra al usuario que el proceso de instalación ha terminado.	
Excepciones	Paso	Acción
	-	Si hay un error en la instalación el sistema aborta este caso de uso y debe instalarse el sistema de nuevo.

RF-42	Realizar copia de seguridad	
Versión	1.0	
Objetivos asociados		
Requisitos asociados		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando un administrador desee crear una copia de seguridad.	
Precondición	El usuario entra al programa de creación de copia de seguridad y ha copiado el material fotográfico en un sitio seguro.	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario un botón para iniciar la construcción del archivo de copia de seguridad
	2	El usuario da inicio al proceso.
	3	El sistema construye el archivo y notifica al usuario para que lo descargue.
	4	El usuario descarga el archivo y este caso de uso termina.
Poscondición	El sistema informa al usuario que se ha terminado la copia de	

	seguridad.	
Excepciones	Paso	Acción
	-	Si hay un error, este caso de uso termina y no se genera ningún archivo de copia de seguridad.

RF-43	Crear práctica	
Versión	1.0	
Objetivos asociados	OBJ-02 Publicar el material. OBJ-05 Administrar las prácticas de clase.	
Requisitos asociados	RI-05 Información sobre prácticas.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario quiere crear una práctica nueva en el sistema.	
Precondición	El usuario solicita al sistema la entrada al programa de creación de prácticas.	
Secuencia normal	Paso	Acción
	1	El sistema muestra una interfaz gráfica que permita recoger la información asociada a la práctica.
	2	El usuario diligencia el formulario y envía los datos al sistema.
	3	El sistema pregunta si los datos suministrados son correctos.
	4	El usuario confirma que los datos son correctos.

	5	El sistema crea la práctica y notifica al usuario del éxito de la operación. El caso de uso termina.
Poscondición	El usuario observa la pantalla generada por el sistema en donde se notifica el estado de la creación de la práctica.	
Excepciones	Paso	Acción
	3	Si el usuario no confirma los datos, este caso de uso vuelve al paso 1. Si decide cancelar la operación, este caso de uso termina.

7.6 REQUERIMIENTOS NO FUNCIONALES

A continuación se listan todos los requerimientos no funcionales que determinarán la construcción y el diseño del sistema.

7.6.1 Requisitos de la Interfaz de usuario

La interfaz de usuario deberá satisfacer los siguientes requisitos¹:

Fuentes de entrada de datos: SIMUPAT debe permitir acceder a todos los menús del sistema utilizando tanto el mouse como el teclado, en el caso específico del técnico se permitirá la utilización de un lector de código de barras como fuente de ingreso de datos.

Destino de salida de datos: Los destinos primarios de salida de datos serán la pantalla y la impresora, seguidos por cualquier dispositivo de almacenamiento.

Elementos de la interfaz gráfica: Las interacciones se basarán en acciones físicas sobre elementos visuales (iconos, botones, imágenes, mensajes de texto, barras de desplazamiento y navegación) y en selecciones de tipo menú con sintaxis y órdenes.

Consistencia en la presentación: El usuario debe ver la información y los objetos siempre con el mismo aspecto, siguiendo una misma lógica, y ubicados en sitios iguales.

¹ MARCOS MORA, Mari Carmen. Pautas para el diseño y la evaluación de interfaces de usuario.

Consistencia en el comportamiento: se refiere a que los objetos deben actuar siempre de la misma manera y así no sorprender al usuario.

Consistencia en las técnicas de interacción: las técnicas de interacción (las teclas de atajo, la utilidad de los botones del ratón, la regla nemotécnica aplicada a los menús) deben ser las mismas en las distintas pantallas o programas.

Comprensión del texto y los mensajes: Se deberán presentar textos descriptivos y comprensibles al usuario utilizando un lenguaje fácil y lo menos técnico posible. Esto debe aplicarse tanto a los mensajes del sistema, las etiquetas de las opciones y cualquier texto que se presente al usuario.

Navegación guiada: Se debe proveer un contexto sobre la ubicación del usuario, hacia dónde puede dirigirse, de dónde procede y permitirle pasar con rapidez de un programa a otro.

Divulgación progresiva: Se mostrará al usuario sólo aquellas funciones a las que tenga derecho y a medida que las vaya necesitando, no es preciso ni aconsejable mostrar al usuario todas las funciones que ofrece el programa.

Efectos inmediatos: Las operaciones serán rápidas, incrementales, con efectos inmediatos y proporcionarán realimentación visual de las acciones que ejecuta el usuario.

Operaciones reversibles: Toda operación realizada por el usuario podrá ser reversible.

Validación de datos: La herramienta controlará las entradas para evitar el ingreso de valores erróneos o fuera del rango especificado para cada tipo de dato.

Mensajes de error informativos: Los mensajes de error deben ser informativos y constructivos para que el usuario comprenda su equivocación, pueda remediarla en ese momento y no repetirla en adelante.

Visualización de imprimibles: Visualización en pantalla de los objetos tal y como quedarían impresos.

Ayuda: todos los programas deberán contar con un modulo de ayuda en línea donde se encuentren documentadas e ilustradas todas las funciones que contiene.

Ayuda contextual: se proporcionará ayuda referente a la acción que se ejecuta y sin tener que abandonar la pantalla de trabajo.

Presentación ajustable: La presentación debe ajustarse automáticamente a las diferentes resoluciones de pantalla que pueda tener el usuario Web, la mínima resolución de pantalla soportada será de 800*600 y no se necesitara una barra de desplazamiento horizontal para poder visualizarla por completo. La altura de la página será un valor inferior al doble de la altura de la pantalla.

Fotografías: Las fotografías se almacenarán en formato JPG, en resolución máxima de 640*480.

Miniaturas: Las fotografías se visualizarán inicialmente en forma de miniaturas y podrán ampliarse seleccionando dichas miniaturas.

Paginación de resultados: Los resultados de las búsquedas que retornen un listado se mostrarán en varias páginas en caso de ser necesario.

Alarmas: El sistema debe notificar a cada usuario las alarmas relevantes a su rol, en forma de textos de un color apropiado.

Imagen institucional: La interfaz gráfica utilizará los colores institucionales de la universidad.

Errores del sistema: El usuario deberá ser informado acerca de los errores, ocasionados por hardware, software o redes, que detengan la labor que se está ejecutando.

Perfil de usuario: Se debe visualizar en todo momento el perfil del usuario (nombre de usuario y su categoría) que haya iniciado sesión en el sistema.

7.6.2 Requisitos de la interfaz de hardware

Los autores aclaran, que aunque estos requerimientos deben ser tenidos en cuenta a la hora de construir el sistema, una prueba de los mismos requiere una implantación definitiva, lo cual escapa a los objetivos de este trabajo de grado.

Canal interno: Las operaciones o transacciones relevantes al manejo del inventario, se efectuaran en lo posible usando un canal interno de comunicación (La red UIS).

Copia local: Se deberá tener una copia espejo local para respaldar un fallo en el servidor, y facilitar las labores de backup.

7.6.2.1 Puesto de trabajo del usuario

Accesibilidad: El usuario accederá al sistema desde cualquier equipo con acceso a Internet.

Uso de código de barras: El sistema debe proveer soporte para la impresión y lectura de código de barras

7.6.3 Requisitos de la interfaz de software

Protocolo: El sistema presentará sus diferentes estados, mensajes de error, reportes, y demás tipo de información mediante Web, es decir, usando el protocolo HTTP y el lenguaje HTML 4.01.

Prevención de entrada no autorizada por “fuerza bruta”: El sistema evitará que códigos maliciosos entren a las funciones privilegiadas del sistema mediante el uso de algoritmos de fuerza bruta.

Códigos maliciosos y técnicas de inyección de código: La herramienta deberá prevenir inicios de sesión ocasionados por códigos maliciosos remotos y ataques ocasionados por inyección de código malicioso.

Programa de instalación: El sistema debe contar con un programa de instalación que pueda ejecutarse sin mayores cambios en las plataformas Windows y Linux, facilitando la migración física de un servidor a otro, si así se requiere.

Copias de seguridad: El sistema debe proporcionar un programa que realice copias de seguridad del contenido de su base de datos.

Protección de fotografías: Las fotografías de las piezas estarán protegidas por una marca de agua configurable por el administrador del museo.

Privacidad en los datos del usuario: El sistema evitará que la contraseña de acceso de un usuario determinado sea visible para otra persona, incluidos los administradores.

Independencia del explorador: la aplicación debe visualizarse correctamente en los siguientes exploradores: IE versión 5.0 o superior y Mozilla Firefox 4.0.

Sistema manejador de base de datos: El sistema será compatible con diferentes manejadores de base de datos.

7.6.4 Requisitos de la interfaz de comunicaciones

Control de transacciones: El sistema debe poder recuperarse de un fallo grave de comunicación, mediante el uso de control de transacciones.

Comunicación por Internet: El sistema debe comunicarse con los usuarios vía Internet.

7.6.5 Requisitos de rendimiento

Los autores aclaran, que aunque estos requerimientos deben ser tenidos en cuenta a la hora de construir el sistema, una prueba de los mismos requiere una implantación definitiva, lo cual escapa a los objetivos de este trabajo de grado.

Tiempo de respuesta de inventario: El Sistema deberá responder, en un 70% de las oportunidades, en menos de diez segundos para operaciones críticas (Manejo de inventario) y en menos de 20 segundos para operaciones de consulta.

Carga pico: Como mínimo el sistema deberá soportar una carga pico de 150 sesiones concurrentes.

7.6.6 Requisitos de diseño y desarrollo

Lenguaje unificado: Las especificaciones y demás diagramas de diseño deberán hacerse usando UML 2.0

Uso de herramientas de código libre: El sistema será construido usando software libre.

Documentación: Se debe construir documentación clara, a todo nivel, del sistema para facilitar el desarrollo de nuevas funciones, y las labores de mantenimiento.

8 ESPECIFICACIONES DE DISEÑO Y ARQUITECTURA DEL SISTEMA

Debido a que el presente sistema ha sido desarrollado utilizando la metodología del *Proceso unificado de Racional* (RUP, por sus siglas en inglés), es de esperar que esté construido con base en una arquitectura bien definida, ya que la teoría de la Ingeniería del Software nos dice que el RUP es *centrado en la arquitectura*. Teniendo en cuenta lo anterior, los autores de este proyecto han considerado dedicar una sección amplia a la cobertura de este tema tan importante para el entendimiento del presente proyecto.

De igual manera, los autores han creído conveniente explicar cada uno de los componentes de la arquitectura de SIMUPAT, bajo los diferentes puntos de vista, no solo desde la perspectiva de unas especificaciones de diseño sino complementaria a la forma que en que dichas especificaciones han sido implementadas, es decir, a cada aspecto que se relaciona con el diseño del sistema se le complementa explicando como se ha implementado tal aspecto dentro del sistema terminado. Lo anterior facilita, a juicio de los autores, la lectura de esta sección para administradores, desarrolladores, y demás personal interesado en administrar, extender, o construir un sistema similar al expuesto en este libro.

Como colofón, los autores hacen hincapié en la necesidad por parte del lector de conocer la interpretación de los diferentes diagramas UML que aquí se muestran, para una concepción más profunda de lo que se explica textualmente. Aunque los autores explican algunos aspectos del diagrama, no profundizarán en el significado de los mismos, es decir, no explicarán los diversos símbolos mostrados en los diagramas UML ya que suponen que el lector los conoce y los sabe interpretar.

En esta sección se considerará la arquitectura de SIMUPAT bajo los siguientes puntos de vista:

- Despliegue: En esta vista se consideran los diferentes componentes *hardware* del sistema, las relaciones entre ellos y el papel que desempeñan dentro de la aplicación.
- Implementación: En esta vista se consideran los diferentes archivos y recursos *software* que componen el sistema, las relaciones entre ellos, su organización, y el papel que desempeñan dentro de la aplicación.
- Procesos: En esta vista se estudia la topología del sistema considerando los diferentes procesos que se ejecutan como resultado de la interacción entre el usuario y la aplicación.
- Diseño: En esta vista se estudian los diferentes componentes del sistema desde el punto de vista del diseño del software que lo compone, es decir, como están organizadas las diferentes clases, funciones y variables que constituyen la funcionalidad del sistema y que están implementadas en los archivos y componentes descritos en la vista de implementación.

8.1 VISTA DE DESPLIEGUE

Como su nombre lo indica, el presente proyecto ha sido concebido como un Sistema de Información basado en Web, lo cual implica que utiliza Internet, y más específicamente el protocolo HTTP, para la comunicación entre las diferentes partes que lo componen. Bajo el punto de vista que estamos considerando, dichas partes son:

- El servidor: Es un computador que aloja los diferentes archivos en los cuales está implementado el sistema y que ejecuta las instrucciones escritas en los mismos para poder prestar los servicios que ofrece el sistema a los clientes. Dichas instrucciones son ejecutadas y sus resultados servidos al cliente en formato HTML; de la misma manera, las peticiones de los clientes son transmitidas al servidor mediante el protocolo HTTP para su ejecución. El servidor también es el responsable de alojar y gestionar los datos de la aplicación, es decir, de alojar y ejecutar un programa manejador de base de datos (DBMS)
- Los clientes: Los clientes son otros sistemas que se conectan al servidor para solicitar servicios de éste. Para que un computador sea cliente de SIMUPAT es necesario que cuente con una conexión a Internet (o que haya una comunicación HTTP entre el servidor y el cliente,

aunque sea a nivel local) y un explorador Web. El número de clientes concurrentes que pueden conectarse al sistema depende de la capacidad de procesamiento del servidor y de las prestaciones de la red a la cual estén, tanto el cliente como el servidor, conectados.

- El lector de código de barras e impresora: Como una de las características de SIMUPAT es el manejo de códigos de barras para la agilización de las funciones de inventario, un lector de código de barras (opcional) puede ser conectado a una máquina cliente para utilizar las ventajas mencionadas. Este es el caso del computador en donde opera el técnico encargado del Museo, ya que para que su trabajo con los bloques de resina sea más rápido, se le ha sido suministrado un lector de código de barras. La impresora es necesaria si un cliente quiere imprimir informes de inventario o rótulos para los bloques de resina.
- Infraestructura de comunicación: Comprende los diferentes equipos utilizados para hacer posible la comunicación entre los clientes y el servidor. En el caso de SIMUPAT, dicha infraestructura es la necesaria para mantener una conexión a través de Internet, es decir, los enrutadores, switches, repetidores, proxys, y demás equipo involucrado en el funcionamiento de la Web.

Una concepción más específica de la arquitectura de SIMUPAT bajo el punto de vista del despliegue es el diagrama combinado de componentes, que es un híbrido entre el diagrama de componentes y el diagrama de despliegue en UML.

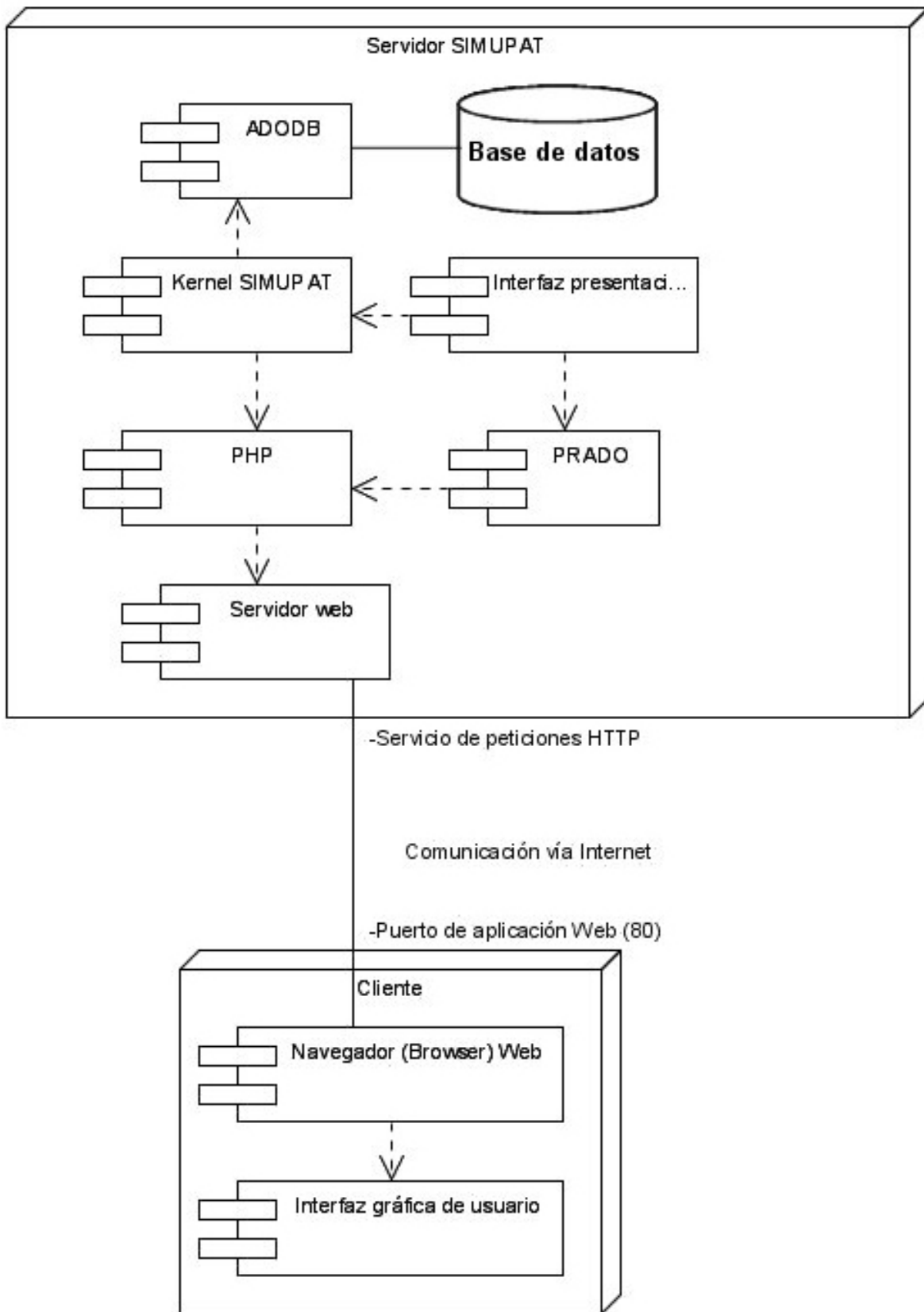


Figura 8.1. Diagrama combinado de componentes y despliegue de SIMUPAT.

Aunque en esta vista hablamos de la infraestructura *hardware* de la arquitectura, los autores han creído conveniente mostrar en el diagrama dicha infraestructura hardware junto con los principales paquetes que conforman SIMUPAT y el software auxiliar, como el manejador de bases de datos y el servidor web, para que el lector tenga una concepción global sobre los componentes que se ejecutan en el servidor y en el cliente. Las flechas punteadas muestran la *dependencia* de dichos paquetes entre sí, no el flujo de los datos dentro del sistema.

8.2 VISTA DE PROCESOS

En este punto se consideran los diferentes procesos que conforman la interacción entre el usuario final y SIMUPAT. Debido a que el sistema utiliza la Web como medio de comunicación entre el usuario y el servidor, es natural que el sistema siga una lógica basada en eventos, es decir, que responda de una u otra forma dependiendo de la acción del usuario sobre la GUI, comportamiento típico de la navegación Web y del software utilizado en la actualidad. Sin embargo, a pesar que HTML brinda un esquema de manejo de eventos, dicho esquema adolece de los siguientes inconvenientes:

- No hay soporte para manejar estados: Como el HTML es un lenguaje de marcado, no un “verdadero” lenguaje de programación, no posee mecanismos para guardar información del estado de la aplicación en un momento determinado, es decir, no hay variables u otro medio para almacenar información de forma dinámica.
- No hay procesamiento de información: EL HTML no posee de estructuras de control u otro medio para controlar un flujo que permita construir un algoritmo que lleve a cabo una u otra tarea. Por lo tanto, el HTML por si sólo únicamente muestra contenido estático.
- Alternativas de respuesta a eventos limitada: HTML sólo provee dos formas de responder a un evento: enviar los datos de un formulario (operación que se suele llamar POST) y redirigir al navegador a otra página HTML (seguimiento de un vínculo o link).

Existen varias soluciones a los problemas mencionados, las cuales se han diseñado de tal forma que sean usadas en combinación con el HTML para la construcción de sitios más robustos. Las soluciones más comunes son:

- Uso de lenguaje de guiones embebidos del lado del cliente (como JavaScript o VBScript)
- Uso de applets para visualización de contenido (como los applets de Java)
- Uso de lenguajes de guiones del lado de servidor (cómo PHP, JSP o ASP)

SIMUPAT hace uso de la primera y la última solución de la siguiente forma: el JavaScript y el HTML como medio de mostrar y capturar los datos hacia y desde del usuario (en efecto, la capa de presentación), y PHP como medio para procesar dichos datos en el servidor (las capas entre la capa de presentación y la capa de datos) y de esa forma escribir el resultado usando HTML para su visualización final por parte del cliente.

Sin embargo, aún cuando una aplicación Web sea fortalecida con el uso de los lenguajes de guiones del lado de servidor (como PHP), el desarrollo de dichas aplicaciones es un proceso complejo, en el cual es fácil mezclar el código de las diferentes capas del sistema, y que a la larga (de no tomarse precauciones) puede volverse muy difícil de mantener y depurar. La razón es que comúnmente el código PHP es mezclado con el código HTML, lo cual crea desorden y confusión al no estar definidas y separadas las diferentes capas del sistema. Si hubiese un daño, por ejemplo, en una función que accede a una base de datos, habría que buscarla a lo largo de todo el código del sistema y arreglarla en todos los archivos en donde aparezca. En un sistema como SIMUPAT, con centenares de funciones y decenas de archivos, esta labor sería en extremo tediosa. Por esta razón los autores han buscado la forma de separar el código responsable de la presentación y el código PHP de tal forma que el mantenimiento del sistema sea más fácil: la solución es el *framework* PRADO.

PRADO no solamente separa la capa de presentación del código PHP, sino que provee una completa API para manejo de eventos ocasionados por el usuario y para generar rápidamente contenido HTML. Dicha API contiene funciones, entre muchas otras, para hacer rejillas, cajas de texto, botones, formularios, plantillas de presentación, etc. Se recomienda al lector mirar la bibliografía del presente documento si desea profundizar en PRADO.

La lógica de eventos de PRADO, y por tanto la lógica de eventos de SIMUPAT, es muy similar a la de otros lenguajes visuales de programación de aplicaciones de escritorio. Cada uno de los eventos del usuario que se quiere manejar es asociado con un método que responde a dicho evento: por ejemplo, si al evento de pulsar el botón X se le asocia la función Y, cada vez que el usuario pulse X, la función Y será ejecutada. El diagrama de secuencia mostrado a continuación, tomado de la documentación de PRADO muestra dicha lógica de eventos con un ejemplo sencillo: un botón que cuando se pulsa hace aparecer la frase “Hello World”.

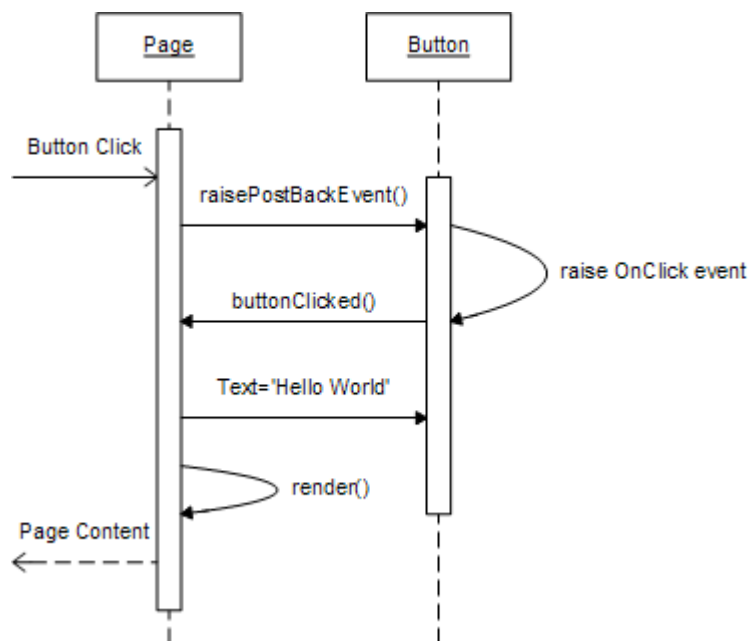


Figura 8.2. Diagrama de secuencia que muestra los eventos en PRADO.

Es la lógica de eventos de PRADO lo que determina en gran medida el flujo de los procesos discutidos en esta sección. No se presentará una explicación detallada de todos los procesos que lleva a cabo el sistema, sino de la forma general en la cual estos procesos son ejecutados.

Aunque los diagramas UML que muestran el proceso que se sigue cuando SIMUPAT procesa una petición del cliente deberían ser suficientes, los autores piensan que es necesaria una explicación en prosa para clarificar cada uno de los pasos involucrados en este proceso. Dicha explicación, escrita en forma de una lista de ítems y tomando como ejemplo el diligenciamiento, procesamiento y almacenamiento en la base de datos de un formulario, es la siguiente:

- En primer lugar el usuario diligencia el formulario, utilizando los controles proveídos por el código HTML de la página.
- El usuario pulsa el botón que envía el formulario (POST)
- El sistema, usando código JavaScript, verifica que la información del formulario cumple con ciertos requisitos tales como: que los campos requeridos hayan sido diligenciados, etc.
- Si la información es válida, el contenido del formulario es enviado vía HTTP hacia el servidor.
- El servidor recibe los datos enviados por el usuario.

Esta primera parte contempla el esquema común en las aplicaciones Web, en donde los datos son transmitidos desde el cliente hasta el servidor. A partir de este momento, PHP toma el control del proceso.

- PRADO dispara el evento ocasionado por el usuario (en este caso pulsar el botón) y llama al método encargado de manejar el evento.
- El método manejador del evento recolecta los datos del usuario, instancia clases de la capa de Reglas del Negocio y transmite a éstas los datos recolectados.

En este punto hemos abandonado la capa de Interfaz con la capa de Presentación y estamos a punto de entrar a la capa de las Reglas del Negocio. En este momento PRADO pierde el control de la aplicación y el *kernel* (Véase la sección 8.4.3) toma las riendas del proceso.

- Las clases de la capa de Reglas del negocio toman los datos y crean los objetos necesarios para encapsular dichos datos. Adicionalmente los métodos de esta clase procesan los datos, de acuerdo a las reglas construidas con base en los resultados de la fase de análisis (Véase la Especificación de Requerimientos de Software en el capítulo siete). Finalmente las clases de la capa Interfaz con la capa de Datos son instanciadas.
- Las clases de la Interfaz con la capa de Datos efectúan la conexión con la base de datos y envían instrucciones SQL al manejador de bases de datos para su ejecución.

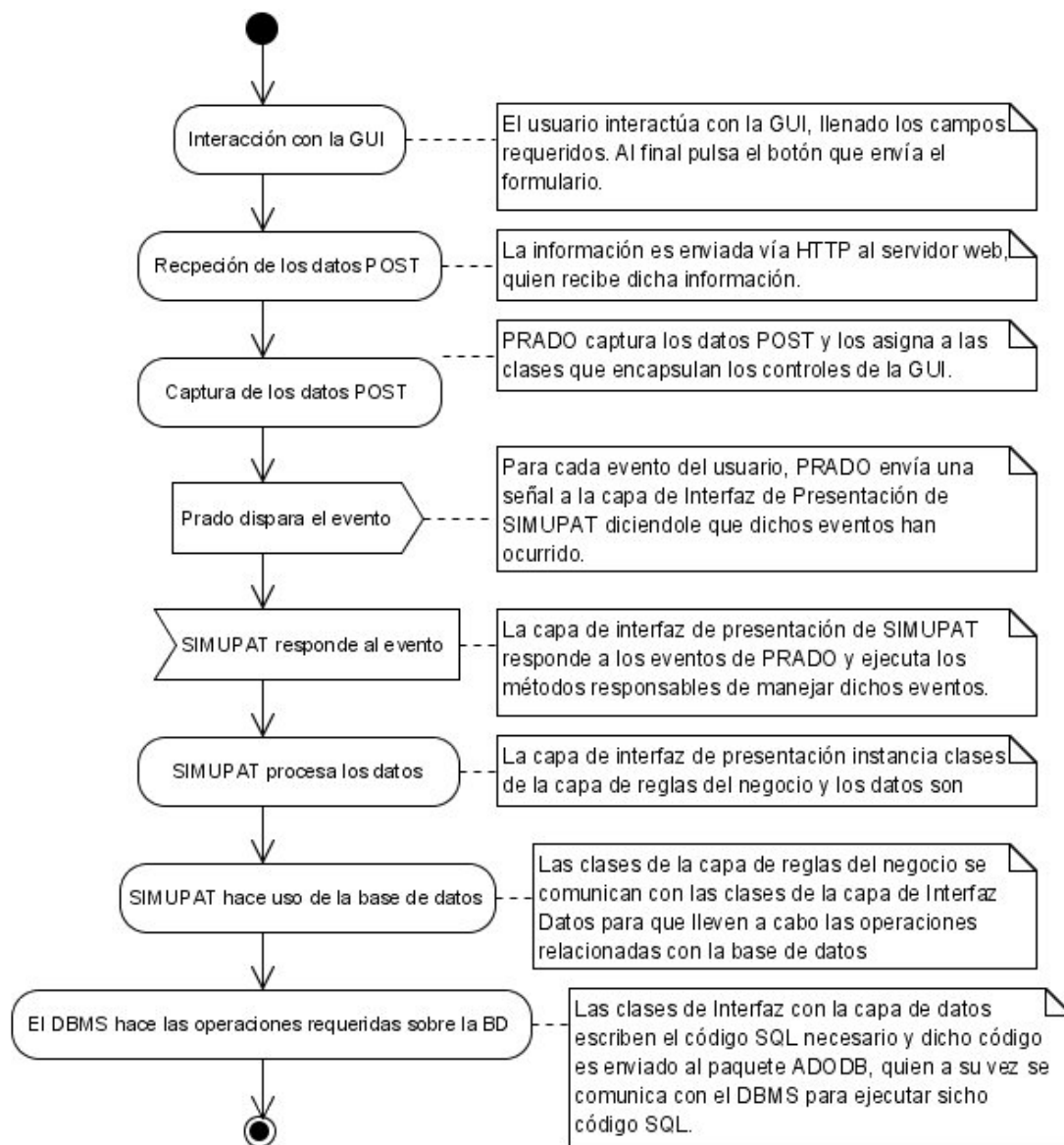


Figura 8.3. Diagrama que actividades que muestra el proceso de respuesta de petición del cliente, desde que el usuario interactúa con el formulario hasta que se utiliza la base de datos.

En este momento SIMUPAT pierde el control del proceso y el DBMS ejecuta las instrucciones SQL necesarias. Luego de las operaciones sobre la base de datos, el DBMS envía información de retorno a las clases de Interfaz con la capa de Datos y SIMUPAT retome de nuevo el hilo del proceso.

- Las clases de la Interfaz con la capa de datos reciben la información de retorno que el DBMS ha suministrado.

- Las clases de la interfaz con la capa de datos devuelven a las clases de Reglas del Negocio la información de retorno.
- Las clases de reglas del Negocio devuelven al método manejador del evento (perteneciente a la capa Interfaz con Presentación) la información sobre la operación.
- El método manejador ejecuta código o invoca otros métodos para que PRADO escriba la salida HTML al usuario.
- PRADO escribe el código HTML de la nueva página de salida al usuario y dispara los eventos relacionados con la carga de la página. Si para estos eventos hay métodos de manejo asociados, dichos métodos son ejecutados.
- El código HTML es enviado al usuario
- El navegador interpreta el HTML y muestra al usuario la página de salida.

Es necesario aclarar que el proceso ha sido descrito de forma simplificada, ya que el funcionamiento de PRADO es mucho más complejo que lo descrito aquí y merece un apartado especial para describirlo con mayor detalle. Asimismo hay que aclarar que no siempre las capas de Interfaz con los Datos o Reglas del Negocio se ejecutan, ello depende de la complejidad de la operación que el usuario esté llevando a cabo. Lo mismo aplica para los diagramas de actividades mostrados, ellos muestran el proceso completo desde la presentación hasta la base de datos, aunque no en todos los casos SIMUPAT lleve a cabo tal proceso como fue descrito, ya que depende de la tarea específica que el sistema esté atendiendo como respuesta a una petición del usuario.

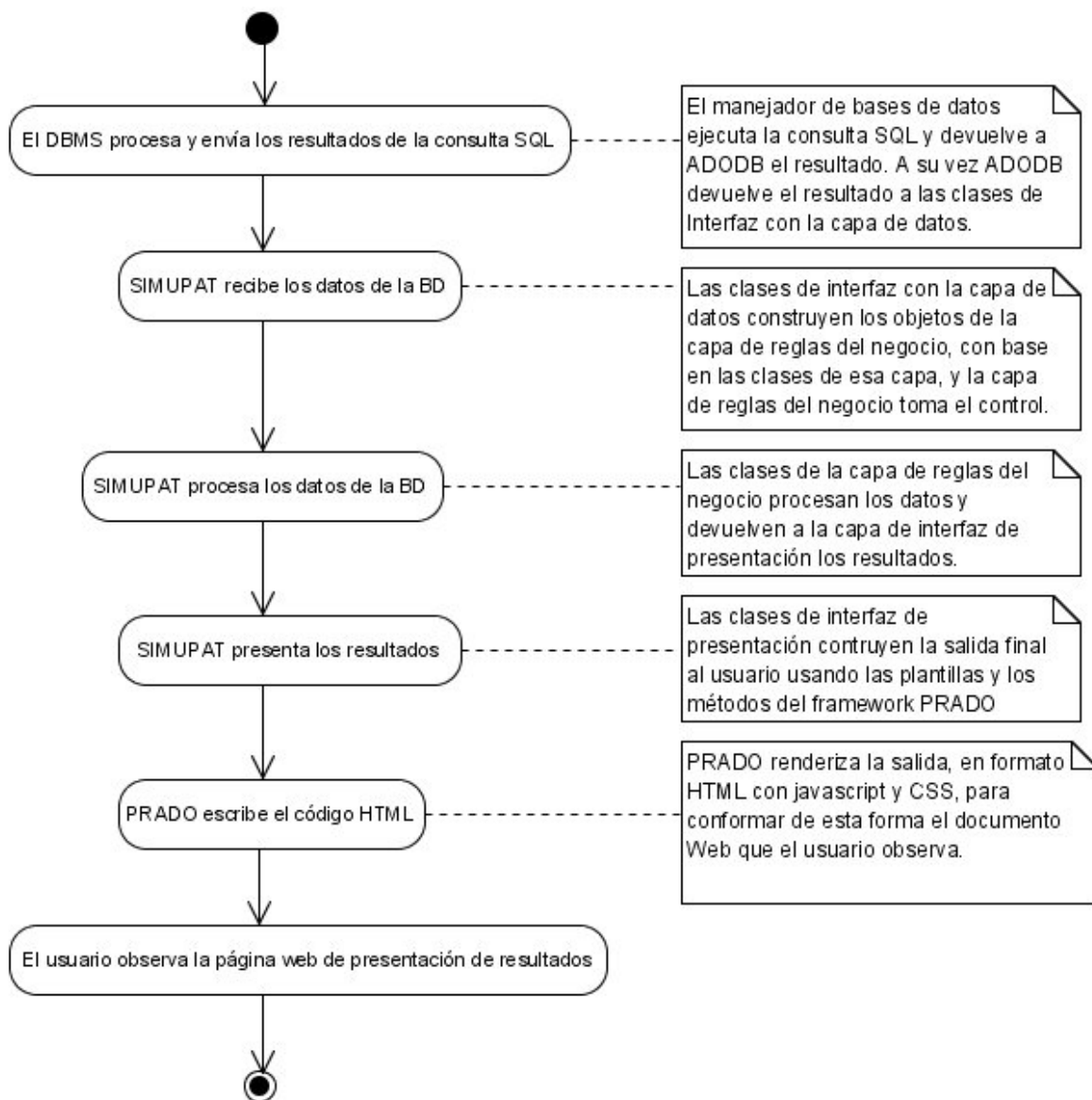


Figura 8.4. Diagrama de actividades que muestra el proceso de salida de resultados hacia el cliente, efectuado por SIMUPAT, desde la consulta en la base de datos hasta la presentación final.

Aunque se ha descrito, en forma general, la secuencia de eventos que ocurren durante una transacción con SIMUPAT, vale la pena profundizar un poco sobre la forma en que los eventos son disparados en PRADO. Para ello, además del diagrama de secuencia mostrado anteriormente, los autores muestran a continuación el ciclo de vida de una página construida usando PRADO. Esta información ha sido tomada, junto con el diagrama, de la documentación del proyecto PRADO.

Como se ve en el diagrama, PRADO inicializa primero sus directorios, carga los módulos definidos, y procesa el archivo de configuración de la aplicación, tras lo cual inicia el proceso de autorización.

Una vez que dicho proceso concluye, carga el estado anterior de la aplicación y procesa los eventos de carga de la página. Finalmente escribe la salida en formato HTML.

Posteriormente, cuando el usuario interactúa con la interfaz gráfica de usuario PRADO dispara, como ya se dijo, los eventos manejadores del evento y salva el estado de la aplicación. Una vez que esto sucede las clases de SIMUPAT realizan el manejo del evento, como se dijo en la secuencia de pasos.

Como colofón a esta sección los autores aclaran que el diagrama de actividades mostrado, en el caso de PRADO¹, ilustra de forma muy general el funcionamiento de dicho framework y que claramente no es suficiente para que el lector desarrolle en PRADO, mas sí lo es para entender la lógica de eventos que este framework maneja. Como una medida auxiliar para que el lector entienda esta lógica de eventos, basta con decir que es muy similar a la lógica de eventos implementada en el framework .NET (y en ASP.NET) para el desarrollo de aplicaciones Web. De igual manera, la forma en que se separa la capa de presentación, de las capas relacionadas con el negocio y los datos es muy similar a la del framework .NET aunque PRADO aún no dispone de un entorno integrado de desarrollo, como sí lo posee ASP.NET.

¹ Tomado de la documentación en formato electrónico de PRADO.

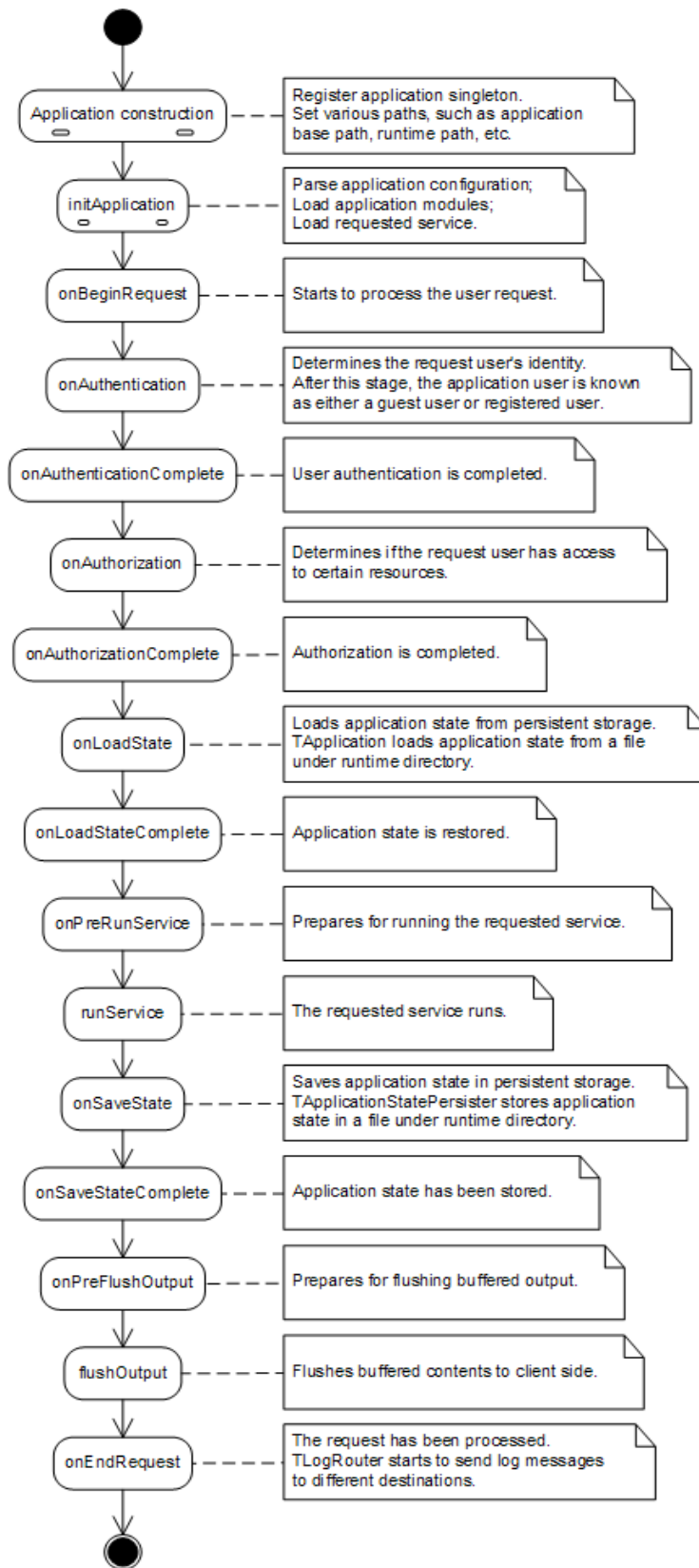


Figura 8.5. Diagrama de actividad del ciclo de vida de una aplicación en PRADO.

8.3 VISTA DE IMPLEMENTACIÓN

El presente sistema está implementado en archivos de texto plano, con codificación UTF-8, los cuales están almacenados en diferentes carpetas ubicadas en la carpeta de trabajo del servidor Web (típicamente “www” o “htdocs”). Estos archivos se dividen, principalmente, en los siguientes tipos:

- Archivos de plantillas: Son archivos que contienen código en XML y HTML (el DTD usado para estos archivos es determinado por PRADO) cuya función es indicar a PRADO como escribir en formato HTML los diferentes marcos o plantillas de las páginas que conforman al sistema. Los archivos de plantillas tienen extensión **.page* o **.tpl*, extensiones determinadas por PRADO.
- Archivos con extensión PHP: Son archivos escritos en el lenguaje PHP que contienen los métodos de respuesta a los eventos disparados por PRADO. Todos los archivos con esta extensión pertenecen a la capa de Interfaz con Presentación y poseen código PHP escrito para trabajar con PRADO. Todos estos archivos tienen código escrito bajo forma de clase, ninguno de ellos tiene código procedural.
- Archivos con extensión INC: Son archivos escritos en lenguaje PHP que contienen la implementación de las capas de Reglas del Negocio y la de Interfaz con la capa de Datos. Estos archivos (excepto por los pertenecientes al paquete utilidades y a los archivos de configuración) no poseen en absoluto código procedural, sino que poseen código escrito bajo forma de clases.
- Archivos de configuración: Son archivos utilizados para definir constantes y otros parámetros que conforman la configuración del sistema. Con excepción de los archivos *constantes.inc*, *config.inc*, *lista_errores.inc*, escritos en PHP, los archivos de configuración tienen extensión XML y han sido escritos usando dicho lenguaje.
- Archivos de hojas de estilos: Son archivos, con extensión **.css* que contienen la definición de las hojas de estilo en cascada usadas por el sistema para la presentación de datos al usuario.

- Archivos de volcado de base de datos: Estos archivos, usualmente con la extensión **.sql*, son generados por el sistema (bajo petición del Administrador del Sistema) como parte de una copia de seguridad de los datos. Estos archivos contienen instrucciones SQL para restaurar el contenido de la base de datos en caso de ser necesario.
- Imágenes: Son los archivos que contienen las fotografías de las piezas almacenadas en el sistema y que conforman la otra parte de la copia de seguridad. Estos archivos son de extensión **.jpg* siempre y están ubicados en una carpeta diferente a la del sistema.

Antes de continuar con el despliegue de los archivos en el sistema, vale la pena introducir la notación en formato de espacio de nombres (namespace), la cual separa las diferentes carpetas de una ruta mediante punto, en vez de slash (/), siendo la raíz el nombre: Aplicación. Será esta notación la utilizada por los autores en esta sección, entendiéndose por Aplicación el directorio en donde SIMUPAT ha sido instalado. Hay que aclarar que aunque en el presente documento los nombres han sido colocados con tildes, los nombres de las carpetas no llevan tilde y están siempre en minúscula. Las carpetas que conforman el sistema son:

- Aplicación (simupat): Es la carpeta principal en donde el sistema ha sido instalado. Contiene el archivo *index.php* que es el único punto de entrada a la aplicación. (Todas las demás subcarpetas están prohibidas para su navegación directa por parte del cliente)
- Aplicación.assets: Esta subcarpeta contiene objetos utilizados para el funcionamiento de PRADO. Es obligatorio que el demonio HTTP tenga permiso para escribir en esta carpeta.
- Aplicación.imágenes: Aquí se guardan algunas imágenes utilizadas para construir la GUI, tales como: íconos, líneas divisorias, logotipos, etc.
- Aplicación.themes: En esta carpeta se guardan los archivos de hojas de estilo utilizados para construir la GUI.
- Aplicación.themes.tema_SIMUPAT: En esta carpeta se guardan las imágenes utilizadas por los archivos de hojas de estilo que definen la GUI de SIMUPAT.

- `Aplicación.protected`: En esta carpeta se guardan los archivos que conforman la implementación del código de la aplicación.
- `Aplicación.protected.controls`: En esta carpeta se guardan las plantillas y el código de los controles PRADO que fueron desarrollados por los autores para el sistema SIMUPAT. Muchos de ellos implementan mejoras al framework PRADO y son una forma muy útil de reutilizar código de presentación.
- `Aplicación.protected.kernel`: Esta carpeta contiene, por así decirlo, el núcleo del sistema. Dentro de esta carpeta están escritas las clases que conforman la capa de Reglas del Negocio y la Interfaz con la capa de Datos. Su nombre, `kernel`, proviene de un término matemático utilizado en el álgebra lineal para denotar el núcleo de una transformación lineal; el término también es usado para notar el componente principal de los sistemas operativos. Este `kernel` es una API completamente independiente de PRADO y ha sido escrita en PHP; para más información sobre como usar esta API, vea el Manual del Programador.
- `Aplicación.protected.kernel.datos`: Esta subcarpeta del `kernel` contiene el objeto que conecta el sistema con el manejador de base de datos. Para más información al respecto vea el Manual del Programador.
- `Aplicación.protected.kernel.excepciones`: Esta subcarpeta del `kernel` contiene las clases encargadas de atrapar los errores disparados por las clases del `kernel` y de imprimir, en pantalla o a un archivo, un trazado inverso para ayudar a los desarrolladores o equipo de mantenimiento a descubrir cual fue la causa del error.
- `Aplicación.protected.kernel.instalación`: Esta subcarpeta del `kernel` contiene clases que preparan al sistema para ser usado por primera vez, de ahí su nombre, las clases de esta carpeta instalan el sistema.
- `Aplicación.protected.kernel.instalación.sql`: Esta carpeta contiene archivos SQL utilizados por las clases de instalación.

- `Aplicación.protected.kernel.interfaz_datos`: Esta subcarpeta del kernel contiene las clases de la capa de Interfaz con la capa de datos, es decir, las clases encargadas de comunicar al DBMS las instrucciones SQL pertinentes.
- `Aplicación.protected.kernel.reglas_negocio`: Esta subcarpeta del kernel contiene las clases de la capa de las reglas del negocio, es decir, en donde los datos son procesados y manejados de acuerdo a los requerimientos funcionales del sistema (Véase la Especificación de Requerimientos de Software)
- `Aplicación.protected.kernel.utilidades`: En esta subcarpeta del kernel se definen diversas funciones para manejo de fechas, cadenas, gráficos, etc.
- `Aplicación.protected.pages`: Esta carpeta, perteneciente a la capa de Interfaz con la capa de Presentación, contiene las clases responsables de manejar los eventos de disparo PRADO como resultado de la interacción con el usuario. El contenido de esta carpeta es totalmente dependiente de PRADO y no funcionará si dicho framework no está instalado en el servidor.
- `Aplicación.protected.pages.administración_piezas`: Esta carpeta contiene las clases que proveen de funcionalidad a las páginas relacionadas con el módulo de Administración de Piezas (véase Objetivos Específicos).
- `Aplicación.protected.pages.administración_piezas.administrar_diagnosticos`: Esta carpeta contiene las clases que proveen de funcionalidad a las páginas relacionadas con la creación, edición, y eliminación de diagnósticos dentro del sistema.
- `Aplicación.protected.pages.administración_piezas.administrar_organos`: Esta carpeta contiene las clases que dan funcionalidad a las páginas relacionadas con la creación, edición y eliminación de órganos dentro del sistema.
- `Aplicación.protected.pages.administración_piezas.administrar_palabras_clave`: En este directorio se almacenan las clases que manejan los eventos de las páginas en donde se crea, edita, o elimina una palabra clave dentro del sistema.

- `Aplicación.protected.pages.administración_sistema`: En este directorio se almacenan las clases que responden a los eventos de las páginas en donde el administrador ordena la creación de copias de seguridad, en forma de archivos de volcado, de los datos del sistema.
- `Aplicación.protected.pages.consulta`: En esta carpeta se guardan las clases responsables de manejar los eventos de las páginas de consulta de piezas, ya sea ramificada, o por búsqueda directa por palabras en el sistema.
- `Aplicación.protected.pages.excepciones`: Este directorio contiene páginas HTML (estáticas) que sirven para sobrescribir las páginas de error del servidor web. Estas páginas informan al usuario sobre los errores: "Página no encontrada", "Error interno del servidor", "En mantenimiento", etc.
- `Aplicación.protected.pages.instalación`: Este directorio contiene las clases que guían al administrador durante el proceso de instalación del sistema (Consulte el Manual del Administrador)
- `Aplicación.protected.pages.inventario`: Este directorio contiene las clases manejadores de los eventos generados en las páginas de administración de inventario, por ejemplo la administración de los bloques de resina, impresión de rótulos con código de barras, etc.
- `Aplicación.protected.pages.usuarios`: Este directorio contiene las clases que responden a los eventos de la administración de usuarios, por ejemplo creación de usuarios, cambio de contraseñas, etc.
- `Aplicación.runtime`: En este directorio PRADO guarda diversos archivos necesarios para su funcionamiento en tiempo de ejecución.



Figura 8.6. Organización de las carpetas de SIMUPAT dentro de la carpeta de documentos Web en el servidor.

8.4 VISTA DE DISEÑO

Finalmente hemos llegado a la que, en opinión de los autores, es la vista que mayor comprensión dará al lector sobre la forma en la cual ha sido construido el sistema. Debido a que ha sido concebido como una aplicación Web, SIMUPAT ha heredado la arquitectura de las tres capas, una forma extendida de la típica arquitectura cliente-servidor.

Aunque existe abundante literatura sobre el tema, los autores consideran necesario definir algunos de los conceptos que se utilizarán en esta sección. La razón es que los diferentes autores dan definiciones diferentes sobre dichos conceptos y es posible entender erróneamente esta sección si no se llega a un consenso sobre los mismos. Tales conceptos son:

- **Módulo:** Un módulo o subsistema es un conjunto de clases, funciones y recursos, que contribuye a realizar un objetivo o funcionalidad del sistema. Por ejemplo, el módulo de administración de piezas comprende desde las páginas de la presentación al usuario, hasta las clases que conectan con la base de datos, en la capa de interfaz con la capa de datos. Como se puede apreciar, un módulo posee componentes en todas las capas de la aplicación.
- **Capa:** Una capa es un conjunto de clases que llevan a cabo una funcionalidad afín, es decir, que realizan labores que proveen una determinada característica al sistema; por ejemplo, las clases de la capa de Interfaz con la capa de Datos hacen que el sistema tenga capacidad de interactuar con una base de datos. Como puede apreciarse, una capa provee la misma característica funcional a todos los módulos del sistema que requieran de dicha característica. Cada capa está conectada con las otras capas del sistema mediante una capa híbrida llamada Interfaz.
- **Paquete:** Un paquete es una agrupación de clases más pequeña que una capa; de hecho, una capa se compone de uno o varios paquetes.

Una vez clarificados los términos, se procede con la descripción de la arquitectura del sistema. Como resultado de la aplicación del diseño orientado a objetos del UML, la arquitectura de SIMUPAT comprende capas bien definidas con clases dentro de ellas, tales capas son:

- **Capa de presentación:** Compuesta por el código HTML y JavaScript que conforman la GUI. Esta capa se ejecuta en el computador cliente, y su código es construido por las capas superiores.
- **Capa de Interfaz de Presentación:** Esta capa se compone por las plantillas y las clases manejadoras de eventos de la GUI. Se compone de varios paquetes, notados con el nombre de la capa, seguido del nombre del módulo al que pertenecen. Las clases de esta capa han sido desarrolladas en PHP 5.0.4 para su uso con PRADO y dependen de dicho framework para su funcionamiento, por lo que se ejecutan en el servidor. Son estas clases las que determinan el comportamiento general de la GUI ante las acciones del usuario final.

- Capa de reglas del negocio: Esta capa es la que contiene toda la lógica del funcionamiento del museo de patología, según lo consignado en la Especificación de requerimientos de software. Esta capa no posee sino un solo paquete, que tiene el mismo nombre de la capa, y posee una clase para cada entidad involucrada en la lógica del museo.
- Capa de interfaz con la capa de datos: Esta capa contiene clases para la interacción con la base de datos. Específicamente, las clases pertenecientes a esta capa utilizan otra capa de abstracción, llamada ADODB, para comunicarse con la base de datos (MySQL actualmente). ADODB es una API de acceso a datos con licencia de código libre que ha sido empleada por los autores del presente proyecto.
- Capa de seguridad: Esta capa contiene la implementación de la seguridad del sistema para evitar ataques a la integridad del sistema.
- Capa de utilidades: Esta capa contiene funciones de utilidades que llevan a cabo tareas diversas como comparar fechas, objetos, manejo de vectores, etc.
- Capa de datos: Conformada por el manejador de bases de datos, el paquete ADODB y la base de datos en si.
- Capa de configuración: Contiene las definiciones de todos los valores parametrizables del sistema, así como las diferentes configuraciones bajo las cuales puede funcionar SIMUPAT.

Son estas cuatro capas principales, las cuales junto con PRADO y ADODB, conforman el conjunto de clases que hacen posible el funcionamiento de SIMUPAT. En el diagrama de paquetes mostrado a continuación, se muestra claramente la estructura general de las capas en SIMUPAT y la dependencia entre las diferentes capas que conforman el sistema. Como todos los diagramas expuestos en esta sección, el diagrama de paquetes está escrito en UML 2.0 y su propósito es brindar al lector una visión global de la arquitectura, desde el punto de vista de diseño, del sistema.

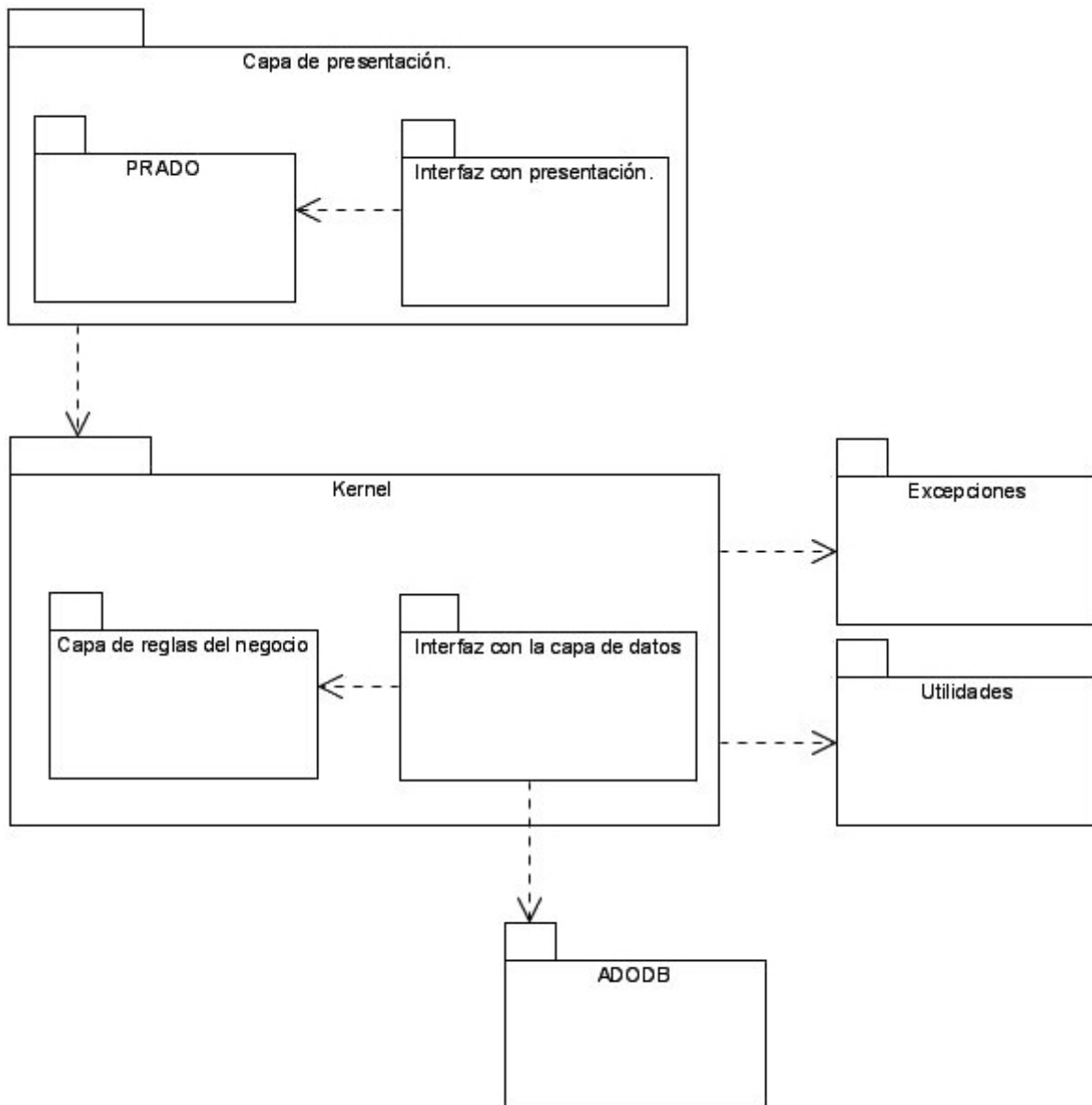


Figura 8.7. Diagrama de paquetes que refleja la arquitectura de SIMUPAT.

Una vez que hemos enunciado de forma general las funciones de cada capa, es hora de profundizar en cada una de ellas para brindar al lector no solo como fue diseñada cada capa, sino como fue implementada también.

8.4.1 Capa de presentación

La capa de presentación, como ya fue enunciado, es la responsable de conformar la interfaz gráfica de usuario, es decir, la forma en la cual los datos son recibidos y desplegados desde y hacia el usuario final. Como SIMUPAT es un sistema concebido para funcionar en Web, se hace necesario que la capa de presentación utilice el lenguaje estándar definido por la W3C para tal efecto: el HTML. Sin embargo, el HTML no brinda por si mismo todos los elementos para hacer una interfaz bien diseñada, con apariencia limpia; por lo tanto, se hace necesario combinar el HTML con otras técnicas para construir la GUI. Dichas técnicas son la utilización de hojas de estilo en cascada (CSS de *Cascade Style Sheet*) y el JavaScript, utilizado en el sistema para efectuar diferentes validaciones de formularios antes que éstos sean enviados para su procesamiento. Tanto el código HTML como el código JavaScript son escritos dinámicamente por SIMUPAT, quien utiliza su capa de Interfaz con Presentación para tal efecto, mientras que las hojas de estilo residen en un directorio determinado (véase la vista de implementación). Los eventos que son generados por el usuario, como por ejemplo hacer clic sobre una casilla de verificación, ocasionan un POST que envía información a PRADO para que éste llame el método de manejo asociado con dicho evento, tal y como se vio en la sección 8.2.

8.4.2 Capa de interfaz con presentación

Esta capa, como ya se dijo, es la encargada de procesar los eventos generados por la interacción del usuario con la GUI, y de escribir el código HTML/JavaScript necesario para la visualización de los resultados. Esta última labor, escribir la salida, es efectuada de la siguiente manera:

El marco general de la presentación, llamado plantilla, es almacenado en un archivo **.page* o **.tpl* que contiene código combinado HTML/XML, cuya sintaxis es definida por PRADO y que determina la ubicación visual de los componentes. Considere el siguiente código de ejemplo:

```

<com:TContent id="body">
<h1> Quiz </h1>
<com:TLabel Text="Dime tu nombre:" id="etiqueta_nombre" />
<com:TTextBox id="caja_texto_nombre" TextMode="SingleLine" />
<center>
<br>
<fieldset>
<legend> Pulse en el bot&ocute;n </legend>
<center>
<com:TButton id="boton_procesar_quiz" Text="Enviar datos" OnClick="procesar" />
</center>
</fieldset>
</center>
</com:TContent>

```

Figura 8.8. Ejemplo de una plantilla de página para usar con PRADO, cuyo nombre de archivo es quiz.page.

El código anterior muestra la estructura típica de una plantilla, los componentes TLabel, TTextBox, y TButton son declarados en ésta, de tal forma que ello determina como aparecerán en la GUI. Note la definición del botón, en donde se define la propiedad OnClick; como seguramente habrá adivinado, esta propiedad define el nombre del método manejador del evento OnClick del botón.

Hay que aclarar que los archivos de plantilla NO SON la salida mostrada al usuario, ya que éste ve código HTML/JavaScript puro, sin nada de definiciones de eventos o código de ningún tipo, esa es la razón por la cual la capa de presentación está completamente separada de las capas en donde los datos son procesados. La plantilla es una guía de cómo la página debe ser mostrada al usuario y de los componentes que serán usados en dicha página, el trabajo de escribir el HTML que muestre esos componentes al usuario es llevado a cabo por PRADO. La página de salida generada al usuario puede verse en la figura 8.10.

En los controles personalizados que se escriben para trabajar con prado, la estructura de la plantilla es similar, excepto que la extensión del archivo de la plantilla es *.tpl*. Estos controles personalizados pueden ser llamados dentro de las plantillas de páginas o dentro de otros controles, de modo que el código de la plantilla sea reutilizado al máximo.

Por otra parte está el código PHP que se encarga de manejar los eventos declarados en la plantilla, en nuestro ejemplo el evento OnClick del botón rotulado "boton_procesar_quiz". El código de ejemplo, en líneas generales, sería:

```

<?php
/**
 * Archivo que ilustra el esquema típico de una clase de interfaz de presentación.
 * Este archivo es creado por los autores para documentar el libro final de SIMUPAT
 * @author Autores SIMUPAT @date 13/10/2006
 * @package Ejemplo
 */

/**
 * Clase de ejemplo para mostrar la interacción entre PRADO y SIMUPAT.
 */

class quiz extends TPage
{
    public function procesar($sender)
    {
        //Recuperamos el nombre de la caja de texto
        $nombre=$this->caja_texto_nombre->text;
        //Imprimimos el nombre en la etiqueta
        $this->etiqueta_nombre->Text=$nombre;
        $this->boton_procesar_quiz->Visible=false ;
        $this->caja_texto_nombre->Enabled=false;
        //resto del código manejador...
    }
}
?>

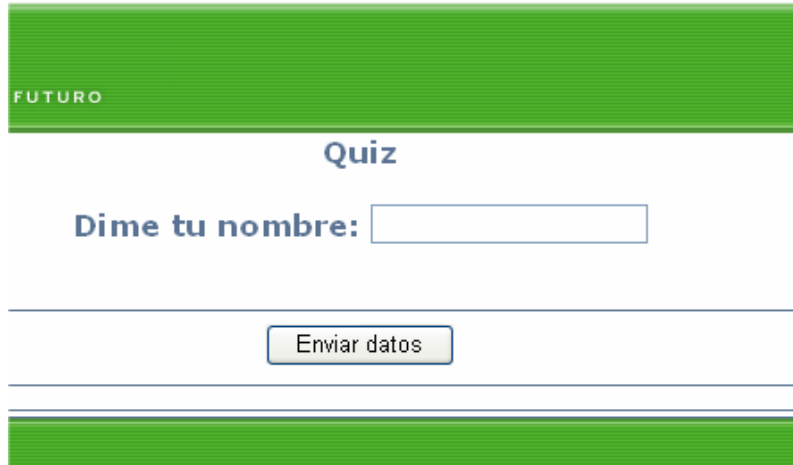
```

Figura 8.9. Ejemplo del código PHP de SIMUPAT que responde a los eventos de PRADO para la página de ejemplo *quiz.page* y cuyo nombre de archivo es *quiz.php*

Como se puede ver en el ejemplo, cada página del usuario es manejada por una clase del mismo nombre. En el ejemplo, la página cuya plantilla es *quiz.page* es manejada por la clase *quiz*, implementada en el archivo *quiz.php*. Las ventajas que tiene encapsular las páginas en objetos son muy grandes: es posible incluir páginas dentro de otras, heredar páginas, crear diferentes plantillas para una misma página, etc.

Continuando con el ejemplo, cuando el usuario digita el nombre y pulsa el botón, el evento *OnClick* es disparado por PRADO y el método *quiz::procesar(\$sender)* es invocado para que responda a dicho evento. En el código de ejemplo, el método recoge el contenido de la caja de texto, que en realidad es un objeto de tipo *TTextBox*, imprime el nombre digitado en la etiqueta cuya identificación es *etiqueta_nombre*, vuelve invisible el botón y deshabilita la caja de texto. Si el lector es familiar con la programación visual en escritorio, por ejemplo la programación en Delphi,

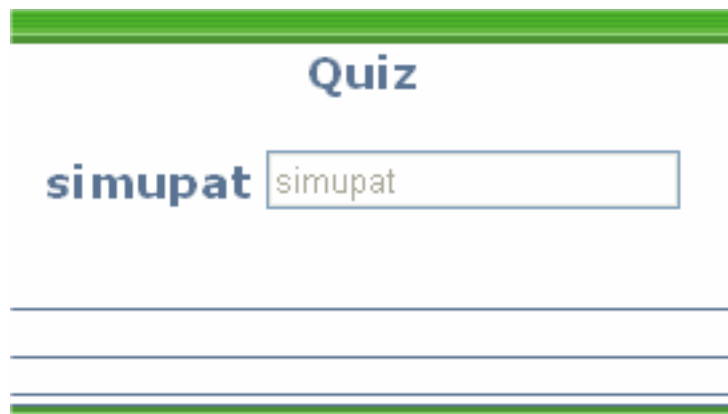
observará en seguida que la idea es la misma: las plantillas son los formularios y las clases de manejo son el código que posee lo que dichos formularios o plantillas deben hacer.



The screenshot shows a web page with a green header bar containing the word "FUTURO" in white. Below the header, the word "Quiz" is centered in a blue font. Underneath, the text "Dime tu nombre:" is followed by a text input field. Below the input field is a button labeled "Enviar datos". The page is framed by horizontal lines, with a green bar at the bottom.

Figura 8.10. Fragmento de la captura de pantalla para la página generada por el código de ejemplo quiz.php y la plantilla quiz.page.

No solamente las páginas son clases, sino que prácticamente todos los elementos que las conforman lo son: las cajas de texto, botones, etiquetas, menús, etc. Todo está encapsulado gracias a PRADO, y por lo tanto todo se puede manejar de forma dinámica, aprovechando las ventajas de la programación orientada a objetos con los elementos que forman la interfaz gráfica de usuario.



The screenshot shows the same "Quiz" page as in Figure 8.10, but now the text "simupat" is visible in the input field. The word "simupat" is also displayed in a blue font to the left of the input field. The rest of the page layout, including the "Enviar datos" button and the green header/footer bars, remains the same.

Figura 8.11. Fragmento de la página de salida una vez que el usuario ha escrito el nombre "SIMUPAT" y pulsado el botón "Enviar datos".

Ya que se ha enunciado brevemente como funciona la programación en PRADO, continuemos con la descripción de esta capa que ha sido escrita usando dicho framework. Respetando la lógica de

la arquitectura en tres capas, los componentes de esta capa únicamente preprocesan los datos capturados al usuario para que estén listos para su procesamiento en la capa de Reglas del Negocio. Dicho preprocesamiento consiste en validar los datos que han sido introducidos por el usuario, en capturar errores y hacer que los datos sean entendibles por las capas que se encargarán de su procesamiento definitivo. De forma similar, esta capa construye la presentación de los resultados provenientes de las otras capas al usuario final.

8.4.3 El kernel

Estableciendo una analogía con la arquitectura de los sistemas operativos, las capas de Reglas del Negocio e Interfaz con la capa de Datos han sido bautizadas como el “kernel” o núcleo del sistema. El kernel está escrito en PHP y es completamente independiente de PRADO, constituyendo una API completamente encapsulada en forma de una jerarquía de clases debidamente documentada en el Manual del Programador, aunque en esta sección se incluyen los diversos diagramas de clases que muestren la composición de las clases de las capas del kernel.

A continuación se estudiarán las capas pertenecientes al kernel, como ya se dijo, la de reglas del negocio e interfaz con la capa de datos, junto con dos paquetes especiales que pertenecen al kernel: el paquete de excepciones y el de utilidades.

8.4.3.1 Capa de reglas del negocio

Como está establecido en la teoría de la arquitectura de las tres capas, esta capa se encarga de representar en forma de objetos las entidades que conforman la información manejada por el sistema y las reglas que determinan las interacciones entre estas entidades, llamadas “reglas del negocio”. Cada entidad es representada por una clase, notada con la letra C (mayúscula) seguida del nombre de la entidad en minúscula sin tildes, por ejemplo, la clase Cdiagnostico representa a la entidad diagnóstico, es decir, la información que el sistema debe manejar sobre un diagnóstico y los métodos para hacerlo. Por comodidad cada clase ha sido implementada en un archivo que tiene su mismo nombre y la extensión *.inc*. El diagrama de clases UML que muestra la estructura de esta capa se muestra a continuación.

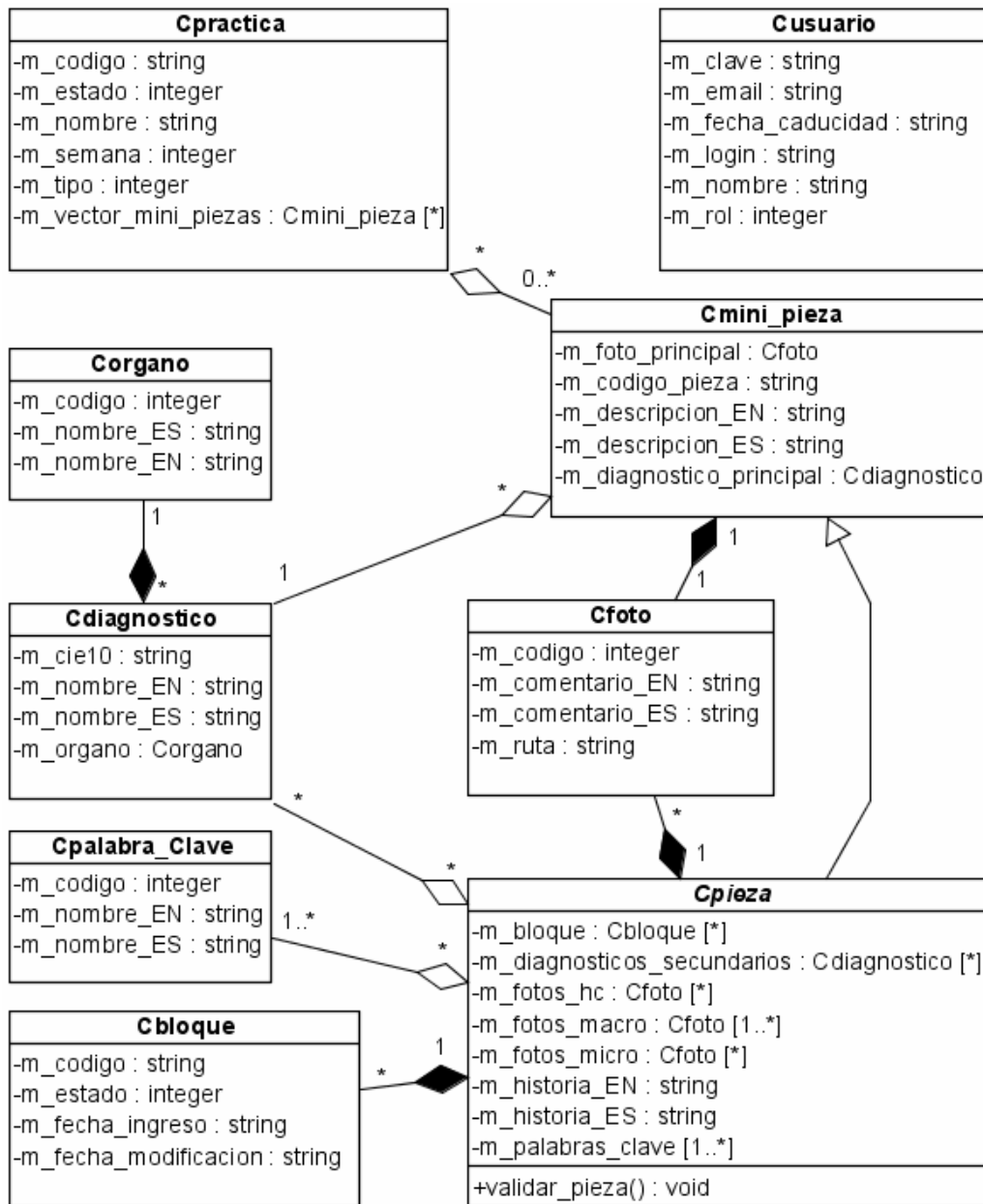


Figura 8.12. Diagrama de clases correspondiente a la Capa de Reglas del Negocio en SIMUPAT. Los métodos necesarios para establecer y obtener los atributos han sido omitidos por comodidad.

8.4.3.2 Capa de interfaz con la capa de datos

Esta capa es la encargada de construir y enviar instrucciones SQL al programa manejador de bases de datos, en la actualidad MySQL aunque es posible implementar cualquier otro DBMS como msql, Informix u Oracle. La compatibilidad con un determinado manejador de bases de datos está determinada por los siguientes puntos:

- Disponibilidad de un controlador (*driver*) que comunique el DBMS y la API de ADODB.
- El DBMS debe tener integridad referencial y control de transacciones.

Como se puede ver en las condiciones para que un programa manejador de base de datos sea compatible con SIMUPAT, las clases de la capa de interfaz con la capa de datos están escritas usando el paquete ADODB, que es una API escrita en PHP y con licencia BSD basada en el popular ADO de Microsoft. (Véase la sección 9.1.1)

Las clases de esta capa se dividen en dos grandes grupos: las colecciones, y las clases con acceso a datos. Analizaremos ambos tipos de clases a continuación:

Las colecciones son clases que consultan en la base de datos, es decir, se encargan de las labores de consulta que sean necesarias, como la búsqueda de piezas en el sistema mediante palabras clave, la búsqueda ramificada, la consulta de las prácticas disponibles, etc. Cada colección se encarga de una entidad determinada, por ejemplo, la colección de diagnósticos se encarga únicamente de las consultas sobre diagnósticos en la base de datos, la colección de órganos hace lo propio con los órganos, etc. Las excepciones a la afirmación anterior son las palabras clave, las fotografías y los bloques de resina (cuyas clases en la capa de Reglas del Negocio son *Cpalabra_clave*, *Cbloque*, y *Cfoto* respectivamente) ya que el manejo de estas entidades corresponde de forma exclusiva a la entidad *Pieza* y por lo tanto ésta es la que se encarga de las consultas que se relacionen con dichas entidades. Cada clase colección realiza las siguientes funciones:

- Valida que las consultas que son enviadas desde las otras capas sean de consulta, es decir, que únicamente tengan la cláusula *SELECT* y que opere sobre el ámbito de la colección, por ejemplo, la colección de órganos únicamente puede consultar la tabla *órganos* en la base de datos.
- Hace efectiva la consulta y devuelve los resultados en forma de vectores de objetos de la entidad respectiva, así por ejemplo, la colección de prácticas (*Colección_practicas*) devuelve un vector con objetos de tipo *Cpractica*.

- Devuelve información sobre el resultado de la consulta, entre otros, cuantos elementos han sido retornados.

Por otro lado, las clases con acceso a datos son extensiones de las respectivas clases de las Reglas del Negocio, de tal forma que dichas clases tengan funciones para interactuar con la base de datos. Cada una de estas clases hereda de la clase en la capa de reglas del negocio que trate la misma entidad, por ejemplo, la clase Cpieza_db de esta capa hereda de la clase Cpieza de la capa de reglas del negocio para de esta forma manejar las operaciones de datos sobre las diferentes tablas que guardan la información de cada pieza del museo. La notación de estas clases es: el nombre de la clase padre en la capa de las reglas del negocio concatenada con la cadena “_db” para indicar que se trata de una clase de acceso a datos. Las funciones de las clases con acceso a datos son:

- Generar código SQL que refleje las operaciones necesarias para crear, editar o eliminar la entidad que ha dado origen a la clase, por ejemplo, la clase Cusuario_db genera el código SQL necesario para crear, editar o eliminar un usuario en la base de datos.
- Ejecutar el código SQL utilizando control de transacciones, para de esta forma preservar la integridad de la base de datos en caso de error.
- Ejecutar código SQL que se encargue de construir la entidad respectiva a partir de los datos almacenados en la base de datos, por ejemplo, ejecutar SQL para que a partir de los datos contenidos en la tabla órganos se construya el objeto Corgano respectivo.

La excepción a esta regla son las entidades Foto, ya que su administración en la base de datos está a cargo de la clase Cpieza_db. Los diagramas de clases que ilustran la estructura de la capa de Interfaz con la capa de Datos son:

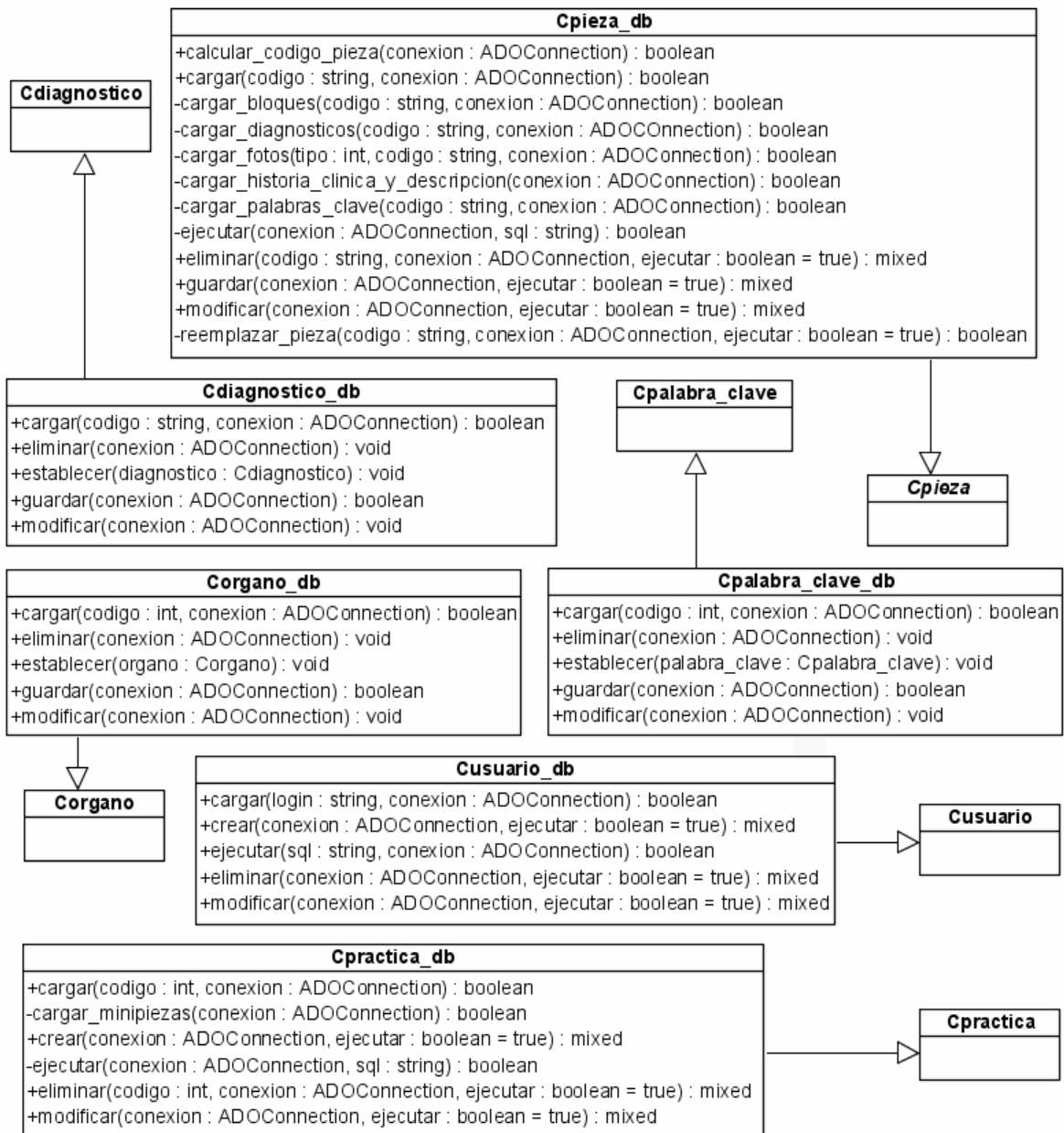


Figura 8.13. Diagrama de clases con acceso a datos para la capa de Interfaz con la capa de datos en SIMUPAT.

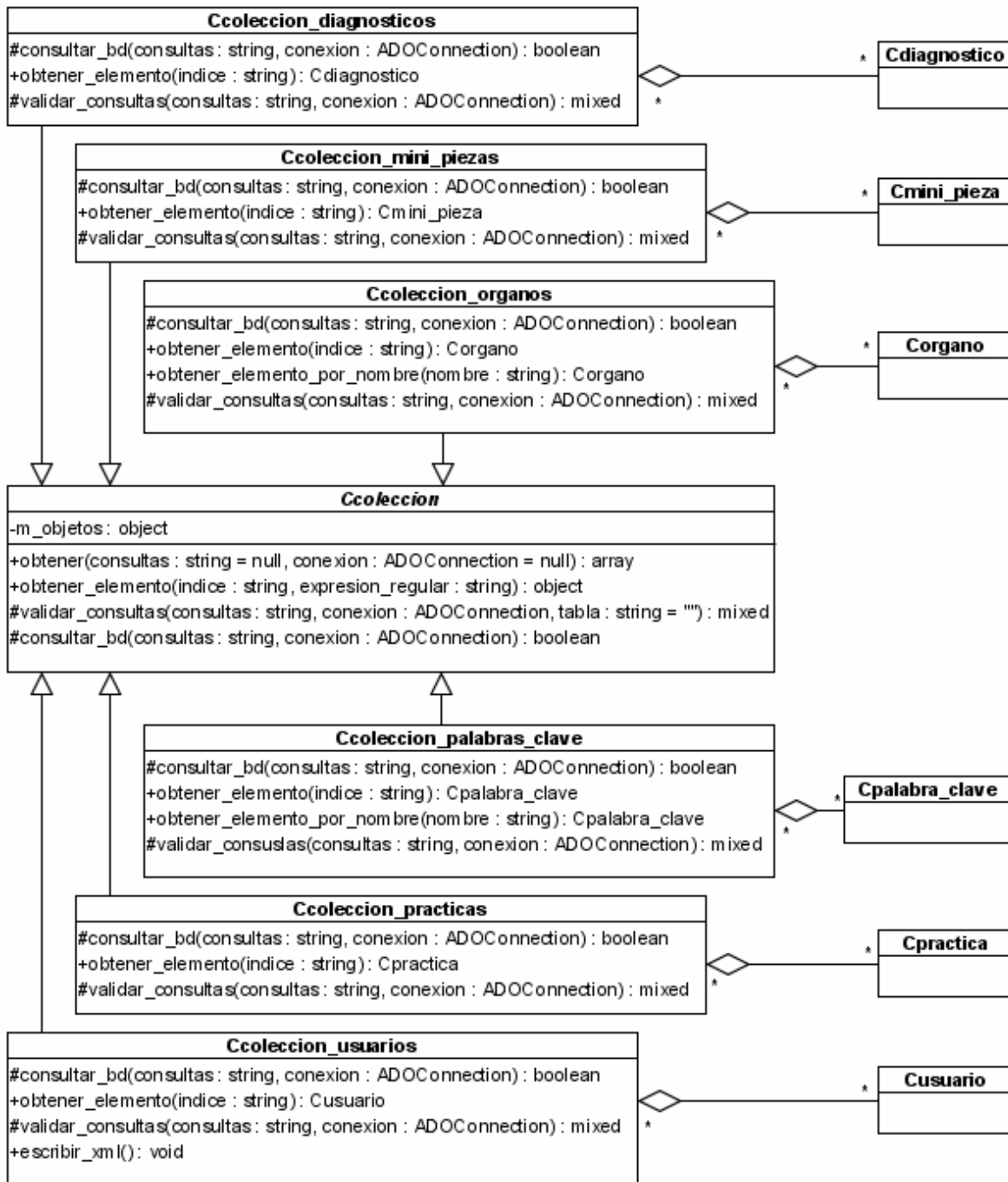


Figura 8.14. Diagrama de las clases de colección para la capa de Interfaz con la capa de datos en SIMUPAT.

8.4.3.3 Las excepciones

El manejo del error es uno de los aspectos más importantes, por no decir que es el más importante, en todo sistema que pretenda ser estable y robusto, y por ello los autores han prestado

especial cuidado a este t3pico. Por ello vale la pena discutir ampliamente sobre como SIMUPAT administra el error buscando siempre que su soluci3n sea lo m3s r3pido posible.

Comencemos por decir que hay dos grandes tipos de error: los errores atrapados en presentaci3n y los errores atrapados en el kernel (vea la secci3n 8.4.3).

Los errores atrapados en presentaci3n obedecen a un error del usuario en el manejo del sistema, como por ejemplo dejar vaci3o un campo de texto obligatorio, y no representan realmente una falla del sistema, sino una operaci3n con datos err3neos. Tales errores son capturados en las capas de presentaci3n e interfaz con presentaci3n y resultan en un mensaje de error mostrado al usuario, como en la figura 8.15. Estos errores "normales" son de baja relevancia en este documento y por lo tanto no se discutir3 m3s sobre ellos.

The screenshot shows a form titled "Par3metros del usuario de la base de datos". It contains four input fields: "Dominio (localhost si la base de datos est3 en el mismo servidor que SIMUPAT):" with the value "localhost"; "Nombre de la base de datos:" which is empty; "Nombre de usuario:" which is empty; and "Contrase1a de acceso:" which is empty. Each of the last three empty fields has a red asterisk to its right. Below the form, a red error message reads "No puede continuar porque:" followed by a bulleted list: "Se requiere que especifique una base de datos", "Se requiere que escriba un nombre usuario", and "Se requiere que escriba una contrase1a". At the bottom right of the error box is a "Continuar" button.

Figura 8.15. Fragmento de una captura de pantalla del m3dulo de instalaci3n, en donde se ha mostrado un error al usuario.

En cambio, los errores atrapados en el kernel representan una situaci3n muy diferente: el error ha traspasado los mecanismos de validaci3n normales y ha llegado hasta las capas de procesamiento de informaci3n, en donde se supone que los datos son correctos y ya han sido validados. Estos errores son producto no del usuario, quien simplemente dispara la situaci3n de error, sino de *bugs* e incongruencias en el c3digo de la aplicaci3n y representan un error en el sistema. Por lo tanto deben ser atendidos con extrema urgencia por el equipo de desarrollo o por el equipo de mantenimiento y soporte t3cnico.

Para manejar dichos errores graves los autores han desarrollado ciertas clases especiales, ubicadas bajo el paquete excepciones y cuyo diagrama de clases se muestra en la figura 8.16, que brindan una forma de estabilizar el sistema y preservar a toda costa la integridad de la base de datos. La clave para entender estas clases es el concepto que ha dado nombre a esta sección: el concepto de excepción.

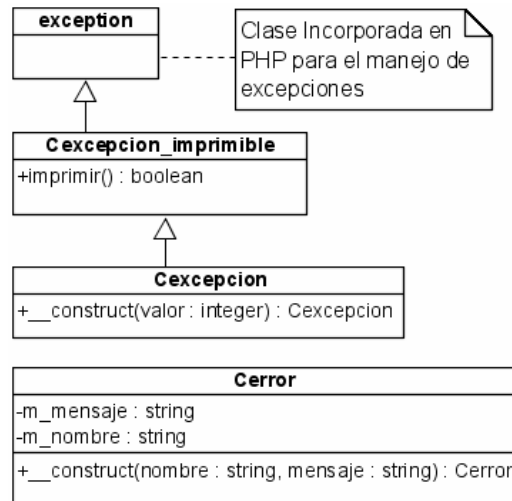


Figura 8.16. Diagrama de clases para el paquete de excepciones en SIMUPAT.

Una excepción es una instrucción especial que cuando es ejecutada detiene la ejecución del método en donde está presente e intenta ejecutar un código de salvamento. Cuando las clases del kernel detectan un error, se ejecutan las instrucciones de excepción en donde se instancia un objeto de la clase Cexcepción; dicha clase tiene las siguientes funciones:

- Buscar en la lista de errores la descripción textual del error para imprimir un mensaje en pantalla con el nombre del error y el texto descriptivo.
- Construir un trazado inverso del orden en el cual los métodos fueron ejecutados durante la situación de error, de tal forma que se observe cuales métodos entraron en ejecución y en que línea y archivo se originó la excepción.
- Para cada método del trazado inverso hacer un volcado de cada parámetro introducido en dicho método, de tal forma que se sepa que tipo de dato entró.

Cuando la excepción es disparada, el código de salvamento indica a la clase Cexcepcion que imprima en pantalla o a un archivo la información necesaria para investigar la causa del error y hace que la función o método que estaba en ejecución retorne un valor lógico de falso. Esto indica a los otros componentes del sistema que ha surgido un error y hará que ellos disparen más excepciones hasta detener de forma segura la ejecución. Con la información suministrada por la clase Cexcepcion, los desarrolladores y personal de soporte sabrán que archivo fue el responsable, y en que línea del código falló el sistema. Un ejemplo del código que dispara una excepción y una pantalla con trazado inverso se muestran en las figuras 8.17 y 8.18 respectivamente.

```

* El valor de retorno indica si la funcion tuvo exito
* Este objeto devuelve falso si la pieza no ha sido encontrada
* o si previamente ha habido un error en la base de datos
* en cuyo caso se imprimiran las excepciones
* @param ADOConnection $conexion Objeto de conexión a la base de datos
* @param integer $codigo Código del órgano
* @return boolean
* @throws ERROR_NO_ENTERO En caso que el parámetro codigo no sea un número entero
*/
public function cargar($codigo, $conexion)
{
    try
    {
        if (!is_int($codigo))
        {
            throw new Cexcepcion(ERROR_NO_ENTERO);
        }

        // Primero verificamos que el organo existe
        // Se retorna falso si no existe
        $consulta="SELECT * FROM organos WHERE id=".$codigo;
        $recordset_organos=$conexion->Execute($consulta);

        if ($recordset_organos->RecordCount()<=0)
        {
            return false;
        }

        $vector = $recordset_organos->GetRowAssoc();

        $this->establecer_codigo($codigo);
        $this->establecer_nombre($vector[NOMBRE_ES], ES);
        $this->establecer_nombre($vector[NOMBRE_EN], EN);

        return true;
    }
    catch (Cexcepcion $e)
    {
        $e->Imprimir();
        return false;
    }
}

```

Figura 8.17. Implementación del método cargar, perteneciente a la clase Corganos_db, en donde se muestra el punto en donde una excepción es disparada.

Operación anómala del método establecer_nombre (ERROR_NO_CADENA)

El argumento pasado a la función debe ser de tipo string			
#	Archivo	Línea	Función
0	C:\AppServ\www\simupat\protected\kernel\pruebas\~phpdesigner_output_tmp.php	55	Cusuario->establecer_nombre("")

Figura 8.18. Ejemplo de una pantalla con trazado inverso generada por una excepción de la clase Cusuario.

8.4.3.4 Las utilidades

Aunque el repositorio de funciones de PHP es inmenso y las clases que otros desarrolladores han hecho para extender la funcionalidad de PHP son innumerables, se hace necesario a veces escribir código para labores varias dentro del desarrollo. Un ejemplo de lo anterior es la función *codificar_html()* implementada en el archivo *cadena.inc*, la cual se encarga de realizar conversiones entre caracteres especiales (como la tilde) y el código HTML de tal forma que los navegadores las visualicen correctamente.

Los archivos que conforman el paquete de utilidades son procedurales, ya que no es necesario el encapsulamiento en forma de objetos sino simplemente la disponibilidad de dichas funciones en donde se necesiten.

8.4.4 La capa de datos

Como ya se dijo, la capa de datos en SIMUPAT está compuesta por la capa de abstracción ADODB, el programa manejador de bases de datos y la base de datos en sí. Los autores han utilizado las bases de datos InnoDB en MySQL para este desarrollo, pero es posible montar el sistema en otra base de datos relacional que tenga disponible un driver de conexión con ADODB (y por ende con PHP).

La estructura de diseño de la base de datos de SIMUPAT está ilustrada en el diagrama entidad-relación que se muestra a continuación:

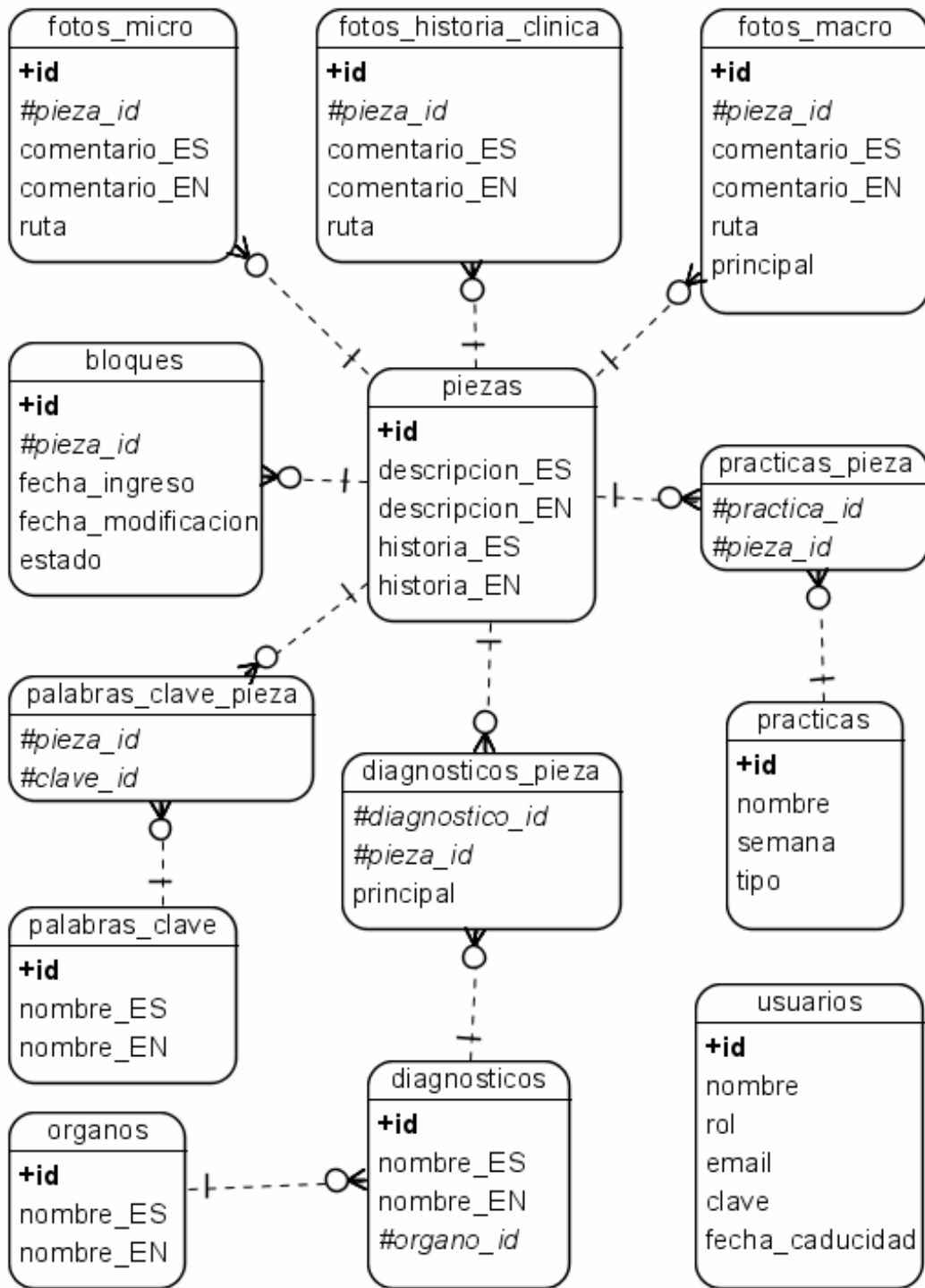


Figura 8.19. Diagrama entidad-relación para la base de datos de SIMUPAT.

En la figura 8.19 puede verse incluso la estructura de cada tabla que compone la base de datos de SIMUPAT, en donde las llaves primarias y foráneas están notadas en negrita o con el símbolo #.

También se pueden observar las relaciones entre cada una de las entidades y si alguna de estas relaciones tiene atributos, éstos también son mostrados.

En esta capa hay que resaltar el *objeto conexión*, que es un objeto cuya clase es definida por ADODB y construido por SIMUPAT en el proceso de instalación, que contiene la información necesaria para conectarse a la base de datos. Esta información es el nombre de la base de datos, el nombre de usuario y la contraseña. El objeto conexión, como ya se dijo, es construido por el módulo de instalación con base en los datos suministrados por el administrador durante dicha fase y almacenado en el archivo *conexion.inc* ubicado en la carpeta datos del paquete kernel.

Cuando la aplicación se inicia el objeto conexión es instanciado por el archivo principal de configuración *config.inc* y de este modo queda disponible a todas las clases que conforman SIMUPAT. Como el objeto conexión encapsula la conexión al manejador de bases de datos, ninguna clase debe preocuparse por realizar la conexión (basta con utilizar los métodos del objeto conexión) y por tanto ninguna clase es dependiente del tipo de la base de datos, del nombre de la base de datos o de la contraseña, siendo estos parámetros absolutamente configurables.

Configuración de la base de datos

SIMUPAT necesita un usuario y una base de datos para funcionar. Actualmente SIMUPAT opera con bases de datos MySQL tenga INNODB. Para añadir funcionalidad con otras bases de datos vea la documentación del programador.

NOTA: El usuario especificado debe tener los siguientes permisos

- CREATE TABLE
- SELECT
- INSERT
- DELETE
- UPDATE

Parámetros del usuario de la base de datos

Dominio (localhost si la base de datos está en el mismo servidor que SIMUPAT):

Nombre de la base de datos:

Nombre de usuario:

Contraseña de acceso:

Figura 8.20. Fragmento de la captura de pantalla de la sección “Configuración de la base de datos” parte del proceso de instalación de SIMUPAT.

8.4.5 La capa de configuración

Aunque SIMUPAT ha sido diseñado y construido siguiendo los requerimientos del Departamento de Patología de la UIS, y dichos requerimientos estén consignados en la Especificación de Requerimientos de Software (ERS) incluida en este libro, en el futuro ciertos “detalles” de los requerimientos pueden cambiar. Es posible que, por ejemplo, en el futuro se exija que una pieza tenga más de siete palabras clave, en la ERS actual se dice que tres, y el sistema debe poder responder a esta nueva situación sin afectar su código interno. Por ello es indispensable que todos los parámetros que pueden cambiar en el futuro puedan cambiarse sin mayor impacto para el sistema.

SIMUPAT ha sido concebido para que sea lo más parametrizable posible y que pueda configurarse fácilmente, por ello se ha construido la capa de configuración, para que el comportamiento de SIMUPAT sea lo más amoldable posible para acoplarse a los detalles de la lógica del negocio. La labor de mantener las configuraciones es del administrador y como medida de seguridad no se ha implementado ninguna interfaz gráfica que cambie los archivos de configuración sino que esto debe hacerse a mano.

Los archivos de configuración se dividen en tres partes:

- Los archivos de autorización.
- Los archivos de definición de constantes y parámetros.
- Los archivos de configuración de la capa de Interfaz con Presentación.
- Los archivos de autorización serán explicados en la sección 8.4.6.1 y no se hablará de ellos en esta sección. Los archivos de definición de constantes y parámetros son los que guardan la definición de los parámetros que utiliza el sistema y son los siguientes: el archivo *constantes.inc* y *config.inc* ubicados dentro de la carpeta kernel (vea la sección 8.3).
- Un punto a resaltar en esta sección es el *modo de funcionamiento de SIMUPAT* que es un parámetro que indica a SIMUPAT si el sistema está en mantenimiento o si está en producción. Si el sistema está en modo “debug” o prueba, las pantallas de excepciones son enviadas a la

salida en pantalla del cliente o de lo contrario son llevadas a un archivo de registro de errores en el servidor. Esta decisión es importante ya que la información de trazado inverso da una gran cantidad información sobre la estructura del sistema y podría ser aprovechada por atacantes informáticos

```
<?php
/**
 * Archivo de configuración.
 * @copyright Copyright (c) 2006, Universidad Industrial de Santander
 * @author Equipo de desarrollo SIMUPAT <simupat@walla.com>
 * @package Configuración
 */

/**
 * Directorio de instalación de SIMUPAT
 */
define("SIMUPAT_DIR",dirname(__FILE__));

//////////
// CONSTANTES //
//////////
/**
 * Incluye el (@link constantes.inc archivo de constantes).
 */
require_once(SIMUPAT_DIR."/constantes.inc");

//////////
// PARAMETROS //
//////////
/**
 * Tamaño máximo para una fotografía en bytes
 */
define("MAXIMO_TAMAÑO_FOTO", 500000);

/**
 * Número máximo de caracteres permitidos para el nombre de usuario
 */
define("MAX_CARACTERES_LOGIN", 15);

/**
 * Número mínimo de caracteres permitidos para el nombre de usuario
 */
define("MIN_CARACTERES_LOGIN", 4);
```

Figura 8.21. Fragmento del archivo de configuración config.inc en donde pueden verse algunas de las definiciones de parámetros de SIMUPAT.

8.4.6 La seguridad

La seguridad es, sin duda alguna, uno de los factores decisivos a la hora de construir un sistema de información. Aunque no es el propósito de esta sección estudiar con detalle los diferentes aspectos que deben ser considerados para considerar un sistema como “seguro”, vale la pena unificar conceptos antes de profundizar sobre el esquema de seguridad adoptado por SIMUPAT.

En primer lugar no existe un sistema completamente seguro; ya que cada medida de seguridad resta, de una forma u otra, funcionalidad al sistema que se está protegiendo. Si un sistema de información, por ejemplo, se aísla de Internet para evitar ataques externos a la red de la organización en donde está implantado, se pierden las enormes ventajas que tendrían los empleados como, por ejemplo, hacer sus labores cuando están de viaje. Podríamos decir que el punto máximo de la seguridad es tener al sistema apagado, aunque es obvio que un sistema apagado es inútil y no del todo seguro ya que un desastre natural, por ejemplo, podría dañar el hardware del que está compuesto.

¿Entonces, que es la seguridad informática? Los autores se atreven, sin ser una autoridad en la materia, a dar la siguiente definición: La seguridad informática es un conjunto de políticas de uso y administración encaminadas a cumplir los siguientes objetivos:

- Mantener la integridad de los datos: Es decir, que la información administrada por el sistema permanezca íntegra, es decir, coherente según la lógica de la organización en donde dicho sistema ha sido implantado. Esto implica asegurar que los datos no se perderán y que no se corromperán por una u otra causa.
- Mantener la privacidad de los datos: Es decir, asegurar que la información administrada por el sistema sea vista y/o alterada únicamente por las personas autorizadas para ello por las directivas de la organización.
- Contribuir a la auditoria de la organización: Es decir, contribuir a que el personal de la organización haga buen uso de la información administrada por el sistema y de esta forma prevenir sabotajes o negligencia.

Los anteriores objetivos son un modo muy general de ver la seguridad informática, pero los autores esperan que sea suficiente para el propósito de esta sección. Para cumplir los objetivos anteriormente expuestos existen las siguientes políticas de seguridad; por razones obvias, nos limitaremos a considerar las más comúnmente adoptadas.

- Poseer copias de seguridad de los datos para poder recuperar el sistema de un desastre que haya ocasionado la pérdida o la corrupción de la información.
- Implantar restricciones a la información que los usuarios del sistema manejan, de acuerdo a su rol dentro de la organización. (Por ejemplo, que un administrador pueda cambiar ciertos datos, mientras un usuario no administrados pueda consultarlos únicamente)
- Cifrar la información de tal forma que sólo los usuarios con derecho a la misma puedan decodificarla y por lo tanto consultarla.
- Mantener un registro de cada operación efectuada por el usuario dentro del sistema, de tal forma que en caso de un sabotaje se pueda utilizar dicho registro en la investigación.
- Educar a los usuarios y crear una cultura dentro de la organización que contribuya a la implantación y la ejecución de las políticas de seguridad diseñadas por la administración.

Las políticas anteriores suponen que la identidad de cada usuario es verificada por el sistema, lo cual conlleva al tradicional esquema de nombre de usuario y contraseña. Dicho esquema de seguridad está constituido por dos etapas fundamentales:

- Autenticación: Significa que el sistema verifique la identidad del usuario, es decir, que el usuario sea quien dice que es.
- Autorización: Significa que el usuario, una vez autenticado, tenga acceso únicamente a las funciones e información del sistema para las cuales esté autorizado por la administración.

8.4.6.1 El proceso de autenticación y autorización en SIMUPAT

Una vez que se ha discutido sobre algunos aspectos generales de la seguridad informática, es hora de abordar lo que los autores han implementado en SIMUPAT. Antes de comenzar, los autores reconocen que las políticas implementadas en SIMUPAT están lejos de ser suficientes (de hecho **ninguna** política de seguridad es suficiente) y que aún queda mucho trabajo por hacer en ese aspecto.

SIMUPAT ha adoptado el tradicional esquema de nombre de usuario y contraseña para efectuar la autenticación de los clientes que se conectan a él. Por defecto un cliente que no pasa por el proceso de autenticación se le llama *invitado* (en la configuración de seguridad se nota por un carácter especial: '?') y sólo posee labores de consulta en el repositorio fotográfico y de información del sistema. La información sobre los usuarios se almacena en dos lugares principales: la base de datos y los archivos de configuración.

En la base de datos se almacena la información del usuario para dar soporte a los casos de uso relacionados con la creación, edición y eliminación de usuarios. Sin embargo en el proceso de autenticación, la base de datos **no** es consultada. En su lugar, los archivos de autenticación son leídos para tal efecto. El siguiente diagrama de actividades muestra el proceso de autenticación en el sistema:

- El usuario intenta acceder a una función que requiere autenticación o entra al programa de autenticación.
- El sistema verifica si el usuario está autenticado (labor realizada mediante el envío y validación de *cookies*).
- Si la *cookie* almacenada en el computador del usuario es válida y no ha expirado, el sistema recupera el rol del usuario; de lo contrario considera que el usuario es un invitado. Para efectos de esta explicación, supondremos que el usuario no ha sido autenticado.
- El sistema redirige al usuario a la página de autenticación, en donde se le pide que de su nombre de usuario y su contraseña.
- El usuario digita y envía su información de autenticación.
- El sistema lee el archivo de autenticación (escrito en XML) y busca allí si existe un usuario cuyo nombre de usuario exista. En caso de no existir, el sistema envía un error y el proceso falla.

- El sistema mira en el archivo de autenticación la contraseña correcta y determina si el usuario que intenta autenticarse ha suministrado la contraseña acertada. En caso de fallo, el sistema envía un error y el proceso falla.
- Una vez que el usuario ha sido autenticado, el sistema envía una *cookie* al cliente e inicia una sesión en el sistema con la información sobre el rol del usuario.
- El sistema verifica si el rol del usuario está autorizado (verificando el archivo de autorización) para acceder a la función que se intenta ejecutar.
- Si el rol no está autorizado, el sistema envía al usuario al programa de autenticación, de lo contrario el programa protegido es ejecutado.

El lector podrá notar que el proceso tradicional de autenticación y el empleado por SIMUPAT es bastante similar, y tiene razón, aunque se ha modificado para corregir ciertas vulnerabilidades. Pero antes de hablar de dichas correcciones, es esencial que el usuario conozca la estructura de un archivo de autenticación y su relación con los registros de la base de datos.

Un archivo de autenticación es un documento XML cuya definición ha sido proveída por PRADO y que tiene una estructura como la mostrada a continuación:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <authorization>
    <deny users="?" roles="TECNICO,RESIDENTE,ADMINISTRADOR_MUSEO"/>
    <allow roles="ADMINISTRADOR_SISTEMA" />
  </authorization>
</configuration>
```

Figura 8.22. Estructura del archivo de autorización para las páginas del módulo de administración de piezas en SIMUPAT.

En contraste, en la base de datos no hay almacenada ninguna regla de autorización sino sólo información de los usuarios. Si el lector consulta el diagrama entidad-relación de SIMUPAT (figura 8.19) observará que hay un campo llamado contraseña en la tabla usuarios. En efecto en dicho campo ha sido almacenada la contraseña del usuario. Como de seguro el lector adivina, esta forma de autenticación es muy peligrosa si no se toman las debidas precauciones; posteriormente

analizaremos las debilidades de esta forma de hacer las cosas. Digamos en este momento que el contenido de la base de datos siempre está sincronizado con el contenido de los archivos de autenticación (los de autorización son inaccesibles para cualquier programa del sistema y deben ser modificados a mano) de tal forma que cualquier cambio en el contenido de la base de datos, como la adición de un nuevo usuario o el cambio de una contraseña, se vea reflejado instantáneamente en el comportamiento del sistema. Las clases del kernel son las encargadas de este trabajo y corresponde al lector que desee profundizar en este punto que consulte el Manual del Programador.

Ahora que se tiene una visión general del proceso de autenticación, es hora que los autores muestren sus ventajas y desventajas. Las ventajas de este esquema son:

- Si se desea definir un nuevo rol en el sistema o cambiar las políticas de autorización, sólo hay que efectuar cambios en el archivo de autorización y no en el código fuente del sistema. (Esto no es del todo cierto, aunque los cambios en el código del sistema son realmente mínimos)
- Si se desea añadir más módulos al sistema, las reglas de autorización se aplican automáticamente a los nuevos módulos sin necesidad de escribir código fuente para ello, basta con añadir archivos de autorización a los nuevos módulos.
- La autenticación de usuarios y la autorización de roles están en archivos diferentes.

Las desventajas son:

- Si un intruso logra leer los archivos de autenticación o el contenido de la base de datos (lo cual de por sí ya es una grave intromisión) tendrá acceso a las contraseñas de los usuarios.
- La base de datos y el archivo de autenticación deben estar siempre sincronizados, si se pierde dicha sincronización, o si el archivo de autenticación se pierde, el sistema funcionará mal.

Considerando lo expuesto y muchos otros aspectos, los autores han desarrollado soluciones a las desventajas y vulnerabilidades del esquema utilizado. En primer lugar la contraseña del usuario **nunca** es realmente almacenada en el sistema, lo que se almacena realmente es su resumen

hash. Un resumen hash es una representación de la información que cumple con las siguientes propiedades:

- No importa la longitud del dato representado (su longitud en bits), la representación hash siempre tiene un tamaño constante.
- Existe un único resumen hash para un dato determinado, lo cual garantiza en nuestro caso, que nunca dos contraseñas darán origen al mismo resumen.
- El resumen hash es irreversible, es decir, teniendo el resumen es prácticamente imposible obtener el dato original.

Por lo tanto es el resumen hash almacenado en los archivos de autenticación el que es comparado contra el resumen hash de la entrada del usuario que se va a autenticar, por lo que la contraseña nunca es almacenada en el sistema. Además, si un atacante pudiera acceder a la base de datos y leer la tabla que almacena la información de los usuarios, no podría obtener sus contraseñas ni deducir su longitud, ya que el resumen hash es irreversible y tiene tamaño constante. La figura 8.21 muestra como es un archivo de autenticación en SIMUPAT con la clave enmascarada por un algoritmo HASH:

```
<?xml version="1.0" encoding="utf-8"?>
<module id="users">
  <user name="usuario1" password="b2c02e55772de785844e79033bf4f885" />
  <role name="RESIDENTE" users="usuario1" />
</module>
```

Figura 8.23. Estructura de ejemplo para el archivo de autenticación de SIMUPAT. No se puede ver la clave del usuario, sólo su resumen HASH.

El segundo inconveniente, la sincronización entre el archivo de autenticación y el contenido de la base de datos, es solucionado brindando mecanismos que permitan escribir de nuevo el archivo de autenticación con base en el contenido de la base de datos. Tales mecanismos son ejecutados cada vez en el proceso de instalación cuando se recupera de una copia de seguridad.

8.4.6.2 Las copias de seguridad

La elaboración de copias de seguridad es una de las labores más importantes en los procedimientos que un administrador lleva a cabo y facilitar esta labor es función fundamental de

un buen sistema de información. Por ello SIMUPAT posee un mecanismo para la creación de copias de seguridad: los archivos de volcado (llamados *dump*).

Un archivo de volcado no es más que un archivo que da instrucciones al DBMS para que vuelva e introduzca los datos de modo que el resultado sea una copia del contenido de la base de datos usada para construir dicho archivo. Lo anterior se aclara con el siguiente ejemplo, suponga que tiene una tabla que contiene nombres y teléfonos y que hace un volcado; el resultado sería algo como lo que sigue, asumiendo que la tabla ya haya sido creada en la base de datos:

```
INSERT INTO agenda VALUES ('Fulano de Tal', '5555555');
INSERT INTO agenda VALUES ('Sutano', '1111111');
-- - Resto de las instrucciones.
```

Figura 8.24. Ejemplo de volcado de instrucciones SQL en un archivo.

La principal ventaja de hacerlo de este modo es que un archivo de volcado está escrito en lenguaje SQL que es estándar y por lo tanto no debería presentar incompatibilidades entre los diferentes manejadores de base de datos. Aunque en la práctica hay ciertas incompatibilidades, éstas son menores y relativamente fáciles de corregir. El hecho que el volcado sea estándar es muy importante, como lo muestra la siguiente situación hipotética:

Suponga que la organización tiene SIMUPAT 1.0 actualmente, pero que antes tenía ANTIGUOPAT¹, un sistema viejo que fue reemplazado por SIMUPAT y que además existe COMERPAT² un sistema desarrollado por una compañía que efectúa las mismas funciones que SIMUPAT. Ahora, un desastre mayor ha hecho que la instalación de SIMUPAT haya quedado inutilizada y que no hay copia de dicho sistema. Un archivo de volcado permitiría que la organización usase como sistema de emergencia, mientras se consigue otra copia de SIMUPAT, a ANTIGUOPAT sin mayor demora en el llenado de su base de datos o que comprara, como último recurso a COMERPAT y usar el archivo de volcado para llenarlo con datos.

¹ Este es un nombre ficticio, cualquier parecido con la realidad es coincidencia.

² Este es un nombre ficticio, cualquier parecido con la realidad es coincidencia.

SIMUPAT provee de un caso de uso para crear un archivo de volcado, con lo cual el administrador del sistema puede crear uno en cualquier momento, descargarlo a través de Internet, y guardarlo en un lugar seguro junto con el material fotográfico.

Pero no basta con proveer un medio de generar automáticamente un archivo de volcado, es indispensable que el sistema provea un programa automático para recuperar sus datos desde un archivo de volcado.

La razón de lo expuesto en el párrafo anterior es que no siempre es posible que el administrador del sistema sea el administrador del servidor de datos y por lo tanto que no tenga acceso a la consola del DMBS. Incluso es posible que el administrador no sepa de bases de datos y que no tenga el conocimiento para manejar un DBMS, mientras que usar el sistema es relativamente sencillo. SIMUPAT cuenta con un módulo de instalación que permite la recuperación de su base de datos usando un archivo de volcado. Finalmente mostramos un fragmento de un archivo de volcado generado por SIMUPAT.

```
INSERT INTO organos VALUES ('1','gl&acute;ndula suprarrenal','adrenal');
INSERT INTO organos VALUES ('2','arteria','artery');
INSERT INTO organos VALUES ('3','vena','vein');
INSERT INTO organos VALUES ('4','vaso linf&acute;tico','lymphatic vessel');
INSERT INTO organos VALUES ('5','apendice','appendix');
INSERT INTO organos VALUES ('6','vejiga','bladder');
INSERT INTO organos VALUES ('7','hueso','bone');
INSERT INTO organos VALUES ('8','m&acute;dula &oacute;sea','bone Marrow');
INSERT INTO organos VALUES ('9','cerebro','brain');
INSERT INTO organos VALUES ('10','gl&acute;ndula mamaria','breast');
INSERT INTO organos VALUES ('11','cart&iacute;lago','cartilage');
INSERT INTO organos VALUES ('12','colon','colon');
INSERT INTO organos VALUES ('13','oido','Ear');
INSERT INTO organos VALUES ('14','es&oacute;fago','esophagus');
INSERT INTO organos VALUES ('15','ojo','eye');
INSERT INTO organos VALUES ('16','trompa de Falopio','fallopian Tube');
```

Figura 8.25. Fragmento del archivo de volcado generado por SIMUPAT del contenido de la tabla **organos**.

8.4.6.3 SIMUPAT y la protección contra la inyección del código malicioso y las técnicas de “ingeniería social”

Para nadie es un secreto que un sistema en Internet está sujeto a la acción ilícita de personas cuyo objetivo es sabotear los sistemas de la red y que deben tomarse medidas contra ello. Aunque una protección total es prácticamente imposible, siempre es mejor algo que nada y los autores han implementado en SIMUPAT estrategias para protegerse, estrategias que discutiremos en esta sección.

Para comenzar la inyección de código malicioso, junto con la llamada “Ingeniería Social”, son las formas más comunes de ataque que se han efectuado (existen muchas otras formas de ataque, como el “Man in the middle” pero que no se consideran aquí) y que son relativamente fáciles de hacer si el blanco, el sistema que va a ser atacado, está mal administrado o no posee defensas contra ello.

Comencemos por analizar la “ingeniería social”, técnica que ha demostrado ser muy efectiva aún contra sistemas con políticas de seguridad estrictas y con una buena administración. Esta técnica consiste en averiguar las contraseñas de los usuarios del sistema objetivo mediante el análisis del ambiente social en el cual dichas personas se desenvuelven, sus gustos, sus aficiones, sus datos personales y demás información que pueda estar relacionada con las contraseñas que estas personas usan para acceder al sistema. El atacante tiene como objetivo recopilar la mayor información posible sobre el usuario, al cual llamaremos “víctima”, para lo cual hurgan entre su basura, espían sus conversaciones telefónicas, vulneran su correo electrónico o sus dispositivos móviles, hablan con sus vecinos o hasta se vuelven sus amigos. Con la información recopilada el atacante prueba construir contraseñas basadas en dicha información, como fechas de cumpleaños, nombres de parientes queridos, etc.

La efectividad de la ingeniería social se ve determinada, entre otras, por las siguientes causas:

- Contraseñas fáciles de deducir: Los usuarios escogen, por razones nemotécnicas, palabras o números que le son familiares como fechas de cumpleaños, nombres de artistas favoritos, etc. Esto hace que dichas contraseñas sean muy vulnerables a la “ingeniería social”.
- Contraseñas compartidas: Es muy común que los miembros de la organización revelen su contraseña a terceros (por lo general compañeros de trabajo) con lo cual se aumentan las posibilidades de que un atacante sepa la contraseña.
- Contraseñas que no se cambian regularmente: Es común que el usuario mantenga la misma contraseña durante meses y años, aumentando el tiempo que tiene disponible el atacante para averiguarla.
- Contraseñas tontas: Se presenta comúnmente que los usuarios, sea por pereza o por no usar otras contraseñas más “importantes” como la del correo electrónico o la del banco por miedo a que haya una “puerta trasera” que facilite al administrador que se entere de la contraseña, utilizan contraseñas tontas que son fáciles de romper, por ejemplo: “1234” o “sinclave”, “password”, etc.

SIMUPAT reduce el riesgo de ataques de ingeniería social trabajando sobre dos últimos puntos de la lista anterior: las contraseñas “eternas” y las contraseñas “tontas”.

La solución implementada por los autores en SIMUPAT para obligar al usuario a cambiar su contraseña regularmente es caducar dicha contraseña, luego de cierto tiempo determinado por la administración, de modo que se vea obligado a cambiarla. En detalle, cuando el programa de autenticación detecta que la contraseña es correcta, pero que ha caducado, envía un correo al usuario informando que la contraseña ha caducado. Dicho correo tiene un enlace que lleva al usuario a una página en donde puede cambiar sus datos; dicho enlace tiene la información de autenticación (en forma de resumen hash) para que el usuario pueda entrar a cambiar la clave. Una vez que la clave ha sido cambiada, se calcula una nueva fecha de caducidad. De la misma forma si una contraseña es extraviada por parte de un usuario, cuando el administrador recibe la notificación, la contraseña caduca de inmediato, evitando cualquier acceso hasta que sea cambiada por una nueva contraseña de acceso.

En cuanto a las contraseñas tontas, SIMUPAT no permite que una contraseña tenga menos de ocho caracteres, estos parámetros son configurables. Además, para asegurarse siempre que la dirección de correo electrónico de un nuevo usuario del sistema es correcta, SIMUPAT caduca la clave inicial del usuario (generada de forma automática) y envía un correo al nuevo usuario para que finalice el proceso de registro con SIMUPAT.

Pese a todo, es claro que si los usuarios del sistema no toman conciencia de la importancia de la seguridad del mismo, cualquier medida adoptada fracasará. Es labor de las directivas de la organización educar a los usuarios.

La segunda forma de ataque, la inyección de código, es un ataque que requiere conocimientos sobre el sistema que se va a vulnerar, conocimiento de aspectos técnicos como el sistema operativo del servidor, el lenguaje de desarrollo, el motor de base de datos, etc. Una vez que el atacante sabe dicha información, intenta inyectar código malicioso en los diferentes campos de texto de la aplicación, como la interfaz de consulta, el programa de autenticación, y cualquier otro componente de la aplicación que reciba entrada del usuario. La idea es que cuando el sistema ejecute código en su operación normal se ejecute también el código malicioso inyectado y de este forma el sistema vez comprometida su integridad.

Un caso particular de la inyección de código es la inyección de código SQL, llamada en el ámbito de la seguridad informática SQL *injection*. Esta consiste en inyectar código SQL que revele información confidencial, que dañe el contenido de la base de datos, o que active una vulnerabilidad o una puerta trasera (como la clonación de un usuario administrador) que se aprovechada para un ataque posterior.

En este punto el sistema de autenticación de SIMUPAT entra al rescate: si un atacante crea un clon de un usuario privilegiado con el objetivo de crear una puerta trasera, el sistema no le dejará entrar a los datos privilegiados ya que dicho usuario clonado **no** está en el archivo de autenticación.

Para prevenir este tipo de ataque, SIMUPAT filtra el contenido de todas las cajas de texto que componen la interfaz gráfica del sistema, eliminando cualquier resto de código HTML, PHP y SQL antes su procesamiento, previniendo la inyección de código.

8.5 CONSIDERACIONES FINALES

- Los autores esperan que con la lectura de este capítulo, el usuario tenga una idea bastante completa de cómo fue diseñado SIMUPAT y de sus características principales de forma que pueda leer con más propiedad el Manual del Programador, en donde hay una descripción detallada de cada una de las clases y métodos que conforman la API de SIMUPAT. El carácter modular de esta arquitectura, orientada a objetos, viene de una concienzuda fase de análisis y diseño con las herramientas que el Lenguaje Unificado de Modulado brinda a los desarrolladores de software; una referencia completa de los casos de uso y requerimientos en los cuales se basaron los autores se encuentra en la Especificación de Requerimientos de Software.
- Finalmente, ha sido mucha la experiencia que sobre diversos aspectos del desarrollo de software los autores han adquirido durante la realización de este proyecto; por lo tanto, para quien esté pensando en desarrollar un sistema similar a SIMUPAT, vale la pena que lea la última sección de este informe de proyecto (sección 11): las recomendaciones.

9 DOCUMENTACIÓN Y MANUALES

Aunque este libro, es en sí mismo, parte de la documentación de este trabajo de grado, los autores incluyen este capítulo para que el usuario de SIMUPAT o quien desee profundizar en el uso y construcción del sistema sepa en donde se encuentran los archivos de ayuda y documentación construidos durante la realización de este proyecto.

Con excepción de la guía de instalación, los autores no han incluido una versión impresa de estos manuales y documentación por razones que se explicarán en cada sección de este capítulo. Sin embargo todo este material en formato electrónico está disponible en el CD que contiene la distribución de SIMUPAT.

9.1 EL PROCESO DE INSTALACIÓN DE SIMUPAT

Esta sección describe como instalar SIMUPAT, válida siempre y cuando se cumplan los requerimientos de hardware y software enunciados en la sección 5.5 y se disponga de una copia legal de este sistema.

El proceso de instalación se puede realizar de dos formas: de forma completamente manual y usando el programa de instalación. Las dos formas serán explicadas a continuación:

9.1.1 Proceso de instalación de forma manual

El proceso de instalación de forma manual comienza con la copia de la distribución SIMUPAT en la carpeta de documentos HTML de su servidor Web. Al mismo nivel de directorio deben copiarse todos los archivos y carpetas de PRADO y de ADODB. Una carpeta especial, llamada "fotos_simupat" que debe tener la misma estructura de la figura 9.1 debe copiarse también junto a la carpeta del sistema. Una carpeta "fotos_simupat" vacía se provee con la distribución de SIMUPAT.

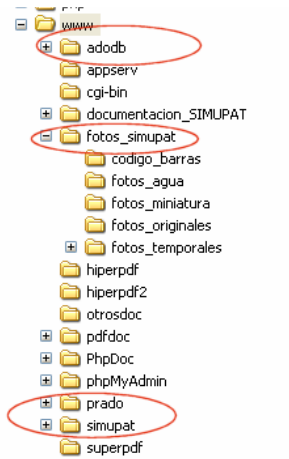


Figura 9.1. Captura de pantalla que muestra la organización de carpetas que tiene una copia instalada de SIMUPAT y que ilustra como copiar SIMUPAT junto con los paquetes ADODB y PRADO.

Una vez se tenga la estructura de carpetas de la figura 9.1, asegúrese que el PHP tenga los paquetes y extensiones instaladas. Para hacerlo revise el archivo de configuración de PHP, llamado php.ini (en Windows) y php.ini (a veces php.d) en Linux. Cuando termine escriba lo siguiente en el explorador: http://localhost/camino_a_simupat/simupat?page=instalacion.instalacion “Camino a simupat” debe reemplazarse por la URL que ubica el directorio SIMUPAT dentro del servidor Web. Si observa una pantalla como la que sigue es porque hasta el momento todo ha marchado correctamente:

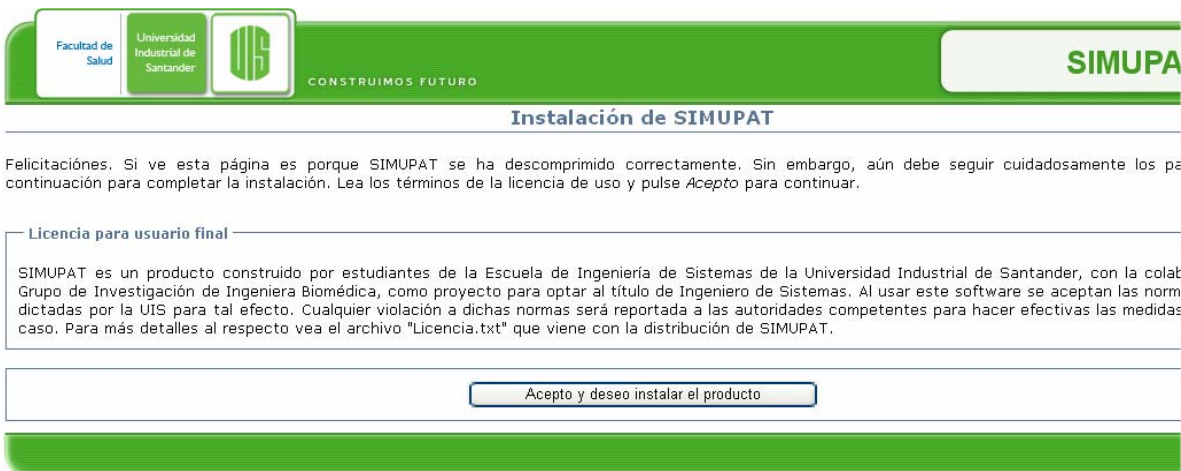


Figura 9.2. Figura que muestra la página de instalación de SIMUPAT.

Si observa una pantalla en blanco, es porque alguna de las extensiones requeridas en PHP no ha sido instalada o PRADO no se ha ejecutado correctamente. Hasta no solventar este problema, no podrá continuar.

Sin embargo si aparece el mensaje “**directory XX must be writable within the Web Server**” o algo similar, asegúrese que el servidor Web tenga permisos para escribir sobre todas las carpetas de SIMUPAT, PRADO, ADODB y “fotos_simupat”.

En caso que aparezca una pantalla como la mostrada en la figura 9.3, con el error “**Page not found**”, es porque su distribución de SIMUPAT carece del programa de instalación. Pruebe con el link: *http://localhost/camino_a_simupat/*, si la interfaz gráfica de usuario aparece (aunque no es funcional) puede continuar, de lo contrario deberá adquirir otra copia de SIMUPAT o arreglar los problemas de requerimientos de software.

Page 'instalacion.der' Not Found

Error 404

La URL requerida no ha sido encontrada en el servidor. Compruebe si la URL está bien escrita e intente nuevamente.

Si cree que este es un error en el sistema contacte al: [administrador](#).

Figura 9.3. Ejemplo de un error “Página no encontrada”.

Posteriormente, entre a la consola de su manejador de bases de datos y cree una base de datos y un usuario que tenga permisos CREATE TABLE, INSERT, UPDATE, DELETE, SELECT. Abra el archivo que contiene el objeto conexión, llamado *conexion.inc* y que está ubicado en la carpeta */protected/kernel/datos/* de la distribución SIMUPAT.

En dicho archivo ubique la línea que dice “`$_SESSION['conexion'] = ADONewConnection('mysqlt')`” y reemplace “mysqlt” por el driver para ADODB que corresponda a su manejador de base de datos.

Si su manejador de base de datos es MySQL, deje esta línea intacta. Una lista muy breve de los controladores más comunes se incluye a continuación, en la tabla 9.1.

Tabla 9.1. Lista de controladores para ADODB y su nombre en el objeto conexión.

Nombre del DBMS	Nombre del controlador
Microsoft Access	access
Microsoft Access usando ADO	ado_access
Informix	informix (use informix72 si el controlador es anterior a Informix 7.3)
Microsoft SQL Sever 2000 o superior	mssql
Oracle	oracle
PostgreSQL versión 8.0	postgres8
SQLite	sqlite
Sybase	sybase
Extensión DB2 para PHP	db2

```

* @author Equipo de desarrollo SIMUPAT <simupat@walla.com>
* @package Interfaz_Datos
*/

/**
 * Incluye el archivo de (@link config.inc configuración).
 */
require_once(dirname(__FILE__).'/../config.inc');

/**
 * Incluye la libreria de (@link www.sourceforge.net/adodb ADODB).
 * El paquete ADODB es el responsable de comunicarse con la base de datos.
 */
require_once(SIMUPAT_DIR.'/../../adodb/adodb.inc.php');

// Creamos el objeto conexion
// El parámetro tomado por el objeto ADONewConnection de ADODB se refiere al driver de la base de datos
//en este caso es MySQL con control de transacciones activado.
$_SESSION['conexion'] = @ADONewConnection('mysql');

// Conectamos a la base de datos
// PRIMER PARAMETRO: Nombre DSN del servidor donde está el demonio de MySQL
// SEGUNDO PARAMETRO: Nombre de usuario de la base datos
// TERCER PARAMETRO: Contraseña para acceder a la base de datos (NOTA: Se recomienda que esta contraseña pa
por un algoritmo HASH )

//CUARTO PARAMETRO: Nombre de la base de datos
$_SESSION['conexion']->PConnect(localhost,simupat,tapumis,dbsimupat);

//La siguiente linea de código evita conflictos con el modo en que los cursores ADODB recorren la base de
$_SESSION['conexion']->SetFetchMode(ADODB_FETCH_NUM);

```

Figura 9.4. Captura de pantalla del código del archivo conexion.inc.

Más abajo en el código del objeto conexión, cambie los parámetros “dbsimupat” por el nombre de la base de datos creada, “simupat” por el nombre del usuario de la base de datos y “tapumis” por la contraseña. Guarde los cambios en el archivo.

Ahora abra el contenido de los archivos de volcado que crean las tablas de SIMUPAT, ubicados en la carpeta /protected/kernel/instalacion/mysql y que se llaman *crear_tablas_simupat.sql* y *llenar_tablas_simupat.sql*. Ejecute el contenido SQL de estos archivos, primero el crear tablas y luego el llenar tablas, en la consola de su manejador de bases de datos.

Una vez que haya seguido estas instrucciones, necesitará de la creación manual de un usuario administrador. Escoja un nombre de usuario, típicamente “root”, y escoja una contraseña a la cual le aplicará el algoritmo HASH MD5. Ahora, en el directorio /protected/pages/usuarios en el archivo lista_usuarios.xml, introduzca el usuario de la siguiente manera.

```
<?xml version="1.0" encoding="utf-8"?>
<module id="users">
<user name="root" password="b2c02e55772de785844e79033bf4f885" />
<role name="ADMINISTRADOR_SISTEMA" users="root" />
</module>
```

Figura 9.5. Captura de pantalla que muestra el contenido inicial del archivo lista_usuarios.xml.

Reemplace “root” por el nombre de usuario y en el campo password coloque el resultado de aplicar el resumen HASH/MD5 en formato hexadecimal. Guarde el archivo.

Ahora, en la consola de su manejador de bases de datos, escriba lo siguiente:

```
INSERT INTO usuarios VALUES ('login', 'nombre_usuario', ADMINISTRADOR_SISTEMA, email,
clave, fecha_caducidad);
```

En donde login se refiere al nombre de usuario, “nombre_usuario” a su nombre completo, “email” a su dirección de correo electrónico, “clave” al resultado del resumen HASH y “fecha_caducidad” a cualquier fecha futura.

Ahora ingrese al sistema e inicie sesión, como muestra la figura 9.6 y ejecute la “Ayuda” que le mostrará como crear los demás usuarios y a crear copias de seguridad.

Inicio | Iniciar Sesión | Sugerencias | Ayuda

Nombre de Usuario:

Contraseña:

Figura 9.6. Fragmento de la captura de pantalla del inicio de sesión.

9.1.2 Proceso de instalación usando el programa de instalación.

El proceso de instalación descrito en la sección 9.1.1 es necesario por si va a utilizar un manejador de bases de datos diferente a MySQL o si ha perdido los archivos del programa de instalación. Si este no es el caso, los autores recomiendan el uso del programa de instalación suministrado.

Efectué los mismos procedimientos que con la Instalación manual, hasta llegar a donde se digita la URL: *http://localhost/camino_a_simupat/simupat?page=instalacion.instalacion* y si observa lo que la figura 9.2 muestra, podrá utilizar el programa de instalación.

Lea los términos del contrato de licencia y pulse “Siguiente”, en la pantalla que sigue diligencie el formulario con los datos del usuario de la base de datos que usted ha creado, y con el nombre de dicha base de datos.

Parámetros del usuario de la base de datos

Dominio (localhost si la base de datos está en el mismo servidor que SIMUPAT):

Nombre de la base de datos:

Nombre de usuario:

Contraseña de acceso:

Figura 9.7. Formulario de configuración de la base de datos en el programa de instalación.

A continuación el sistema intentará crear las tablas de datos que serán usadas por SIMUPAT. Antes de pulsar el botón *Crear tablas del sistema* verifique

- NO hay restos de instalaciones previas de SIMUPAT en la base de datos.
- (En caso de re-instalación) Tiene copias de seguridad actualizadas.

Las tablas de SIMUPAT fueron creadas con éxito

Figura 9.8. Pantalla que informa que las tablas han sido creadas con éxito.

Una vez diligencie el formulario, de clic en “Continuar” y luego en “Crear tablas”. Si hay algún error, se imprimirá un cuadro de excepción que muestra el error y el contenido del trazado de la pila de ejecución. Si se observa el volcado del objeto conexión, se podrá leer el mensaje de error retornado por el DBMS:

Luego de recibir el mensaje informativo pulse “Continuar” y el sistema lo llevará a una pantalla con dos opciones disponibles. Seleccione la opción “Recuperar de una copia de seguridad” o “Efectuar una instalación nueva”. Seleccione la segunda, ya que la primera está documentada en el módulo de ayuda.

Creacion del usuario administrador

Ahora debe especificar un usuario que será el administrador del sistema. Para ello diligencie todos los campos del formulario mostrado a continuación.

NOTA: El login para el administrador será *root*

Datos del administrador

Nombre completo:

Clave:

Confirmar la clave:

Correo electrónico:

Figura 9.9. Formulario de creación del primer administrador del sistema.

Diligencie los datos y pulse “Crear administrador”. Borre la carpeta que contiene este programa, o sálvela en una ubicación segura, e inicie sesión (nombre de usuario: root) para crear los demás usuarios del sistema.

9.1.3 Gestión de configuraciones

Las configuraciones de SIMUPAT se dividen en dos: los archivos XML y los archivos INC. No se hablará en esta sección de todos los parámetros configurables, lo que requiere la lectura del *Manual del Programador*, sino de los que forman parte del ajuste final del proceso de instalación.

Se recomienda tener varias copias del archivo de configuración y guardar el original. SIMUPAT por defecto ha sido configurado para trabajar bajo los requerimientos del departamento de patología. Un ajuste obligatorio después de cada instalación es el siguiente.

Abra el archivo de configuración config.inc ubicado en /protected/kernel y mire la definición de la constante URL_SIMUPAT ubicada al final de dicho archivo. Coloque el nombre de dominio (sin incluir el /simupat) de su servidor Web y guarde el archivo.

Por último, abra el archivo "application.xml" y ponga el atributo "Mode" en el modo "Performance". Esto hará que el sistema sea mucho más rápido, aunque si experimenta dificultades, vuelva a poner el sistema en modo "Debug".

9.2 MANUAL DE USUARIO

El manual de usuario de SIMUPAT está en formato de película SWF o Flash y se accede de forma autónoma o mediante el uso del vínculo "ayuda" ubicado en la presentación gráfica de SIMUPAT. Estos manuales ilustran de forma gráfica y animada la operación normal de las funciones más complicadas del sistema, de modo que el usuario sepa exactamente que hacer en cada caso. Los archivos de ayuda gráfica se encuentran en el directorio /ayuda y cuentan con botones de navegación para su visualización paso a paso y con globos informativos que muestran las instrucciones en cada caso.



Figura 9.10. Captura de pantalla de una animación del Manual de Usuario.

9.3 MANUAL DEL PROGRAMADOR

El manual del programador es la documentación más extensa construida por los autores y que documenta cada clase, cada método (con sus entradas y salidas), cada excepción, atributos y paquetes que fueron desarrollados para construir SIMUPAT. Esta ayuda está construida en formato de página Web y cuenta con elementos para su navegación, así como numerosos hipervínculos que referencian de forma cruzada los elementos del manual.

Para acceder a la ayuda, ingrese a la carpeta “manual_del_programador” dentro de la distribución de SIMUPAT y ejecute el archivo “index.html”. De ser necesario, desactive temporalmente la protección contra código script para poder operar con el árbol dinámico de componentes, en este punto observará la pantalla de la Figura 9.11.

Utilizar el manual es muy fácil, cada paquete se compone de una colección de clases las cuales corresponden a las diferentes capas de SIMUPAT. Las clases cuyo paquete sea “Interfaz_presentación” son construidas para PRADO y por ello su padre es por lo general “TPage” (PRADO no está documentado en este manual)

Paquetes disponibles

- [Configuración](#)
- [Controles](#)
- [Excepciones](#)
- [Interfaz_Datos](#)
- [Interfaz_Presentacion](#)
- [Interfaz_Presentacion.administracion_piezas](#)
- [Interfaz_Presentacion.administracion_practicas](#)
- [Interfaz_Presentacion.consulta](#)
- [Interfaz_Presentacion.instalacion](#)
- [Interfaz_Presentacion.inventario](#)
- [Interfaz_Presentacion.usuarios](#)
- [Reglas_del_Negocio](#)
- [Reglas_Negocio.instalacion](#)
- [Utilidades](#)

Navegación: Interfaz_Presentacion

- Ayuda : Interfaz_Presentacion
 - Índice de elementos
 - Funciones
 - Clases
 - Archivos

[phpDocumentor v 1.3.0RC4](#)

Plantilla de HTML inspirada por [PHPEdit](#)

Manual del Programador SIMUPAT 1.0

Bienvenido a SIMUPAT!

Esta documentación fue generada por el Equipo de Desarrollo SIMUPAT usando: [phpDocumentor v1.3.0RC4](#)

Figura 9.11. Captura de pantalla de la página principal del Manual del Programador.

El árbol dinámico de navegación está diseñado para navegar fácilmente entre las clases, archivos y elementos que componen cada paquete. Utilice los controles del árbol para expandirlo o contraerlo según convenga.

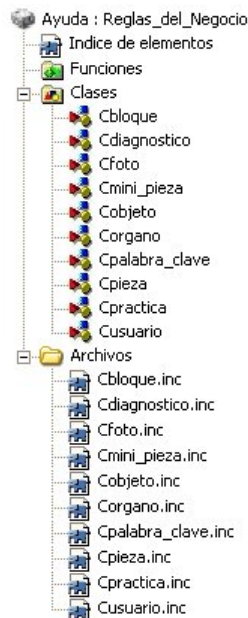


Figura 9.12. Árbol de navegación del Manual del Programador, expandido para mostrar las clases de la capa "Reglas del negocio".

Además, el manual siempre incluye para cada clase el parentesco con otras clases del sistema y además el árbol de clases de todo SIMUPAT.



Figura 9.13. Árbol de clases que muestra las clases que heredan de la clase CObjeto.

Para cada clase, el manual brinda una descripción textual de la clase, las variables miembro que la componen (con una descripción también), los métodos debidamente documentados, y el archivo en donde está implementada.

Paquetes disponibles

- [Configuración](#)
- [Controles](#)
- [Excepciones](#)
- [Interfaz Datos](#)
- [Interfaz Presentacion](#)
- [Interfaz Presentacion.administracion_piezas](#)
- [Interfaz Presentacion.administracion_practicas](#)
- [Interfaz Presentacion.consulta](#)
- [Interfaz Presentacion.instalacion](#)
- [Interfaz Presentacion.inventario](#)
- [Interfaz Presentacion.usuarios](#)
- [Reglas del Negocio](#)
- [Reglas Negocio.instalacion](#)
- [Utilidades](#)

Figura 9.14. Paquetes disponibles en la documentación contenida en el Manual del Programador.

A su vez, cada método está descrito con sus entradas y salidas, con las excepciones que puede disparar, la línea dentro del código fuente en donde se define dicho método y su visibilidad. Los parámetros opcionales están notados entre corchetes cuadrados en donde se indica el valor por

defecto y los métodos que sobrescriben o son sobrescritos en clases hijas o padres son incluidos con referencia a su documentación.

Clase Cpieza
(line 39)

Descripción	Variables de la clase	Constantes de la clase
Métodos	Variables heredadas	Métodos heredados

Cobjeto

```

|
|--Cmini_pieza
|
|--Cpieza
    
```

Implementada en el archivo:
[/protected/kernel/reqlas_negocio/Cpieza.inc](#)

Clase CPieza Esta clase maneja lo relacionado con las piezas del museo, y sus diferentes componentes tales como descripciones, fotografías, historia clinica, nombre, etc.

Para más información consulte la E.R.S en particular la sección de Requerimientos de Información.

- **abstract:** -

Clases heredadas de Cpieza:

Cpieza_db
Clase CPieza_DB Esta clase es la responsable de tareas tales como Cargar, Guardar, y Modificar Piezas en la base de datos.

Figura 9.15. Descripción de la clase Cpieza en el Manual del Programador.

Clase listador_mini_piezas_administracion
(line 21)

Descripción	Variables de la clase	Constantes de la clase	Métodos
Variables heredadas	Métodos heredados		

Resumen:
[CMD_AGREGAR_A_PRACTICA](#)
[CMD_BLOQUES](#)
[CMD_ELIMINAR](#)
[CMD_MODIFICAR](#)

- 🔹 **CMD_AGREGAR_A_PRACTICA** = 'Agregar_a_practica' (line 28)
- 🔹 **CMD_BLOQUES** = 'Bloques' (line 30)
- 🔹 **CMD_ELIMINAR** = 'Eliminar' (line 29)
- 🔹 **CMD_MODIFICAR** = 'Modificar' (line 27)

Nombres de comandos que el listador_mini_piezas_administracion reconoce

Figura 9.16. Relación de constantes en la clase listador_mini_piezas_administracion.


Clase Cpieza

(line 39)

Descripción	Variables de la clase	Constantes de la clase	
Métodos	Variables heredadas	Métodos heredados	


Resumen:

- array [\\$m_bloques](#)
- array [\\$m_diagnosticos_secundarios](#)
- array [\\$m_fotos_HC](#)
- array [\\$m_fotos_macro](#)
- array [\\$m_fotos_micro](#)
- string [\\$m_historia_EN](#)
- string [\\$m_historia_ES](#)
- array [\\$m_palabras_clave](#)

 [\\$m_bloques](#) (line 96)
Tipo : array

Vector para el almacenamiento de los bloques de resina asociados.

- **access:** - private

 [\\$m_diagnosticos_secundarios](#) (line 47)
Tipo : array

Vector para el almacenamiento de los diagnósticos secundarios

- **access:** - private

Figura 9.16 Descripción de las variables de la clase de Cpieza.

Clase Cpieza

(line 39)

Descripción	Variables de la clase	Constantes de la clase	Métodos
Variables heredadas	Métodos heredados		

Resumen de variables miembro heredadas por esta clase
Heredada de la clase [Cmini_pieza](#)






-  [Cmini_pieza::\\$foto_principal](#) - Foto Principal
-  [Cmini_pieza::\\$m_codigo_pieza](#) - Código de la pieza.
-  [Cmini_pieza::\\$m_descripcion_EN](#) - Descripción.
-  [Cmini_pieza::\\$m_descripcion_ES](#) - Descripción.
-  [Cmini_pieza::\\$m_diagnostico_principal](#) - Diagnostico

Figura 9.17. Relación de las variables heredadas de Cmini_pieza por la clase Cpieza en el Manual del Programador.

Toda entidad que pertenezca a la API de SIMUPAT está debidamente documentada y con vínculos en todo el Manual que la referencian. El lector se dará cuenta que en una versión impresa esto se perdería, y sería mas tedioso consultar la documentación; el HTML en cambio, es perfecto para esta tarea.

Clase Corgano
(line 30)

Descripción	Variables de la clase	Constantes de la clase
Métodos	Variables heredadas	Métodos heredados

Resumen:
Corgano [__construct](#) (*integer* \$codigo, *string* \$español, *string* \$ingles)
void [establecer_codigo](#) (*integer* \$codigo)
void [establecer_nombre](#) (*string* \$nombre, *string* \$idioma)
integer [obtener_codigo](#) ()
string [obtener_nombre](#) (*string* \$idioma)

Constructor [__construct](#) (line 59)
Corgano [__construct](#)(*integer* \$codigo, *string* \$español, *string* \$ingles)

Constructor de la clase

Parametros

- **integer \$codigo:** Código o ID del órgano
- **string \$español:** Nombre en Español
- **string \$ingles:** Nombre en Ingles

Información adicional:

- **access** - public

Figura 9.18. Relación de los métodos de Corgano en el Manual del Programador.

Clase Cpieza
(line 39)

Descripción	Variables de la clase	Constantes de la clase
Métodos	Variables heredadas	Métodos heredados

Resumen de los métodos heredados

Heredado de la clase [Cmini_pieza](#)


- ◆ [Cmini_pieza::establecer_codigo\(\)](#) - Función que establece el código de la pieza
- ◆ [Cmini_pieza::establecer_descripcion\(\)](#) - Función que establece la descripción de la pieza según el idioma
- ◆ [Cmini_pieza::establecer_diagnostico_principal\(\)](#) - Función que establece el diagnóstico principal de la pieza
- ◆ [Cmini_pieza::establecer_foto_principal\(\)](#) - Función que establece la foto principal que se va a mostrar
- ◆ [Cmini_pieza::obtener_codigo\(\)](#) - Función que devuelve el código de la pieza
- ◆ [Cmini_pieza::obtener_descripcion\(\)](#) - Función que obtiene la descripción que posee la pieza según el idioma
- ◆ [Cmini_pieza::obtener_diagnostico_principal\(\)](#) - Función para el diagnóstico principal
- ◆ [Cmini_pieza::obtener_foto_principal\(\)](#) - Función que devuelve la foto principal que se va a mostrar

Heredado de la clase [Cobjeto](#)

- ◆ [Cobjeto::comparar\(\)](#) - Compara un objeto con otro

Figura 9.19. Resumen de los métodos heredados por Cpieza de Cmini_pieza.

Aunque el formato HTML es muy bueno para consultar el Manual del Programador, es posible construirlo también en CHM (Formato de ayuda Windows) y en formas HTML que no usen marcos.

 **Método establecer_ruta** (line 132)
void establecer_ruta(string \$ruta, [boolean \$revisar_que_existe = true])

Establecer la ruta de la foto

Parametros

- **string \$ruta**: Ruta de la foto
- **boolean \$revisar_que_existe**: Parametro utilizado para activar o desactivar la revisión de existencia de la foto en el servidor


Información adicional:

- **access** - public
- **throws** - ERROR_TAMAÑO_IMAGEN En caso que el argumento pasado a la función sea el nombre de un archivo cuyo tamaño excede el tamaño máximo permitido
- **throws** - ERROR_NO_JPG Cuando el argumento pasado a la función no es un archivo de imagen jpg
- **throws** - ERROR_NO_ARCHIVO En caso que el argumento pasado a la función sea un nombre de archivo iregular o inexistente en el servidor
- **throws** - ERROR_NO_CADENA En caso que el parámetro suministrado no sea una cadena


Figura 9.20. Detalle de la documentación del método establecer_ruta() en la clase Cfoto.

File: /protected/kernel/reglas_negocio/Corgano.inc


Descripcion	Archivos incluidos (Includes/Requires)	Variables globales
Constantes	Funciones	

 **SIMUPAT_DIR."/excepciones/Cexcepcion.inc"** (line 23)
require_once : SIMUPAT_DIR."/excepciones/Cexcepcion.inc"

Incluye la clase [Cexcepcion](#)

 **SIMUPAT_DIR."/reglas_negocio/Cobjeto.inc"** (line 18)
require_once : SIMUPAT_DIR."/reglas_negocio/Cobjeto.inc"

Incluye la clase [Cobjeto](#)

 **dirname(__FILE__)."/../config.inc"** (line 13)
require_once : dirname(__FILE__)."/../config.inc"

Incluye el [archivo de configuración](#)

Figura 9.21. Resumen de los archivos incluidos (y clases) incluidas en el archivo Corgano.inc.

10 CONCLUSIONES

- La construcción de proyectos interdisciplinarios, por parte de estudiantes del área de Ingeniería de Sistemas con la colaboración de profesionales y personal administrativo de otras áreas, como los docentes y personal del Departamento de Patología, conlleva a la construcción de proyectos más ambiciosos que fortalecen la imagen institucional de la UIS. Lo anterior se vio reflejado en el presente trabajo de grado, en donde la construcción de un Sistema de Información (SIMUPAT) dio origen a una propuesta de innovación pedagógica que no sólo involucra la sistematización del Museo de Patología, sino también una reestructuración física de sus instalaciones. La realización de la propuesta brindará un marco favorable para la adopción de nuevas estrategias de aprendizaje significativo y de esta forma reducir el índice de mortalidad académica¹ presentado por la asignatura de patología.
- El sistema desarrollado se convertirá, a corto plazo, en una fuente de consulta por parte de la comunidad académica interesada. Esto es posible gracias a que la información anatomopatológica (fotografías y texto) estará disponible para la consulta en línea a través de Internet. De esta manera el presente trabajo de grado contribuye con la misión institucional de la Universidad Industrial de Santander, en lo referente a las premisas de formación de personas de alta calidad profesional y su propósito de generación y adecuación de conocimiento.
- El personal administrativo, docente y técnico del Departamento de Patología podrá usar y administrar de una forma más efectiva el museo debido a que el sistema mantiene un control sobre el inventario. Para ejercer dicho control el sistema ofrece características como la generación de reportes y la adopción del código de barras como medio para agilizar las operaciones administrativas y de mantenimiento.

¹ La asignatura de Patología presenta el cuarto mayor índice de mortalidad académica en la Facultad de Medicina de la UIS, luego de farmacología, morfología y bioquímica. El índice actual es del 10% según estudio realizado por la docente Dra. Olga Mercedes Álvarez.

- La creación de Comités de Sistemas de Información es una forma muy efectiva de establecer una relación estrecha de colaboración entre los ingenieros de sistemas desarrolladores y los usuarios del sistema. Prueba de ello es la Especificación de Requerimientos de Software (ERS) que los autores han construido con base en lo tratado en las reuniones semanales del comité de Sistemas de Información del Museo de Patología. Por otro lado, el comité planeó y ejecutó actividades extracurriculares cuyo objetivo fue recaudar fondos para la realización del proyecto de innovación pedagógica. Los fondos recaudados hasta el momento fueron invertidos en la realización de este trabajo de grado, que constituye la primera fase del proyecto de innovación pedagógica.
- Las herramientas de software libre constituyen una buena alternativa cuando los recursos financieros son escasos. Además, su filosofía de código abierto contribuye a que mejoren continuamente y permite su adaptación a las necesidades del proyecto. Sin embargo se debe tener en cuenta que no todas las licencias de software libre son compatibles con las leyes de derechos de autor y otros tipos de licencia de software; la licencia GNU/GPL, por ejemplo, obliga que el producto desarrollado sea también distribuido con dicha licencia.

11 RECOMENDACIONES

En la realización de cualquier empresa, por pequeña que sea, siempre se aprende algo nuevo. Siempre hay equivocaciones, problemas y situaciones que surgen inesperadamente y que al final se quisieran mejorar. Siempre hay pequeños descubrimientos sobre como hacer mejor las cosas y prever los problemas que aparecerán. Siempre aparecen las recomendaciones. Esta sección responde a una gran pregunta: ¿Qué aprendieron los autores realizando este trabajo de grado?

El contenido de esta sección refleja el punto de vista personal de los autores y su propósito es el siguiente: compartir su experiencia para que el lector no cometa los mismos errores y se beneficie de los aportes. En este momento, cuando los autores están a punto de iniciar su vida profesional, es cuando se debe dejar un aporte, por pequeño que sea, para que las promociones futuras de Ingenieros de Sistemas continúen expandiendo el amplio horizonte del conocimiento en este ramo.

El aporte de los autores se centra en un punto concreto: el desarrollo de un sistema de información. El desarrollo de un sistema de información es un proceso muy complejo, mucho más de lo que se puede imaginar en un principio y de lo que dicen los textos, y es un proceso en el cual los Ingenieros de Sistemas son tan solo una parte del gran recurso humano necesario para su realización. En SIMUPAT fue fundamental la creación del Comité de Sistemas de Información (véase la sección 3.3) y las ideas que dicho comité aportó, en forma de requerimientos, fueron decisivas para su realización.

En vista de la complejidad del proceso de desarrollo de un sistema de información, no es de extrañarse que existan muchas metodologías para coordinar su realización. ¿Pero qué es una metodología? Es una forma de hacer las cosas, sólo eso. De ninguna manera una metodología es una camisa de fuerza y no siempre la metodología más famosa es la más adecuada. Este es el caso del UML y el proceso unificado, que el mercado actual suele promocionar con ahínco, los cuales son empleados en proyectos en donde su uso no vale la pena y se vuelven ataduras. El

UML es una herramienta, no una obligación; es un lenguaje de modelado y no una calcomanía “Desarrollado con UML”.

En el caso de SIMUPAT el proceso unificado de Rational (RUP), metodología que usa UML, fue una gran ayuda a la hora de llevar a cabo el desarrollo del proyecto desde sus primeras fases, cuando SIMUPAT era una idea y el plan de trabajo estaba por construirse, cuando los autores comenzaban prácticamente de cero a imaginarlo todo. El plan de trabajo, conocido en el ámbito académico como “Plan de proyecto”, fue abordado por los autores como un verdadero plan de trabajo y no como un simple requisito académico. Los autores invirtieron mucho tiempo en establecer un cronograma de trabajo, en definir las fases de desarrollo, en preparar el ambiente de trabajo, en coordinar las reuniones con el comité, en prepararse a usar la metodología para el enorme trabajo que se avecinaba.

El utópico cronograma de actividades por días, muy bonito sobre el papel y que costó varios días de trabajo, se hizo añicos cuando los autores comenzaron el proceso de desarrollo. A pesar de trabajar disciplinadamente ocho (a veces más) horas diarias, varios inconvenientes y pequeños retrasos hicieron que se perdieran días de trabajo, y los autores no tardaron en verse apurados y cortos de tiempo. El plan falló y tuvo que idearse una solución a este gran problema. La inexperiencia de los autores, en ese entonces, en el arte de predecir cuanto tiempo debe invertirse en el desarrollo de cada módulo, fue un factor decisivo para que se originara esta situación. A mediados de agosto de 2006, los autores no creían que fueran a entregar a tiempo.

Aunque se falló en la planeación de las actividades, los autores encontraron un recurso muy útil para ahorrar tiempo en el diseño de interfaces gráficas de usuario, y a la larga en la elaboración de requerimientos: los diseños de pantallas en papel.

Han leído bien: en papel. A pesar de esta era electrónica que influye en nuestras vidas, el papel fue una de las herramientas más importantes para los autores. Cuando las reuniones con el comité de sistemas de información se atascaron en la discusión de un único punto (el que dio origen a los “bloques de resina”) los autores observaron una dificultad en hacer que el usuario imaginara el sistema y por lo tanto lo que espera de éste. Algunos lectores dirán: “la solución son los prototipos”

y tienen razón, aunque la elaboración de un prototipo toma mucho tiempo, tiempo que los autores no tenían. Por lo tanto, en hojas de papel reciclado, los autores dibujaron un esquema básico, un boceto, de una interfaz gráfica de usuario que mostrara los casos de uso “problemáticos” y cual sería su sorpresa al ver que los usuarios miembros del comité entendían mucho mejor con una simple mirada a los dibujos que con horas de explicación hablada. La elaboración de un dibujo tomaba menos de diez minutos. Pero esto no fue lo mejor, los usuarios comenzaron a dibujar controles y anotaciones sobre los bocetos lo cual era una traducción muy fiel de lo que querían y que fue vital a la hora de construir la Especificación de Requerimientos de Software que forma parte de este libro.

Pero, como se ha expresado en esta sección, una metodología es una forma de hacer las cosas, no una camisa de fuerza. Las pantallas en papel sirven para elaborar bocetos y diseñar (los diagramas UML fueron pintados primero en papel) pero son insuficientes para elaborar una documentación elegante y bien hecha. Elaborar una documentación de cada clase, cada método y cada variable es un trabajo ciclópeo si se realiza en un editor de texto y un “elefante blanco” si se imprime; por esta razón el manual de programador fue elaborado en formato de página Web.

Interactividad es lo que hizo famoso al HTML y es lo que cualquier programador quisiera tener cuando lee la documentación de una API. ¿De qué sirven quinientas páginas de texto impreso con letra bonita, si tengo que ser un mago para seguir una referencia cruzada cuya distancia, medida en páginas, entre los dos ítems referenciados son de por lo menos doscientas páginas? Un documento así, además de ser un desperdicio notable de tinta y papel, es francamente un estorbo. Por esta razón el manual de la API de SIMUPAT, en donde cada clase y cada método están documentados, está en formato HTML para que con solo hacer clic, el lector de dicho manual vaya de una referencia a otra, de un método a otro, de una clase a otra, o navegar las clases guiándose por los paquetes, en segundos. A propósito, el manual del programador escrito en PDF puede llegar fácilmente a las quinientas páginas.

Un manual tan exhaustivo no puede ser escrito usando una herramienta convencional. Colocarse frente a un programa de edición de páginas Web a hacer decenas de páginas de un manual es poco inteligente; la solución son las herramientas CASE de documentación.

Herramientas CASE, un término que significa “facilidad” mas no “fácil”. Aunque los autores utilizaron una herramienta CASE para generar las páginas Web del manual del programador, esto no quiere decir que sea una labor fácil, menos cuando se tienen decenas de clases, centenas de métodos y quizá miles de variables y excepciones que documentar en miles de líneas de código. Se necesita una férrea disciplina para ir documentando a medida que se va programando, ir colocando parámetro por parámetro, función por función, pero es la única forma de construir una documentación útil.

La herramienta CASE utilizada está licenciada por software libre. El software libre, libre como en libertad, mas no como en cerveza gratis¹, ni es necesariamente gratis, ni es del todo libre, ni es “malo” como algunos lo quieren hacer ver. Los autores aprovecharon una gran ventaja del software libre y de código abierto sobre los de código cerrado²: se puede hacer con ellos lo que se quiera, mejorarlos, publicar las mejoras y redistribuirlos libremente. Esta ventaja hace que un programa mejore considerablemente en muy poco tiempo y que paquetes que iniciaron como un conjunto pequeño de clases, hoy sean frameworks y API's comparables, si no mejores, a los desarrollados con licencias de código cerrado. Ese fue el caso de PRADO, usado por los autores en SIMUPAT.

“No hay documentación o ayuda en el software libre” ¡Mentiras! Es obvio que no todo el software libre es bueno, pero eso no quiere decir que si se presenta algún problema no habrá ninguna parte en donde se va a encontrar ayuda. Existen miles de foros, sistemas de reporte de bugs, documentaciones generadas con herramientas CASE (Como la de SIMUPAT) y otro tipo de recursos que explotar si es necesario. Es cierto que no dan garantía alguna y que no responderán

¹ “Free as in freedom, not as in free beer” Frase acuñada por el movimiento de Software Libre para hacer claridad sobre el término inglés “free” que no implica “gratis”.

² Los autores usan la frase “código cerrado” para mostrar la oposición con las licencias llamadas Open Source o “Código abierto”.

si un desastre informático sucede, pero al menos no tienen letra pequeña en sus licencias de uso y en muchos casos son gratuitos.

Licencias, un término legal que se debe tener en cuenta durante la vida profesional, y que en nuestro país (Colombia), tristemente, es tema frecuente por causa de la piratería. Cuando los autores de SIMUPAT iniciaron el desarrollo del proyecto, decidieron que no iban a usar ningún software pirata en ninguna de las fases de su realización. Todo el software usado en SIMUPAT tiene licencia, desde el Windows¹ de los computadores de desarrollo (que venía con los equipos Dell adquiridos por el departamento de patología) hasta el Word² en el cual esta sección ha sido escrita. En el caso de herramientas más íntimamente relacionadas con SIMUPAT, se ha tenido cuidado de obedecer las respectivas licencias de uso.

Se ha dicho que el software libre no es del todo libre. La razón es que aunque nadie lea las licencias del software libre, dichas licencias existen, y en el caso de la GPL³ la norma es clara: todo proyecto o producto que incluya software licenciado GPL debe ser GPL también. Aunque los paquetes usados por SIMUPAT vienen con licencias más permisivas como la BSD⁴, los autores no han redistribuido ninguno de estos paquetes para no violentar ninguna norma legal.

Continuando con el problema del retraso de tiempo en la fase de desarrollo, los autores lograron recuperarse gracias a dos factores fundamentales: uno es la arquitectura de capas y otro es la implementación de excepciones.

Arquitectura de capas y programación orientada a objetos, términos que de cumplirse cabalmente ahorran enormes cantidades de tiempo a quienes los aplican. Este fue el caso de SIMUPAT, en donde el alto grado de encapsulamiento de sus clases y un diseño estricto por capas, fue decisivo

¹ Se refiere al popular sistema operativo de Microsoft.

² Se trata del procesador de texto de Microsoft Office.

³ General Public License, licencia original desarrollada por Richard Stallman cuando originó el movimiento de software libre.

⁴ BSD significa, en español, distribución de software de Berkeley. Esta es una licencia de software libre.

a la hora de la depuración; en vez de corregir código en todo los archivos del sistema, basta con corregir la clase que tiene el problema y el resto del código se actualiza de forma automática.

Pero para corregir, primero hay que detectar problemas. Los primeros problemas son detectados durante el desarrollo, cuando los desarrolladores miran el desempeño del código programado y preparan al sistema para las pruebas con los usuarios. Una vez que se sabe que hay un problema, ¿cómo averiguar su causa? ¿Cómo encontrar el código responsable del problema? La solución adoptada por los autores fue la implementación de las excepciones, clases especiales que cuando son instanciadas dicen en que línea, en que archivo, en que clase, cual fue el trazado de la pila de ejecución cuando se originó el error. Si el lector es programador sabrá imaginar la utilidad de hacer las cosas de esta forma, en vez de emplear deslucidos mensajes como: "Error".

Los errores del párrafo anterior son errores del sistema, de naturaleza grave ya que no dependen de una acción errónea por parte del usuario, sino de una validación que no atrapó el error en el momento oportuno o de un código defectuoso. Los mensajes de error al usuario son más bonitos, más estéticos, pero aún son errores y debe buscarse la forma de reducir su ocurrencia. Una forma es capacitar a los usuarios bajo un sistema de ayuda interactiva que le oriente los pasos correctos que debe seguir en su trabajo; aunque una interfaz gráfica de usuario sea muy bien diseñada es inevitable que el usuario se confunda y se debe hacer algo al respecto. Por lo tanto en vez de imprimir manuales de usuario que quedarán desactualizados con algún cambio en la interfaz gráfica y que de todas formas no muestran al usuario en forma clara como operar el sistema, los autores han optado por construir películas animadas (en flash) que muestren paso a paso al usuario como hacer las cosas. El resultado: un usuario más feliz y un ingeniero de soporte mucho más tranquilo.

Por último con la realización de este proyecto los autores pretenden mostrar como la implantación de tecnología informática puede beneficiar a los profesionales de otras áreas como la medicina, y en concreto la Patología. Una colección bien ordenada, disponible en Internet, brinda prestigio a cualquier institución y trae beneficios, quizá no cuantificables de forma pecuniaria, pero sí en calidad de la formación académica. Este proyecto es tan solo el primer paso, falta una implantación

definitiva y muchas pruebas para que SIMUPAT alcance una estabilidad completa y se observen los resultados. Estamos en el principio de un largo camino, mas no en el final de un proyecto.

BIBLIOGRAFÍA

ARBELÁEZ DE MONCALEANO, Ruby. El proceso de la investigación : I. Planeación. 3 ed. Bucaramanga : Universidad industrial de Santander, 1992. 127 p.

BARCODE ISLAND. Disponible en web: <<http://www.barcodeisland.com/>>

CÁRCAMO SEPÚLVEDA, José. Bases de Datos Relacionales : Un Enfoque Práctico de Diseño. Bucaramanga : Ediciones UIS, 1996. 199 p.

DURAN TORO, Amador y BERNÁRDEZ JIMÉNEZ, Beatriz. Metodología para elicitación de requisitos software. Versión 2.1. Sevilla : 2000. 67 p. Universidad de Sevilla. Facultad de Informática y Estadística. Departamento de Lenguajes y Sistemas Informáticos.

FOWLER, Martin y SCOTT, Kendal. UML Distilled. 2 ed. USA : Addison Wesley. 2000. 185 p.

GRUPO DE DOCUMENTACIÓN DE PHP. Manual de PHP. Disponible en: <<http://www.php.net/manual/es/>>

IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830–1998, Institute of Electrical and Electronics Engineers, 1998.
Es un estándar con recomendaciones para la especificación de requisitos software.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACIÓN. Documentación. Compendio de normas técnicas sobre documentación de Tesis y otros trabajos de grado.

MARCOS MORA, Mari Carmen. Pautas para el diseño y la evaluación de interfaces de usuario. En: Rovira, Cristófol; Codina, Luís; Marcos, Mari Carmen; Palma, María del Valle. Información y documentación digital. Barcelona: IULA; Documenta Universitaria. 2004. Disponible en Web: <http://www.mcmarcos.com/pdf/2004_pautas-iula.pdf>

MYSQLLAB. MySQL 3.23, 4.0, 4.1 Reference Manual. Disponible en: <<http://dev.mysql.com/doc/refman/4.1/en/>>

ORGANIZACIÓN MUNDIAL DE LA SALUD. International Statistical Classification of Diseases and Related Health Problems, 10th Revision, 1992.

Pressman, Roger, Ingeniería del Software : Un Enfoque Práctico. 5 ed. Madrid : McGraw-Hill, 2002. 601 p.

RUMBAUGH, James; JACOBSON, Ivar y BOOCH, Grady. El Lenguaje Unificado de Modelado. Madrid : Addison Wesley, 1999. 432 p.

RUMBAUGH, James; JACOBSON, Ivar y BOOCH, Grady. The Unified Software Development Process. USA : Addison Wesley, 1999. 463 p.

WIKIMEDIA FOUNDATION, INC. Wikipedia, la enciclopedia libre. Disponible en: <<http://es.wikipedia.org/>>

ANEXOS

- A.** Acta de entrega del sistema.
- B.** Prototipos en papel de la interfaz gráfica de usuario.

A. ACTA DE ENTREGA DEL SISTEMA

Acta de entrega del sistema de información basado en Web para apoyar la organización y exhibición de los especímenes anatomopatológicos pertenecientes al museo de patología UIS, SIMUPAT 1.0

Reunidos en las instalaciones del Departamento de Patología de la Universidad Industrial de Santander el día martes 17 de octubre de 2006, los integrantes del Comité de Sistemas de Información para el desarrollo de SIMUPAT certifican que el sistema:

Se construyó siguiendo estrictamente los requerimientos que el comité estableció durante la fase de análisis en las reuniones semanales con los estudiantes de ingeniería de sistemas autores del presente software.

Se terminó dentro del plazo establecido para la construcción de proyectos de grado, según la reglamentación actual que para tal efecto ha sido dictaminada por la Universidad Industrial de Santander (Un semestre académico).

Está en proceso de alimentación de su base de datos con el material fotográfico y textual (información anatomopatológica) de los especímenes almacenados en el Museo de Patología de la UIS; proceso del cual son enteramente responsables los residentes de Patología integrantes del comité, como parte de la realización de su tesis de Postgrado.

Ha superado las pruebas efectuadas por los residentes de patología, miembros del comité, y se encuentra completamente operativo para su puesta en funcionamiento.

Con base en lo expuesto anteriormente, el presente comité:

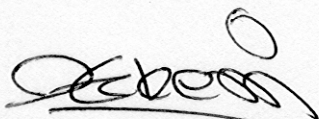
Otorga la calidad de "Terminado" al producto software entregado.

Acepta su plena conformidad con las características del producto entregado.

Acepta que los autores no son en absoluto responsables por la implantación y mantenimiento del producto entregado.

Acepta que la venta, redistribución y modificación del producto entregado debe contar con el consentimiento de los autores, obedeciendo al marco legal brindado por la UIS.

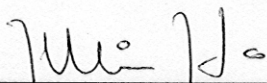
Firmado el día dieciséis del mes de Octubre del año 2006 por:



Dr. Jorge Humberto Echeverri Perico.
Director del Departamento de Patología UIS



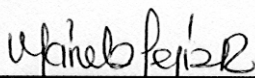
Dra. Olga Mercedes Alvarez Ojeda.
Codirectora del proyecto SIMUPAT.
Directora de la tesis de postgrado para la
organización del Museo de Patología.



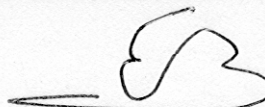
Dr. Julio Cesar Mantilla Hernández.
Director de la tesis de postgrado para la
organización del Museo de Patología.



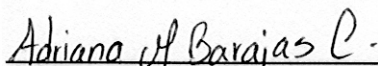
Dr. Pedro Vicente Duarte Ballén.
Residente del Departamento de Patología.
Autor de la tesis de postgrado para la
organización del Museo de Patología.



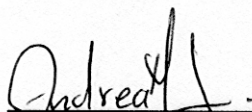
Dra. Mariela Liliانا Mejía.
Residente del Departamento de Patología.
Autora de la tesis de postgrado para la
organización del Museo de Patología.



Genaro Barchelot Gutiérrez
Técnico en procesamiento quirúrgico.



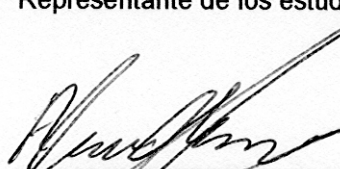
Adriana Marcela Barajas Carreño.
Estudiante de pregrado.
Representante de los estudiantes de Medicina.



Andrea Mileta González Gómez.
Estudiante de pregrado.
Representante de los estudiantes de Medicina.



Omar Saúl Duarte Ballén.
Autor de SIMUPAT.



Alvaro Martínez Duarte.
Autor de SIMUPAT.

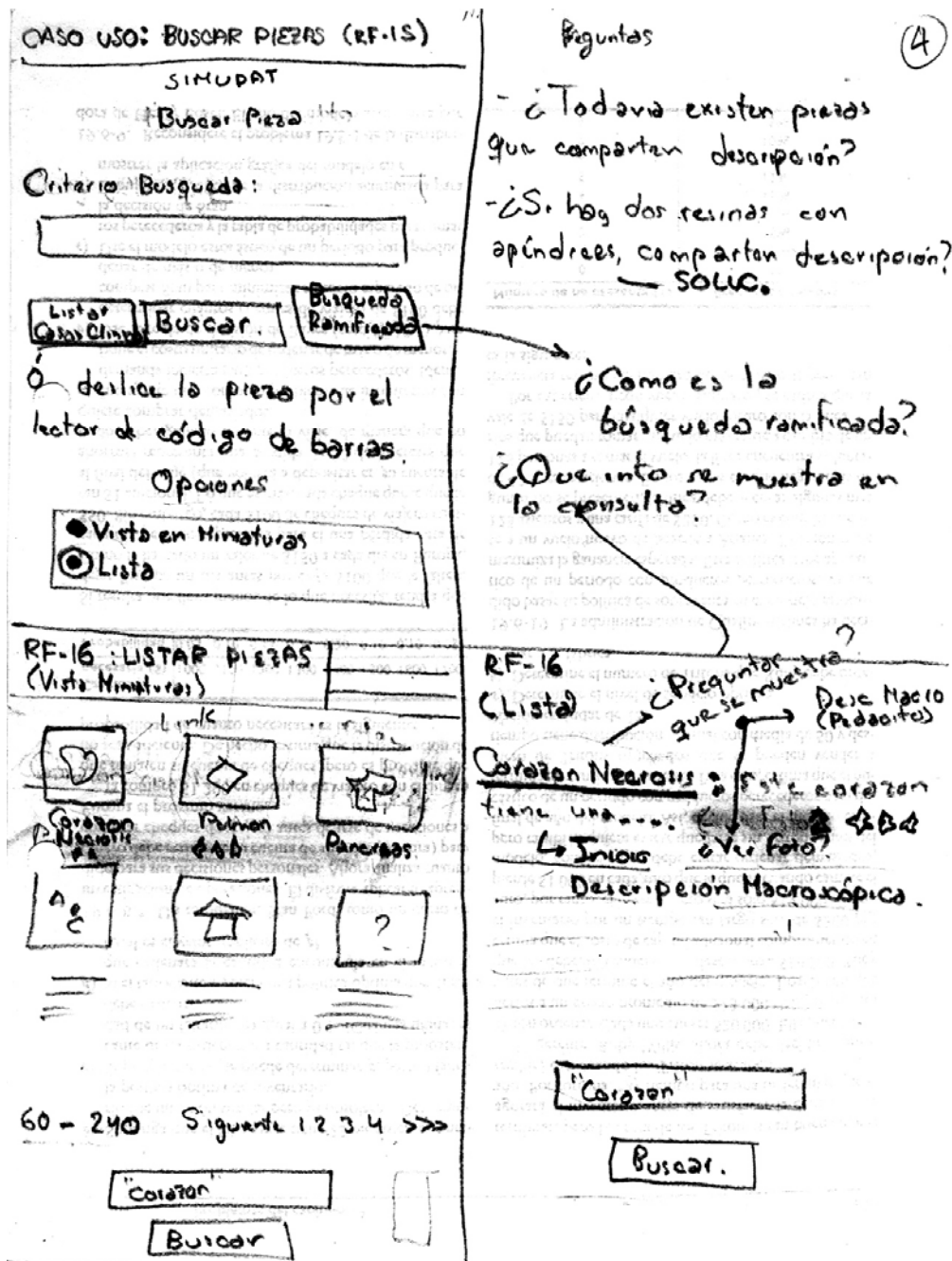


Figura B.2. Prototipo en papel de la interfaz gráfica de usuario para el caso de uso buscar piezas.