



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y
TECNOLOGÍAS DE INFORMACIÓN

“DESARROLLO DE MÓDULOS DE ADMINISTRACIÓN, PARA TAREAS DE
IMPLANTACIÓN Y MANTENIMIENTO DEL NUEVO SISTEMA DE FARMACIA 2.0
EN EL CESFAM VIOLETA PARRA DE CHILLÁN”.

MEMORIA PARA OPTAR AL TÍTULO
DE INGENIERO DE EJECUCIÓN EN
COMPUTACIÓN E INFORMÁTICA.

ALUMNO:
ANDRÉS EDUARDO VERA ACUÑA

PROFESOR GUÍA:
SR: ELVIS ARAVENA BELTRÁN

PROFESOR INFORMANTE:
SR. ALFONSO RODRÍGUEZ RIOS

CHILLÁN – DICIEMBRE DE 2009.

Agradecimientos

Gracias a Dios y a Nuestra Madre María porque sin la ayuda del cielo nada de esto habría sido posible.

Gracias a mis padres, por el firme propósito que me enseñaron a tener en todo lo que es bueno, a pesar de las dificultades del camino.

Gracias a mis hermanos y amigos por su apoyo, ánimo y sus oraciones.

Gracias a mis Profesores de la universidad, de todas las asignaturas que componen la carrera, especialmente a mi profesor guía el Sr. Elvis Aravena, por sus sabios consejos y oportunas sugerencias. Gracias a mi profesor informante el Sr. Alfonso Rodríguez, por su visión amplia en todos los aspectos de los sistemas, gracias a mi profesora Sra. Angélica Caro por sus aportes en las metodologías de migración; gracias a la profesora Jefa de Carrera Srta. María Antonieta Soto, por su paciencia para conmigo en estos años de estudio, gracias a la Sra. Pamela Zamudio, directora del CESFAM Violeta Parra por su apoyo y paciencia, gracias a Claudio Torres por su apoyo y entrenamiento en programación, gracias a muchos compañeros y compañeras que me apoyaron con una palabra amable dándome fuerzas y a tantas personas que conocí en el camino de estos años y que no pude agradecerles tanta ayuda que me brindaron en distintos momentos.

Gracias a la vida que me ha dado tanto...

Resumen

El Centro de Salud Familiar (CESFAM) Violeta Parra de Chillán es un establecimiento de atención primaria de salud, dependiente técnica y administrativamente del Servicio de Salud de Ñuble. En dicho CESFAM se entrega atención de salud a una población de aproximadamente 70.000 personas. Dentro de las prestaciones que el CESFAM otorga a las personas se encuentra la entrega o dispensación de medicamentos e insumos recetados por los profesionales de la salud, los cuales son gestionados a través de la farmacia del centro de salud. Para llevar a cabo su labor, el departamento de farmacia cuenta con recursos humanos especializados en la entrega de medicamentos, labor que se apoya en el sistema informático de control denominado "Sistema de Farmacia 1.0", el cual será reemplazado por el "Sistema de Farmacia 2.0" que reúne las funcionalidades del primero y agrega nuevas funcionalidades necesarias en farmacia.

Para comenzar a utilizar el nuevo sistema de farmacia es necesario migrar los datos históricos almacenados en el "Sistema de Farmacia 1.0" hacia el nuevo sistema de farmacia. Para solucionar la situación se aborda una metodología que involucra la creación de un software para la migración de datos.

El presente trabajo tiene el objetivo de mostrar el desarrollo de módulos de administración para las tareas de exportación, importación, backup y restauración, de datos para el "Sistema de Farmacia 2.0", a fin de lograr la implantación de dicho sistema en el "CESFAM Violeta Parra" de Chillán y dotarlo de funcionalidades que ayuden a mantener la integridad de los datos.

El trabajo comienza con el estudio de distintas estrategias de migración de sistemas informáticos, para luego conocer el sistema que se pondrá en marcha y así determinar el análisis, diseño y construcción del software. Se utiliza diseño orientado a objetos, diseño de base de datos relacional, programación Java bajo el marco de trabajo Struts para aplicaciones Web. Como resultado se obtiene una aplicación con módulos de administración para migrar los datos desde la base de datos del "Sistema de Farmacia 1.0" a la base de datos del nuevo sistema, además de dotar al nuevo sistema de las funcionalidades de crear respaldos de la base de datos y restaurarla a partir de los respaldos creados y posibilitar el cierre de transacciones a partir de una fecha dada moviendo datos históricos en la base de datos.

Índice.

Introducción.....	6
Capítulo 1: Situación Actual.....	8
Descripción de la Organización.....	8
Visión y Misión del CESFAM Violeta Parra de Chillán.	9
Recursos Humanos y Organigrama.....	10
Materiales e Infraestructura.....	13
Atención Brindada en el CESFAM.....	13
La Unidad de Farmacia.	16
Descripción General del Problema.....	19
1.2.1 Puesta en Marcha del Nuevo Sistema de Farmacia 2.0.....	19
1.2.2 Estandarización de la Información de Bases de Datos de los Sistemas de Información.....	20
1.2.3 Desarrollar Módulos para la Administración y Mantenimiento.....	21
Marco Teórico.....	22
1.3.1 Sistemas de Información Heredados.....	22
1.3.2 Estrategias de Migración e Implantación de Sistemas de Información.....	24
1.3.3 Metodología de Desarrollo Iterativo-Incremental.....	31
1.3.4 Enfoque de Desarrollo Orientado a Objetos.....	31
1.3.5 Aplicaciones Web.....	35
Herramientas de Diseño.....	37
1.4.1 Lenguaje Unificado de Modelado UML.....	37
1.4.2 El Framework Modelo Vista Controlador MVC.....	37
1.4.3 Diseño de Bases de Datos.....	39
1.4.4 Patrones de Diseño Abstract Factory, Data Access Object y Data Transfer Object. 42	
Herramientas de Desarrollo.....	45
1.5.1 Lenguaje de Programación Java.....	45

1.5.2	Servlets.	46
1.5.3	Java Server Pages JSP.	47
1.5.4	Framework Struts.	48
1.5.5	Middleware Hibernate.	50
1.5.6	Servidor de Aplicaciones Apache Tomcat.	50
1.5.7	Sistemas Gestores de Bases de Datos (SGBD) MS-Access 97 y MySQL 5.0. 52	
1.5.8	Entorno de Desarrollo Eclipse.....	55
Capítulo 2: Planteamiento de Soluciones.....		56
2.1	Objetivo General.	56
2.2	Objetivos Específicos.....	56
2.3	Estrategias para implementar soluciones.	57
Capítulo 3: Estudio de Factibilidad.		58
3.1	Factibilidad Técnica.	58
3.2	Factibilidad Operativa.	60
3.3	Factibilidad Económica.....	60
Capítulo 4: Estudio del Sistema Informático que se Pondrá en Marcha.		66
4.1	Requerimientos Funcionales y No Funcionales del Sistema de Farmacia 2.0 del CESFAM Violeta Parra.	66
4.2	Casos de Uso del Sistema de Farmacia 2.0.....	70
4.3	Diagrama de Clases.	71
4.4	Modelo de Datos.....	74
Capítulo 5: Tareas de Exportación y Mapeo de Datos desde “Sistema Heredado de Farmacia 1.0”.....		85
5.1	Exportación desde Base de Datos del “Sistema Heredado de Farmacia 1.0” (DBSFARM.mdb) en MS-Access 97 al formato de “dbfarmacia 2.0” en MySQL... 85	
5.2	Mapeo de Datos entre Bases de Datos dbfarmacia 1.0 y dbfarmacia 2.0.	91
5.3	Análisis, Diseño y Pruebas de la Aplicación para Migración, Chequeo de Integridad y Consistencia de Datos de “dbfarmacia10” y “dbfarmacia20” (Migración-FF).	93
5.3.1	Análisis de la Aplicación “Migración-FF”.....	93
5.3.2	Diseño de la Aplicación “Migración-FF”.....	96

5.3.3	Pruebas de la Aplicación “Migración-FF”	106
Capítulo 6: Tareas de Importación de Datos desde “Plataforma Virtual SOME” (dbsome) para Chequeo de Integridad y Consistencia con Sistema Heredado. 111		
6.1	Importación de Datos a MySQL 5.0.	111
6.2	Tablas Coincidentes de las Bases de Datos “dbfarmacia2.0” y “dbsome”.	112
6.3	Análisis, Diseño y Pruebas de la Aplicación para Migración, Chequeo de Integridad y Consistencia de Datos de “dbfarmacia20” y “dbsome” (“Migración-FS”).	113
6.3.1	Análisis de la Aplicación “Migración-FS”	113
6.3.2	Diseño de la Aplicación “Migración-FS”	115
6.3.3	Pruebas de la Aplicación “Migración-FS”	121
Capítulo 7: Desarrollo de Módulo de Administración y Mantenimiento para Backup.... 123		
7.1	Análisis de Requerimientos del Módulo Backup.	123
7.2	Diseño del Módulo Backup.....	125
7.3	Pruebas del Módulo Backup.	126
Capítulo 8: Desarrollo Módulo de Administración y Mantenimiento para Restauración.. 129		
8.1	Análisis de Requerimientos del Módulo Restauración.	129
8.2	Diseño del Módulo Restauración.	131
8.3	Pruebas del Módulo Restauración.....	133
Capítulo 9: Desarrollo Módulo de Administración para Cierre Anual de Transacciones.. 135		
9.1	Análisis de Requerimientos Módulo Cierre Anual de Transacciones.	136
9.2	Diseño Módulo Cierre Anual de Transacciones.	137
9.3	Pruebas Módulo Cierre Anual de Transacciones.	141
Capítulo 10: Puesta en Marcha del Sistema de Farmacia 2.0..... 143		
10.1	Capacitación a los usuarios.	143
10.2	Instalación del Sistema Informático.....	144
10.3	Evaluación de la Puesta en Marcha del Sistema Informático.	145
Capítulo 11: Conclusiones..... 148		
Bibliografía..... 149		
Anexos..... 153		

Introducción.

“La atención primaria de salud es la asistencia sanitaria esencial basada en métodos y tecnologías prácticos, científicamente fundados y socialmente aceptables, puesta al alcance de todos los individuos y familias de la comunidad, mediante su plena participación, y a un costo que la comunidad y el país puedan soportar, en todas y cada una de las etapas de su desarrollo. Forma parte integrante, tanto del sistema nacional de salud, como del desarrollo económico y social global de la comunidad. Es el primer nivel de contacto de los individuos, la familia y la comunidad con el sistema nacional de salud, llevando la atención sanitaria lo más cerca posible al lugar donde viven y trabajan las personas, y constituye el primer elemento de un proceso continuado de asistencia sanitaria.”. Se ha reconocido que la atención primaria de salud es la clave para alcanzar el desarrollo conforme al espíritu de la justicia social.” [OMS-UNICEF, 1978]

Cada establecimiento de asistencia primaria de salud, o centro de salud familiar (CESFAM), como es el caso del “CESFAM Violeta Parra” de Chillán, está organizado en distintas unidades, que hacen posible desarrollar el trabajo del centro de salud y cumplir con sus objetivos, entre estas unidades se encuentran las farmacias de dichos establecimientos.

La atención farmacéutica es el conjunto de prestaciones que se brindan a través de los servicios, insumos y medicamentos, integrados a la atención de salud del paciente, familia y comunidad. El objetivo de la atención farmacéutica es lograr la racionalidad en el uso de los medicamentos, mediante un adecuado suministro de estos; educando al equipo de salud y pacientes sobre su buen uso y estimulando la responsabilidad de cada persona en el uso racional de los fármacos para el cuidado del enfermo [MINSAL, 1995]. Para lograr el objetivo anterior es necesario además, contar con sistemas informáticos que apoyen la gestión de las farmacias y aporten información de control para la toma de decisiones. Dichos sistemas de gestión, fueron por muchos años llevados en forma manual, a través de planillas, libros de registro de entregas, inventario y tarjetas de dispensación de medicamentos a pacientes.

Posteriormente se implementaron sistemas computacionales donde fue posible automatizar en parte la gestión y control de las transacciones y entregas hechas en farmacia.

Hoy en día, cuando la cantidad de población que es atendida en el “Centro de Salud Familiar (CESFAM) Violeta Parra de Chillán” alcanza a más de 70.000 personas, es imprescindible contar con sistemas de información robustos que apoyen además toda la labor de los equipos multidisciplinarios de salud, es decir, que sean capaces de atender a múltiples usuarios e integrarse con otros subsistemas del mismo CESFAM.

El “CESFAM Violeta Parra de Chillán”, se encuentra actualmente frente al desafío de entregar un nuevo modelo de atención: El modelo de salud familiar, en el cual la atención aspira a ser integral y centrada en las personas y sus respectivas familias.

En el marco de este desafío se está implementando, hace 2 años, en el CESFAM, una plataforma de servicios informáticos que incorpora sistemas de información para la gestión de horas de atención sectorizada a pacientes, agenda de Profesionales de la Salud sectorizada, agenda de visitas domiciliarias por parte de Profesionales de la Salud hacia pacientes postrados, sistema de control de atenciones en Servicio de Atención Primaria de Urgencias (SAPU) y sistema de control de gestión de las farmacias del CESFAM. Todos los sistemas mencionados requieren estar integrados y esta integración trae consigo la necesaria migración de los sistemas más antiguos.

En el presente trabajo de Título se aborda el proyecto de la puesta en marcha (implantación) del sistema de control para las farmacias del CESFAM, lo que trae consigo, implementar una estrategia de migración de los datos desde el antiguo “Sistema de Farmacia 1.0” al nuevo “Sistema de Farmacia 2.0”. Adicionalmente el proyecto contempla dotar al nuevo software de funcionalidades para cerrar procesos anuales de transacciones, respaldar y recuperar la base de datos del sistema.

Se comienza con la revisión de la situación actual, descripción de las problemáticas involucradas, presentación de las soluciones y las tecnologías aplicadas, para luego pasar a

describir los procesos de desarrollo de los módulos de migración y administración del “Sistema de Farmacia 2.0”.

Capítulo 1: Situación Actual.

Descripción de la Organización.

Este capítulo es el primer acercamiento a la organización “Centro de Salud Familiar Violeta Parra” (CESFAM Violeta Parra), para la cual se desarrollan los módulos de administración para las tareas de implantación y mantenimiento del “Sistema de Farmacia 2.0”.

En primer lugar se describe la organización bajo distintos aspectos, para comprender mejor las actividades que se realizan en el CESFAM y las necesidades existentes; luego se hace una aproximación general al problema, objetivo del presente trabajo de título, para dar paso a la descripción general de las tecnologías usadas en la solución propuesta.

El CESFAM “Violeta Parra”; se clasifica como Centro de Salud Familiar de Atención Primaria, siendo un servicio público. Lo anterior significa, de acuerdo con los niveles de atención en salud definidos por el Ministerio de Salud de Chile, que es el primer nivel de atención de baja complejidad y alta cobertura de población.

La fundación de este consultorio se remonta al año 1992 y responde a la necesidad de separar los servicios de atención primaria de los del consultorio adosado de especialidades, ubicado al lado del Hospital Herminda Martín de la ciudad.

La ubicación del “CESFAM Violeta Parra” es calle Francisco Ramírez N° 150 de la ciudad de Chillán.

Es de especial relevancia destacar el rol importante que se ha asignado a las tecnologías de información y comunicación (TICs) en el consultorio, como

herramientas que permiten cumplir los objetivos de calidad de la atención, eficiencia y mejoramiento de la gestión de los recursos. Esto se explicita en los objetivos estratégicos para el periodo 2007 - 2010 que motivan los esfuerzos en el consultorio. En dichos objetivos del plan estratégico de la Dirección del CESFAM, se señala lo siguiente:

“Desarrollar una red asistencial articulada que asegure la continuidad de la atención integral de los usuarios/as utilizando herramientas modernas de gestión” e “implementar un sistema de control de gestión”.

Los servicios proporcionados por el “CESFAM Violeta Parra” van dirigidos potencialmente a 70.938 personas inscritas en sus registros. Y que tienen domicilio particular en una vasta área geográfica de la ciudad de Chillán repartida en 5 sectores los cuales son atendidos por equipos multidisciplinarios de Profesionales, Técnicos y Administrativos.

Visión y Misión del CESFAM Violeta Parra de Chillán.

La visión y misión del CESFAM reflejan el objetivo, el quehacer y motivación de la organización y son las siguientes:

Visión de la Organización:

“Contribuir al bienestar de la población del sector, con el fin de lograr familias sanas, en un entorno saludable como reflejo de un satisfactorio trabajo en equipo y en Red”.

Misión de la Organización:

“Consolidar el trabajo del Centro de Salud Familiar, trabajando en equipo, insertos en la comunidad del Sector”.

Recursos Humanos y Organigrama.

El CESFAM cuenta con una dotación de 213 personas que se desempeñan en diversas unidades, ordenadas según el organigrama de la figura N°1 y en los cargos que se muestran en la Tabla N°1.

Director	1	Asistentes Sociales	6
Subdirector	1	Técnicos	54
Médicos	20	Administrativos	54
Odontólogos	12	Chóferes	2
Químico Farmacéutico	1	Auxiliar	1
Psicólogos	7	Estafeta	1
Enfermeras	16	Guardias	4
Matronas	8	Personal Aseo	7
Nutricionistas	10	Personal Mantención	2
Kinesiólogos	6		
Total Dotación: 213 personas.			

Tabla 1: Recursos Humanos del "CESFAM Violeta Parra".

Gran parte del personal detallado en la tabla N° 1, actualmente se agrupa en seis equipos multidisciplinarios de profesionales repartidos así para atender las necesidades de las personas de cada sector del territorio asignado al CESFAM.

El Equipo Directivo del consultorio es el siguiente:

Directora Centro de Salud Familiar Violeta Parra:

Sra. Pamela Zamudio Villarroel - Químico Farmacéutico.

Subdirector Médico:

Sra. Ana María Moyano - Médico Cirujano.

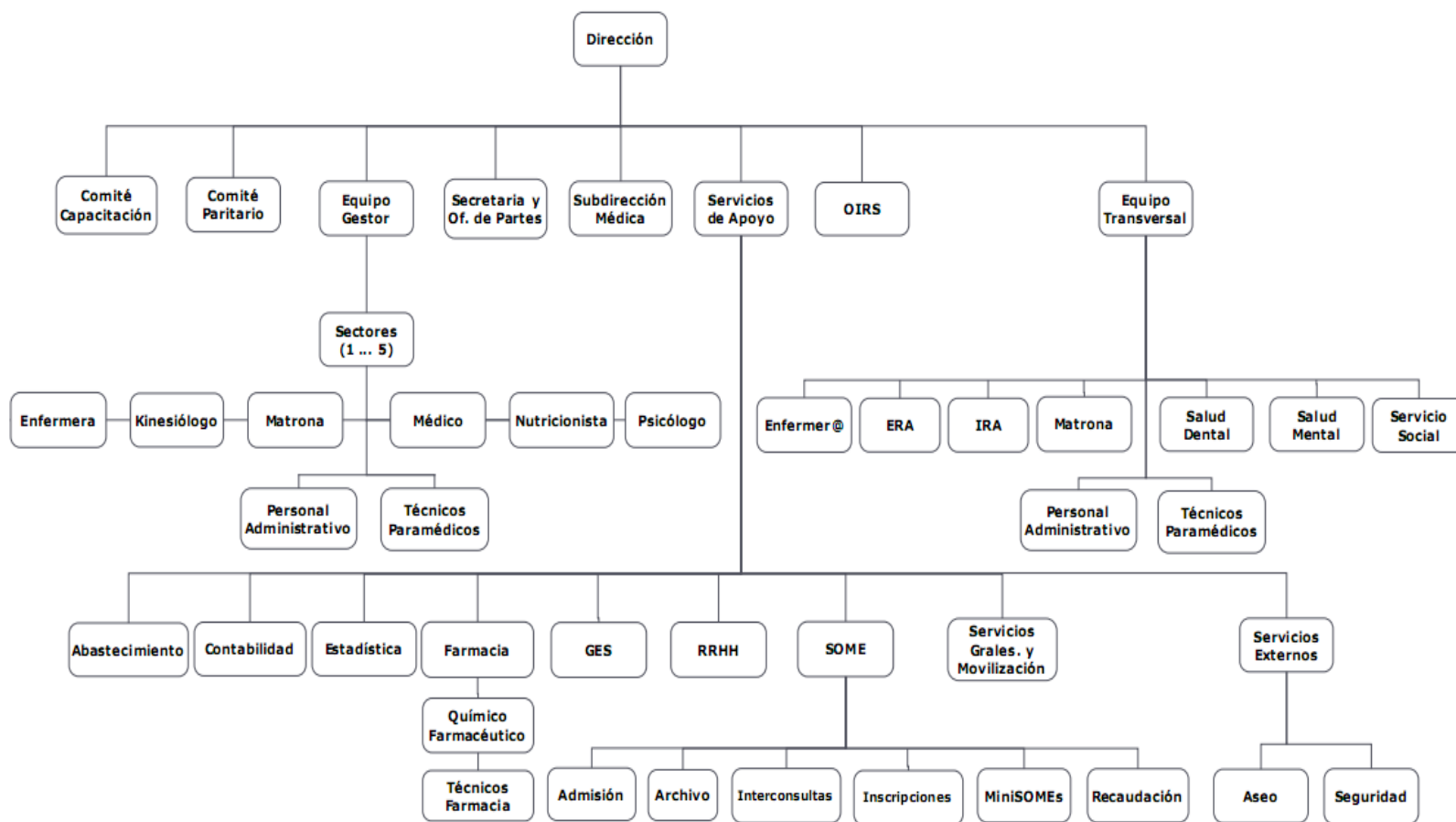


Fig. 1: Organigrama "CESFAM Violeta Parra de Chillán".

Materiales e Infraestructura.

Técnica y administrativamente el CESFAM es dependiente del Servicio de Salud Ñuble. Funciona como unidad presupuestaria y administrativa independiente desde el mes de mayo de 1995. Sus instalaciones son de albañilería reforzada, con una superficie de 1.665,21 m² construidas en un terreno de 4.860 m².

Atención Brindada en el CESFAM.

La Atención Primaria en Salud es la atención que ofrecen los consultorios, CESFAM y postas rurales a través de todo el país y es este tipo de atención la que se brinda en el CESFAM Violeta Parra. Dicha atención constituye la puerta de entrada al Sistema Público de Salud

Por su naturaleza de centro de atención primaria de salud, el CESFAM entrega atención de tipo ambulatoria a los distintos grupos etarios de la población correspondiente. Además existe en el Centro de Salud, desde Julio del año 2007, el Servicio de Atención Primaria de Urgencia (SAPU), el cual, permite la continuidad de la atención los 365 días del año y en horario nocturno, favoreciendo el acceso y oportunidad de atención de urgencias de baja complejidad a nuestros usuarios, tales como: intoxicaciones (por medicamentos, gas, alimentos, cloro), heridas, caídas, accidentes en el hogar, dolor intenso de cabeza, estómago o abdominal, mordedura de animales, síndrome febril, ataque de asma, cuadro diarreico, vómitos, Rush alérgico, crisis hipertensivas, picaduras de insectos, entre otras.

La atención brindada en el CESFAM comprende los siguientes programas de salud: [MINSAL 2007]

- Programa del Niño.

(Se considera niño a las personas comprendidas entre los 0 y 9 años).

1. Consulta de Morbilidad.
2. Consulta de Salud por Profesional Médico.
3. Colocación de Vacunas.
4. Control de Crónicos.
5. Control de Lactancia Materna.
6. Consulta Nutricional.
7. Programa Nacional de Alimentación Complementaria PNAC.
 - Programa del Adolescente.
(Se considera adolescente a las personas entre los 10 y los 19 años).
 1. Consulta de Morbilidad.
 2. Control de Puerperio.
 3. Control de Regulación de Fecundidad.
 4. Control Prenatal.
 5. Control Ginecológico.
 6. Consulta de Morbilidad Obstétrica.
 7. Consulta de Morbilidad Ginecológica.
 8. Control de Enfermos Crónicos.
 - Programa de la Mujer.
(Se considera a las mujeres entre 20 y 64 años).
 1. Control Prenatal.
 2. Control de Regulación de Fecundidad.

3. Control Ginecológico Preventivo.

4. Control de Menopausia y Climaterio.

5. Consulta Morbilidad Obstétrica.

6. Consulta de Morbilidad Ginecológica.

7. Consulta Nutricional.

8. Control de Crónicos.

- Programa del Adulto.

(Se considera adulto las personas comprendidas entre los 20 y 64 años).

1. Consulta de Morbilidad.

2. Consulta de Crónicos.

3. Consulta Nutricional.

- Programa del Adulto Mayor.

(Se considera población adulto mayor al grupo entre los 65 años y más).

1. Consulta de Morbilidad.

2. Consulta Nutricional.

3. Control de Salud.

4. Consulta de Salud Mental.

5. Vacuna Anti-influenza.

- Programa Odontológico.

A diferencia de los otros programas, donde se pretende atender las necesidades de la población de atención primaria, el Programa Odontológico

debe discriminar su atención dado los recursos disponibles. El abordaje de la situación de Salud Bucal de la población requiere aplicar enfoques de Salud Pública: basados en la priorización de las acciones promocionales y preventivas, eligiendo aquellas de mayor impacto y mejor costo-beneficio. Además, busca ser concordante con las políticas del Gobierno Central.

Las prestaciones del programa odontológico se desprenden de la “Propuesta de Incorporación del Componente de Salud Buco-Dental al Plan Garantizado de Salud” del Ministerio de Salud. Estas son:

Altas Integrales Niños de 2 a 4 años.

Altas Integrales Niños de 5 a 9 años.

Altas Integrales Niños de 10 a 14 años.

Altas Integrales Madres Primigestas.

Altas Integrales Adultos Mayores.

Atención de Urgencia.

La Unidad de Farmacia.

El Departamento de Farmacia es fundamental en el funcionamiento diario del CESFAM Violeta Parra, ya que su principal labor es dispensar los medicamentos que los profesionales de la salud prescriben a los pacientes. Según datos estadísticos contenidos en la “Cuenta Pública 2008” de este establecimiento de salud, el año 2008 se invirtieron \$434 millones de pesos en medicamentos y alimentos para los pacientes, lo que corresponde aproximadamente al 19% del presupuesto total anual del consultorio. [CVP, 2008].

La atención de farmacia se organiza para atender básicamente a dos tipos de pacientes, a fin de brindar una mejor y más ordenada atención a las

personas. Estos tipos de pacientes son “crónicos” y de “morbilidad”. A continuación se detalla su descripción:

- a) Paciente Morbilidad: Persona que acude al Centro de Salud en busca de atención médica en forma circunstancial a causa de enfermedad o sintomatología que no sufre periódicamente, o que al menos no ha sido diagnosticada como enfermedad crónica.
- b) Paciente Crónico: Persona que asiste mensualmente o a intervalos regulares, a atención de salud (controles), por haberse diagnosticado enfermedad crónica, y estar afecto a un programa de atención de enfermos crónicos y que por esta razón retira medicamentos de farmacia crónicos en forma regular y controlada.

Por otra parte el Departamento de Farmacia se preocupa de abastecer otras farmacias como son la farmacia SAPU que opera en horario extraordinario en dependencias del mismo CESFAM. También están consideradas las farmacias externas al CESFAM, como es el caso de algunas postas rurales que son abastecidas eventualmente en virtud del plan de redes asistenciales que articula el Ministerio de Salud a través de sus Secretarías Regionales y Servicios de Salud Regionales y/o Provinciales. Por último, dentro de las farmacias externas al Consultorio que son abastecidas, está la farmacia del “CECOF Padre Hurtado” (Centro Comunitario de Salud Familiar) creado por iniciativa del CESFAM y dependiente de él en su gestión y funcionamiento.

En las farmacias de morbilidad y crónicos laboran técnicos paramédicos, capacitados para orientar a los pacientes sobre el uso de los medicamentos que les son dispensados.

Las actividades de gestión de las farmacias en general, desde la perspectiva del flujo de artículos son: [DADES, 2008].

1. Selección de medicamentos e insumos: Relaciona el tipo de prestaciones de salud que se realizan en el establecimiento, con las necesidades. La multiplicidad de medicamentos disponibles en el mercado, debido al avance científico y tecnológico, hace necesario racionalizar su uso en base a:

- El Formulario Nacional de Medicamentos del MINSAL.
- Evidencias fármaco terapéuticas.
- Evidencias fármaco económicas.
- La inclusión de los fármacos de las canastas de patologías GES.
- La definición de arsenales farmacológicos locales (Servicio de Salud de Ñuble).
- Recomendaciones y Normas emanadas del ISP (Instituto de Salud Pública).
- La optimización de los recursos disponibles.

2. Programación. Es la determinación de las necesidades de medicamentos e insumos terapéuticos, representa uno de los problemas básicos del suministro por su influencia sobre la adquisición y oportunidad en la disponibilidad de los productos. Se debe relacionar con los presupuestos disponibles.

3. Abastecimiento: Adquisición de artículos, que relaciona las necesidades con los sistemas de gestión de suministro centralizada o en forma directa. (CENABAST, o compras a través de licitación pública por sistema mercado público).

4. Recepción Almacenamiento y Distribución: Relaciona los procesos específicos relativos a recepción, la conservación, la calidad de los

medicamentos e insumos terapéuticos, garantizando su disponibilidad para la oportuna distribución.

5. Valoración de costos de artículos, para relacionar el monto real de los medicamentos e insumos con las prestaciones otorgadas.

6. Prescripción y Requerimientos: Esta función relaciona el acto profesional de salud dado por la medicina, la odontología, la obstetricia, la enfermería, la nutrición, etcétera, con farmacia.

7. Dispensación y Despacho: Son las actividades orientadas a entregar a las unidades de atención de salud o a los pacientes lo que se requiere, de forma de garantizar la seguridad en la entrega, conservar la calidad de los fármacos e insumos, dar seguimiento en el cumplimiento de la farmacoterapia de pacientes crónicos y brindar la información necesaria para el cumplimiento de la indicación médica.

8. Seguimiento e Información: Es la recolección y registro de datos y la construcción de información relevante para la toma de decisiones en todos los niveles (costo y gasto asociado a consumo; gasto asociado a diagnósticos, costo/gasto asociado a resultados, entre otros).

Descripción General del Problema.

1.2.1 Puesta en Marcha del Nuevo Sistema de Farmacia 2.0.

Algunas de las razones por las que se plantea la necesidad de implantar el nuevo “Sistema de Farmacia 2.0” son, la evidencia de que el sistema heredado de farmacia no cubre todas las actuales necesidades de funcionamiento, ni nivel de detalle de información que se requiere almacenar para las transacciones, por carecer de funcionalidades como: generar cierre anual de transacciones de dispensación a pacientes, cierre anual de movimientos de inventario de productos, cierre anual de movimientos de pedidos a abastecimiento, cierre anual de distribución de insumos y medicamentos a otras farmacias y creación de archivos históricos.

Además la cantidad de transacciones anuales por concepto de dispensación de medicamentos a pacientes, que se registran a través del “Sistema de Farmacia 1.0”, ocasiona que su base de datos haya crecido más allá del límite soportado e impuesto por las características técnicas del formato de la base de datos “MS-Access 97”, que es de 1 Gigabyte de tamaño en el disco duro, para la base de datos completa, teniendo que recurrir cada cierto tiempo a la manipulación directa de los datos, para eliminar tuplas, prescindiendo de la aplicación que los administra, es decir, el “Sistema de Farmacia 1.0”, con el consiguiente riesgo de causar inconsistencias en la información de la tablas.

Para poner en marcha el nuevo “Sistema de Farmacia 2.0” en el “CESFAM Violeta Parra” se requiere proporcionar a dicho sistema, la información de los pacientes y transacciones de años anteriores que se han registrado en la base de datos del antiguo “Sistema de Farmacia 1.0” (sistema de información heredado). Además es necesario, que antes de incorporar dichos datos, estos sean verificados para comprobar su integridad y consistencia con la información de pacientes que es gestionada en el sistema “Plataforma Virtual del CESFAM Violeta Parra” (Sistema SOME), a fin de evitar registros duplicados, incompletos, erróneos o faltantes de pacientes y estandarizar los nombres dados a los distintos programas de salud, tratamientos y diagnósticos.

1.2.2 Estandarización de la Información de Bases de Datos de los Sistemas de Información.

Dado el volumen de datos al que se requiere chequear su integridad y consistencia, previa escritura en la base de datos del “Sistema de Farmacia 2.0”, y debido a que la información se encuentra guardada en distintas tablas y éstas a su vez son guardadas con distintos formatos de sistemas de gestión de bases de datos (“Microsoft Access 97” y “MySQL Server 5.0”), es que se hace necesario implementar una estrategia para ejecutar la lectura de datos automatizada, de cada formato de base de datos, exportar dichos datos a un

formato estándar para procesar la información, de acuerdo a ciertos criterios y restricciones, descartando o modificando valores, para finalmente asignarlos a las tablas y atributos de éstas que correspondan (mapearlos) y así guardarlos en la base de datos del nuevo Sistema de Farmacia 2.0 que estará integrada con la base de datos del Sistema SOME o “Plataforma Virtual del CESFAM Violeta Parra”.

1.2.3 Desarrollar Módulos para la Administración y Mantenimiento.

Tanto el “Sistema de Farmacia 1.0”, como el nuevo “Sistema de Farmacia 2.0” no cuentan con funcionalidades propias para administrar respaldos de sus respectivas bases de datos, ni restauración de las mismas. Esta carencia es un factor de riesgo, ya que, una falla en el sistema o en la máquina que lo soporta, compromete gravemente la continuidad de operación del software y por lo tanto del trabajo de la unidad de farmacia del CESFAM, incluyendo la posible pérdida de información sensible de los pacientes e inventario.

Para evitar consecuencias desastrosas y dependencia de otros software de terceros para las tareas de respaldo y restauración, es necesario dotar al nuevo “Sistema de Farmacia 2.0” de dichas funcionalidades de respaldo y restauración, a fin de que puedan ser utilizadas regularmente por el Usuario Administrador del sistema y así además evitar que se utilice software externo para estas tareas con el consiguiente riesgo de dañar la estructura de la base de datos, su integridad y consistencia entre tablas, debido a su manipulación directa que prescinde peligrosamente de la aplicación que la utiliza, es decir, el “Sistema de Farmacia 2.0”, por cuanto es necesario conocer muy en detalle la estructura y relaciones existentes entre las tablas de la base de datos para poder trabajar directamente con ella sin cometer errores.

Marco Teórico.

Antes de abordar la solución al problema planteado, es necesario realizar una revisión de la problemática subyacente para los sistemas de información heredados, así como también algunas estrategias que han investigado y aplicado exitosamente algunos autores para enfrentar migraciones de sistemas. Esto es necesario porque, si bien el caso en desarrollo trata de una implantación de sistema, el proceso está enmarcado dentro de un proceso de migración desde el “Sistema de Farmacia 1.0” al nuevo “Sistema de Farmacia 2.0” que se pone en marcha. De las diferentes estrategias de migración revisadas se analiza la conveniencia de adoptar algunos de sus elementos.

Se presentan además: la metodología de desarrollo iterativo-incremental; el enfoque de desarrollo orientado a objetos, el cual es aplicado en este proyecto para el desarrollo de la solución para migrar los datos; y la tecnología de aplicaciones Web que se usa en el “Sistema de Farmacia 2.0” y por ende, en la implementación de los módulos de administración de éste.

1.3.1 Sistemas de Información Heredados.

Se define como Sistema de Información Heredado (SIH) o Sistema Legacy, a un sistema informático (equipos informáticos y/o aplicaciones) que ha quedado obsoleto pero continúa siendo utilizado por el usuario (típicamente una organización o empresa) y no se quiere o no se puede reemplazar o actualizar de forma sencilla por ser imprescindible para la gestión [Wikipedia 2009].

“Un Sistema de Información Heredado es cualquier sistema de información que se resiste significativamente a cambios y modificaciones” [BRODI95]. “Además en su mayoría son sistemas que gestionan información y procesos críticos dentro de la organización, lo que significa que una falla en estos sistemas informáticos sería un hecho grave que puede comprometer el desempeño de la organización. Del mismo modo integran el

principal flujo de información en una organización y son el medio que la organización tiene para consolidar esta información acerca del negocio” [Caro *et al.* 2002].

“La problemática subyacente de los SIH está directamente ligada al hecho de que tanto la organización, como la actividad a la que se dedica ésta y sus reglas de negocio, van cambiando con el tiempo, lo que implica entonces que los sistemas de información que posee, también evolucionen necesariamente” [Caro *et al.* 2002].

Caro *et al.* [Caro2002] en su investigación sobre Metodologías de apoyo para Migración de Sistemas Heredados, cita a Bisbal *et al.* [BISBAL99], quién plantea que las soluciones a la problemática de los SIH se encuadran básicamente en 3 categorías: *redesarrollo*, que implica volver a escribir la aplicación existente; *wrapping* (envoltura), que provee una nueva interfaz para el SIH o algún componente de éste, lo que permite mayor accesibilidad desde otras aplicaciones; y *migración*, que mueve al SIH a un nuevo ambiente o plataforma más flexible y actualizada, reteniendo la funcionalidad y los datos del sistema original a la vez que da la posibilidad de incorporar nuevas funcionalidades.

Cada una de estas soluciones tiene un mayor o menor grado de impacto en el SIH y, en consecuencia, en la organización. Así, soluciones parciales tales como *wrapping* son medianamente adoptadas por las organizaciones, debido a que son soluciones a corto plazo que pueden complicar la mantención y administración de los sistemas en el largo plazo; el enfoque de *redesarrollo* tiende a ser demasiado riesgoso y costoso para la mayoría de las organizaciones; y finalmente, la *migración* aparece como una alternativa de menor riesgo y mayor factibilidad de éxito [BISBA99].

El caso del SIH en el departamento de Farmacia del “CESFAM Violeta Parra”, se trata de una aplicación que lleva 9 años en explotación y cuando fue implantada vino a solucionar el problema de ese momento que

era el manejo totalmente manual que se hacía tanto del inventario de farmacia como de las entregas de medicamentos, incluyendo las estadísticas que se diseñaban.

Dicho sistema está programado en Visual Basic versión 6, como aplicación que funciona en una red local o LAN, con una base de datos “Microsoft Access 97” y 5 computadores personales en línea. Cabe señalar además que el SIH es de misión crítica tanto para la farmacia como para el CESFAM dado que en las farmacias se atiende a todos los pacientes para dispensarles los medicamentos que les son prescritos por los profesionales de la salud y en el SIH se registran todas las transacciones a diario.

Por otro lado, a pesar de que el SIH funciona en una red LAN, no comparte información con ningún otro sistema del CESFAM, por lo que es un sistema independiente, lo que no implica que en forma manual se comparta la información generada por este sistema para contrastarla con la de otros y tomar decisiones en distintos niveles del CESFAM.

Respecto de la documentación propia del SIH, sólo fue posible acceder a la documentación del desarrollo del sistema inicial, en el trabajo de título de Hermosilla [HERMOSILLA2000], la que se encuentra desactualizada comparada con el estado actual del SIH que fue modificado al menos en tres oportunidades durante su ciclo de vida.

1.3.2 Estrategias de Migración e Implantación de Sistemas de Información.

Como la migración es una de las alternativas menos riesgosas para afrontar la problemática de los SIH, diferentes investigadores han explorado y presentado diversas formas de migración de las cuales se pueden tomar distintos aspectos según sea posible aplicarlos a un caso particular.

Las estrategias de migración tienen en común algunos elementos como son el hecho de que es necesario conocer a fondo el SIH que es objeto de la migración, lo que se logra a través de la búsqueda de conocimiento en

la documentación del SIH, en los usuarios del mismo y en las personas encargadas de la mantención y actualización del sistema.

Sin embargo muchas veces, es insuficiente el recurrir a las citadas fuentes porque la documentación no existe, es escasa o no está actualizada, los usuarios no tienen conocimiento amplio del sistema y el código fuente no está disponible. En este caso es necesario contar con otras estrategias para extraer el conocimiento requerido.

Caro *et al.* [Caro *et al.* 2002] propone el entendimiento del SIH mediante el análisis de los datos heredados de la base de datos, con herramientas KDD (Knowledge Discovery in Databases) y mediante de esta estrategia reconstruir algunos aspectos básicos del SIH a migrar, de modo que sea posible entender el modelo de datos, el modelo de negocios que se intenta cubrir y determinar el nivel de calidad de los datos del SIH. Estos elementos son esenciales tanto para la migración global del SIH, así como para la migración de la información como datos.

Es aquí donde entra en juego el concepto de “minería de datos”, que es “el proceso de descubrimiento de conocimiento en grandes volúmenes de datos”, pues lo que se pretende es “excavar” entre los datos para hallar información oculta y que posiblemente sea de gran utilidad en la toma de decisiones, [Caro *et al.* 2002] como en el caso de la migración, *conocer el verdadero significado de los datos*. En la práctica, la Minería de Datos forma parte del proceso denominado “proceso KDD”. Este proceso es de utilidad en el caso de la implantación del “Sistema de Farmacia 2.0”, ya que es necesario conocer a fondo el modelo de datos, las relaciones que existen entre las tablas y entre los atributos de estas, a fin de asociar correctamente el modelo de datos heredado con el nuevo y migrar los datos asegurando la consistencia, integridad y calidad de estos.

Existen diferentes puntos de vista respecto de qué técnicas pueden ser consideradas parte de la Minería de Datos. Por ejemplo, para algunos,

cualquier herramienta que ayude a extraer más información a partir de un conjunto de datos es útil [ADRIA96]. Bajo este enfoque entonces, también se deberían considerar como herramientas de Minería de datos a las planillas electrónicas, generadores de informes, lenguajes SQL, generadores de gráficos, herramientas de estadística, etc. Otros en cambio, entienden la Minería de Datos como la aplicación de técnicas de inteligencia artificial, incluyendo entre otras modelamiento avanzado e inducción de reglas [AGRAW93] [FAYYAD96] [Caro2002].

A continuación se presentan algunas metodologías de migración estudiadas por Caro *et al.* [Caro2002] en su investigación:

a) Metodología de Migración Big Bang o Cold Turkey:

Es un método sencillo, pero de alto riesgo que consiste en redesarrollar completamente el SIH, usando una arquitectura, herramientas y bases de datos actualizadas, y trabajando en una nueva plataforma independiente y totalmente aparte del SIH, que es en gran medida lo que se aplicó al migrar el “Sistema Heredado de Farmacia 1.0” al nuevo “Sistema de Farmacia 2.0”, excepto porque se ha conservado lo esencial del modelo de datos. Bajo estas condiciones, los riesgos de este enfoque son muy altos ya que, entre otras razones, a menudo la documentación de estos SIH no existe, está desactualizada como el caso del presente proyecto, o incompleta, y la recuperación de la funcionalidad del SIH, que es fundamental para el éxito del proyecto, se hace difícil. Si a lo anterior agregamos que durante la construcción del nuevo sistema, que puede tomar mucho tiempo, el SIH puede ser adaptado a los nuevos requerimientos que surjan en la organización, al momento de poner en marcha el nuevo sistema y desconectar el SIH se pueden generar problemas debido a que se habrá perdido parte de la funcionalidad del SIH, y los riesgos del fracaso aumentan.

Caro *et al.* [Caro2002] concluye que en la práctica el riesgo de fracasar con esta alternativa es tan alto, que es difícil pensar seriamente en adoptarla [BRODI95], [BISBA99]. Por otra parte no se puede desconocer el hecho que aunque sea una metodología riesgosa, hay muchas ocasiones en que es adoptada, precisamente por su sencillez.

b) Metodología Chicken Little:

Consiste en desarrollar la migración de un SIH en una serie de etapas: analizar y descomponer incrementalmente la estructura del SIH, diseñar incrementalmente el nuevo sistema, crear e instalar los gateways necesarios, migrar incrementalmente la aplicación, interfaces y la base de datos, pasar al nuevo sistema de información. Cada etapa será desarrollada y puesta en funcionamiento en forma incremental. Con este método el SIH opera en paralelo con el nuevo sistema durante la migración. En la medida que la migración avanza el nuevo sistema aumenta de tamaño. Durante la migración cuando un nuevo componente del nuevo sistema esta operativo comienza a funcionar y se conecta al resto de los componentes del SIH mediante gateways, que pueden ser definidos como un módulo de software introducido entre componentes de software en operación, con el objeto de mediar entre ellos. Dichos gateways son los que deberán encargarse de proveer la interoperabilidad necesaria. Una vez que el último componente ha sido terminado y puesto en marcha la migración a terminado. [BRODI95],

Una de las desventajas de esta metodología es que durante todo el proceso de migración se accesan en forma simultánea el SIH y el nuevo sistema, lo que implica un alto costo por mantener la consistencia entre ambos sistemas. En el caso de desarrollo, no se adopta esta metodología por el esfuerzo adicional que significa implementar los llamados “gateways” frente al hecho que el “Sistema Heredado de Farmacia 1.0” es un sistema con un modelo de datos fuertemente cohesionado e interrelacionado y

relativamente pocas interfaces de usuario, que integran las muchas funcionalidades que posee.

c) Metodología de Migración Butterfly: [Wu 1997].

El objetivo de la metodología Butterfly es guiar la migración de SIH, que tiene una misión crítica dentro de la organización. Elimina, durante la migración, la necesidad de los usuarios del sistema de acceder simultáneamente el SIH y el nuevo sistema, y por consiguiente, evita tener que mantener la consistencia entre dichos sistemas de información. Este aspecto de la metodología la hace ideal para utilizarla en el caso de la implantación del “Sistema de Farmacia 2.0”, enmarcada dentro del gran proceso de migración de la aplicación heredada a la nueva.

Esta metodología se basa en la presunción de que los datos de un SIH son lógicamente la parte más importante del sistema y que, desde este punto de vista del desarrollo total del nuevo sistema, no siempre el cambio de los datos heredados es importante, sino más bien su semántica (significado) o modelo para migrar correcta y exitosamente la aplicación.

De acuerdo con esta metodología, la migración de un SIH puede ser dividida en 6 etapas principales, cada una de las cuales está compuesta por una cierta cantidad de actividades de migración, normalmente independientes. En cada etapa algunas de estas actividades son más importantes que otras, y con énfasis en distintos aspectos, como el diseño de la nueva aplicación, la migración de los datos y otras. Las etapas son:

1. Preparación de la migración
2. Comprensión de la semántica del SIH y desarrollo del nuevo modelo de datos.

3. Construcción de un Almacén de Datos de prueba, con muestras de datos del sistema heredado transformados a datos en el formato del sistema nuevo.
4. Migrar incrementalmente todos los componentes del SIH (excepto los datos) a la nueva arquitectura.
5. Migrar gradualmente los datos heredados al sistema nuevo y entrenar a los usuarios en el nuevo sistema.
6. Trasládarse al nuevo sistema terminado.

En cada etapa, algunas de las actividades pueden ser ejecutadas en forma paralela con otras o bien, ser interdependientes. En general, estas 6 etapas deben ser conducidas en una secuencia basada en los principios naturales del desarrollo de sistemas. Sin embargo, de acuerdo con sus creadores, normalmente cada migración de SIH es única y a menudo también el flujo generado es único. Esta metodología permitiría esta flexibilidad.

En la etapa 5 de esta *Metodología Butterfly*, el objetivo es principalmente la migración de los datos heredados (legacy data) y es la parte central de la metodología. Los datos heredados serán migrados gradualmente mediante un procedimiento que contempla el uso de una serie de almacenes temporales, un módulo de software para mapeo y traspaso de los datos (Data-Access-Allocator DAA) y el transformador de datos (Chrysaliser).

Cuando la migración de datos comienza, los datos heredados almacenados se accesan sólo para lectura. Las manipulaciones de estos datos son redirigidas por el DAA y los resultados almacenados en una serie de archivos auxiliares llamados almacenes temporales (Temp Stores TS).

La metodología Butterfly emplea un Data-Transformer (Transformador de Datos), llamado "Chrysaliser", para migrar gradualmente, los datos del sistema heredado, hacia almacenes temporales (TempStores o TS) en el sistema objetivo. Chrysaliser es responsable de transformar los datos desde el formato heredado al formato del modelo de datos objetivo. Esta transformación toma la forma de un conjunto de reglas que son aplicadas a los datos. Las reglas derivan en un proceso de determinación del esquema objetivo que es conceptualmente equivalente al modelo de datos heredado. La metodología de migración "Butterfly" se muestra en la figura a continuación:

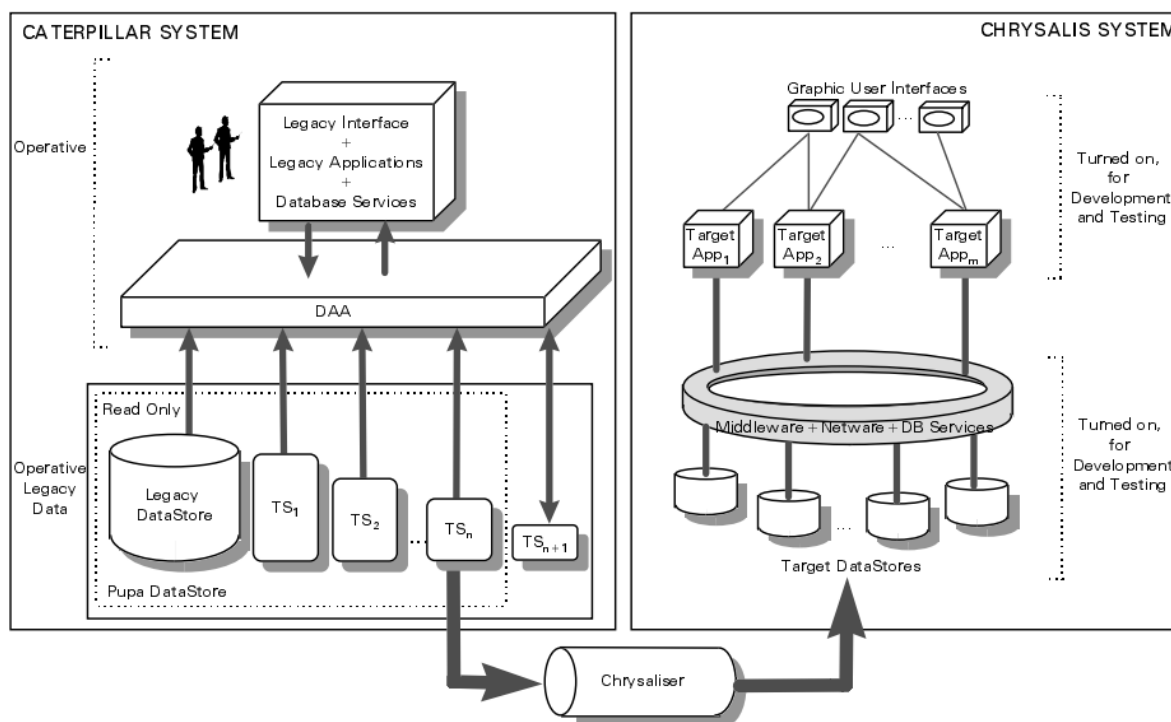


Fig. 2: Estrategia de Migración Butterfly: Migrando datos en almacén temporal TS_n. [Wu 1997]

En el caso de la migración del "Sistema Heredado de Farmacia 1.0", se han tomado diversos elementos de las distintas metodologías, los cuales se han aplicado tanto en la decisión del desarrollo de la aplicación, previo al presente proyecto y en el desarrollo de la aplicación de migración.

1.3.3 Metodología de Desarrollo Iterativo-Incremental.

La metodología empleada para construir los módulos de administración es la metodología iterativo-incremental, la cual combina aspectos del método clásico lineal secuencial, con aspectos de la metodología de construcción de prototipos, lo que permite ir evaluando a medida que la aplicación se va completando [Pressman 2002].

Esta metodología de desarrollo consta de ciclos sucesivos en cada uno de los cuales el sistema en desarrollo va incrementando su funcionalidad hasta lograr completar todos los requerimientos. En un primer momento el sistema ofrece una funcionalidad básica, quedando pendientes varios requerimientos secundarios, que no son tan evidentes. Luego se evalúa el funcionamiento del incremento y en base a dicha evaluación se desarrolla el siguiente incremento más completo. El proceso se repite hasta que el sistema está terminado y en condiciones de operar al 100% de sus funcionalidades.

1.3.4 Enfoque de Desarrollo Orientado a Objetos.

La Programación Orientada a Objetos (POO u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de 1990. Actualmente son muchos los lenguajes de programación que soportan la orientación a objetos.

Algunos conceptos básicos [MVARAS2008]:

a) Objeto:

Las personas tienen una idea clara de lo que es un objeto: conceptos adquiridos que nos permiten sentir y razonar acerca de las cosas del mundo. Un objeto podría ser real o abstracto, por ejemplo una organización, una

factura, una figura en un dibujador, una pantalla de usuario, un avión, un vuelo de avión, etc.

En el análisis y diseño orientados a objetos (OO), interesa el comportamiento del objeto. Si se construye software, los módulos de software OO se basan en los tipos de objetos. El software que implanta el objeto contiene estructuras de datos y operaciones que expresan dicho comportamiento. Las operaciones se codifican como métodos. La representación en software OO del objeto es entonces una colección de tipos de datos y objetos.

Entonces, dentro del software orientado a objeto, un objeto es cualquier cosa, real o abstracta, acerca de la cual almacenamos datos y los métodos que controlan dichos datos.

Un objeto puede estar compuesto por otros objetos. Estos últimos a su vez también pueden estar compuestos por otros objetos. Esta intrincada estructura es la que permite construir objetos muy complejos.

b) Tipo de Objeto:

Los conceptos que poseemos se aplican a tipos determinados de objetos. Por ejemplo, paciente se aplica a los objetos que son personas atendidas por un profesional. Algunas instancias de paciente podrían ser Juan Pérez, José Martínez, etc. En el análisis orientado a objetos, estos conceptos se llaman tipos de objetos; las instancias se llaman objetos.

Así, un tipo de objeto es una categoría de objeto, mientras que un objeto es una instancia de un tipo de objeto.

En el mundo de las bases de datos existen los tipos de entidad, como cliente o empleado. Existen muchas instancias de cada tipo de entidad (como Juan Pérez o José Martínez para el tipo de entidad empleado). Del mismo modo, en OO se define tipos de objetos e instancias de tipo de objeto.

Sin embargo, el término objeto tiene diferencias fundamentales con el término entidad, ya que la entidad sólo se refiere a los datos, mientras que objeto se refiere a los datos y a los métodos mediante los cuales se controlan a los propios datos. En OO, la estructura de datos y los métodos de cada tipo de objeto se manejan juntos. No se puede tener acceso o control de la estructura de datos excepto mediante los métodos que forman parte del tipo de objeto.

c) Métodos:

Los métodos especifican la forma en que se controlan los datos de un objeto. Los métodos en un tipo de objeto sólo hacen referencia a la estructura de datos de ese tipo de objeto. No deben tener acceso directo a las estructuras de datos de otros objetos. Para utilizar la estructura de datos de otro objeto, deben enviar un mensaje a éste. El tipo de objeto empaca juntos los tipos de datos y su comportamiento.

Un objeto entonces es una cosa cuyas propiedades están representadas por tipos de datos y su comportamiento por métodos.

Un método asociado con el tipo de objeto factura podría ser aquel que calcule el total de la factura. Otro podría transmitir la factura a un cliente. Otro podría verificar de manera periódica si la factura ha sido pagada y, en caso contrario, añadir cierta tasa de interés.

d) Encapsulado:

El empaque conjunto de datos y métodos se llama encapsulado. El objeto esconde sus datos de los demás objetos y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultamiento de información. El encapsulamiento evita la corrupción de los datos de un objeto. Si todos los programas pudieran tener acceso a los datos de cualquier forma que quisieran los usuarios, los datos se podrían corromper o utilizar

de mala manera. El encapsulado protege los datos del uso arbitrario y no pretendido.

El encapsulado oculta los detalles de su implantación interna a los usuarios de un objeto. Los usuarios se dan cuenta de las operaciones que puede solicitar del objeto, pero desconocen los detalles de cómo se lleva a cabo la operación. Todos los detalles específicos de los datos del objeto y la codificación de sus operaciones están fuera del alcance del usuario.

Así, encapsulado es el resultado (o acto) de ocultar los detalles de implantación de un objeto respecto de su usuario.

El encapsulado, al separar el comportamiento del objeto de su implantación, permite la modificación de ésta sin que se tengan que modificar las aplicaciones que lo utilizan.

e) Mensajes:

Para que un objeto haga algo, le enviamos una solicitud. Esta hace que se produzca una operación. La operación ejecuta el método apropiado y, de manera opcional, produce una respuesta. El mensaje que constituye la solicitud contiene el nombre del objeto, el nombre de una operación y, a veces, un grupo de parámetros.

La programación orientada a objetos es una forma de diseño modular en la que con frecuencia el mundo se piensa en términos de objetos, operaciones, métodos y mensajes que se transfieren entre tales objetos. Un mensaje es una solicitud para que se lleve a cabo la operación indicada y se produzca el resultado.

Los objetos pueden ser muy complejos, puesto que pueden contener muchos subobjetos, éstos a su vez pueden contener otros, etc. La persona que utilice el objeto no tiene que conocer su complejidad interna, sino la forma de comunicarse con él y la forma en que responde.

f) Clase:

El término clase se refiere a la implantación en software de un tipo de objeto.

El tipo de objeto es una noción de concepto. Especifica una familia de objetos sin estipular la forma en que se implanten. Los tipos de objetos se especifican durante el análisis OO.

Así, una clase es una implantación de un tipo de objeto. Especifica una estructura de datos y los métodos operativos permisibles que se aplican a cada uno de sus objetos.

g) Herencia:

Un tipo de objeto de alto nivel puede especializarse en tipos de objeto de bajo nivel. Un tipo de objeto puede tener subtipos. Por ejemplo, el tipo de objeto persona puede tener subtipos estudiante y empleado. A su vez, el tipo de objeto estudiante puede tener como subtipo estudiante de pregrado y estudiante de postgrado, mientras que empleado puede tener como subtipo académico y administrativo. Existe de este modo una jerarquía de tipos, subtipos, etc.

Una clase implanta el tipo de objeto. Una subclase hereda propiedades de su clase padre; una sub-subclase hereda propiedades de las subclases; etc. Una subclase puede heredar la estructura de datos y los métodos, o algunos de los métodos, de su superclase. También tiene sus métodos e incluso tipos de datos propios.

1.3.5 Aplicaciones Web.

En la ingeniería de software se denomina aplicación web o webapp a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un computador servidor web a través de Internet o de una red de área local (intranet), mediante un programa navegador o browser. En otras palabras, es

una aplicación software que se codifica en un lenguaje de programación (HTML, Javascript, Java, etc.), soportado por los navegadores web (Internet Explorer, Mozilla, Opera, etc.) que utilizan los usuarios.

Las aplicaciones web generan dinámicamente (por el lado del servidor) una serie de páginas web que son mostradas en el navegador del usuario (lado del cliente). Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva en la pantalla del computador.

Una ventaja significativa es que las aplicaciones web funcionan independientemente del sistema operativo instalado en el lado del cliente. En lugar de tener que crear versiones de la webapp para distintos sistemas operativos (Windows, Mac OS X, GNU/Linux), esta se escribe una vez y se ejecuta en todos los computadores que tengan un navegador web instalado y acceso a través de la red. Otra ventaja de las webapps es que por residir computadores servidores en la red, pueden ser accesadas por múltiples usuarios, sin tener que instalarlas en cada computador cliente, como es el caso del nuevo “Sistema de Farmacia 2.0”.

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres capas. Desde el punto de vista del computador cliente, la primera capa está dada por el navegador web que envía solicitudes a una segunda capa que está en el computador servidor (llamado así porque atiende las solicitudes), donde se ejecuta la lógica de la webapp y desde aquí se envían consultas o actualizaciones a la tercera capa constituida por la base de datos, que puede residir en el mismo computador servidor o en otro conectado a la red.

Herramientas de Diseño.

1.4.1 Lenguaje Unificado de Modelado UML.

El Lenguaje Unificado de Modelado (UML, Unified Modeling Language) es un lenguaje de modelado visual de propósito general que se utiliza para especificar, visualizar, construir y documentar los artefactos de un sistema software. Captura decisiones y conocimiento sobre sistemas que deben ser construidos. Se usa para comprender, diseñar, configurar, mantener y controlar la información sobre tales sistemas. Está pensado para ser utilizado con todos los métodos de desarrollo, etapas del ciclo de vida del software, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas de software actuales en una aproximación estándar. UML incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser apoyado por herramientas de modelado visuales e interactivas que dispongan de generadores, tanto de código, como de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos existentes.

UML fue desarrollado en un esfuerzo para simplificar y consolidar el gran número de métodos de desarrollo orientados a objetos que han surgido [Rumbaugh *et al.* 2007].

1.4.2 El Framework Modelo Vista Controlador MVC.

MVC o Model View Controller es un patrón de diseño aportado originariamente por el lenguaje SmallTalk a la Ingeniería del Software. El paradigma MVC consiste en dividir las aplicaciones en tres partes: *Modelo*, *Vista* y *Controlador* [González, R. 2006].

El *Controlador* es el encargado de redirigir o asignar una aplicación (un modelo) a cada petición; el controlador debe poseer de algún modo, un "mapa" de correspondencias entre peticiones y respuestas (aplicaciones o modelo) que se les asignan.

El *Modelo* es la aplicación que responde a una petición, es la lógica de negocio a fin de cuentas.

La *Vista* en particular es la que recibe el resultado de las operaciones necesarias realizadas provenientes del modelo.

En MVC el procesamiento se lleva a cabo entre sus tres componentes. El controlador recibe una orden y decide quién la lleva a cabo en el modelo. Una vez que el modelo (la lógica de negocio) termina sus operaciones devuelve el flujo, vuelve al controlador y este envía el resultado a la capa de presentación. Esto queda clarificado en la Figura Fig. 3).

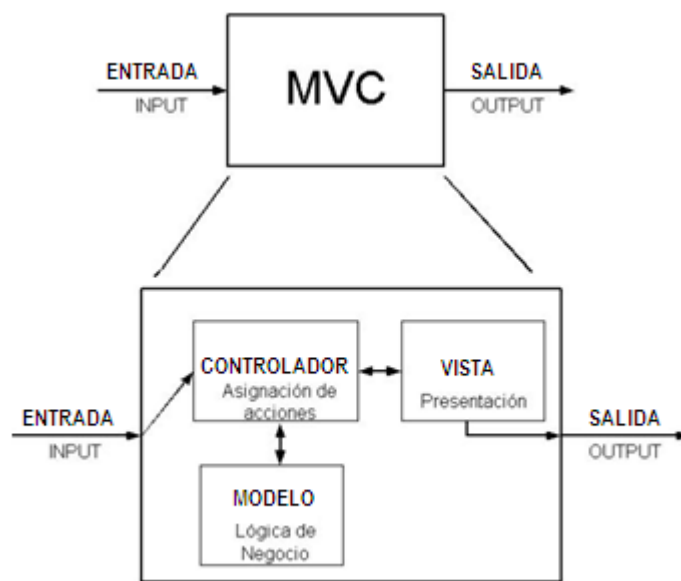


Fig. 3: Esquema de funcionamiento del patrón MVC.

1.4.3 Diseño de Bases de Datos.

Una base de datos se define como “una colección de datos lógicamente relacionados” [Elmasri & Navathe 1997]. La creación de una base de datos es, generalmente, una operación difícil, larga y costosa, que no puede improvisarse, pues involucra la representación del problema a resolver y el modelamiento de la realidad; su diseño es una tarea que está comprendida dentro de un proceso más amplio, que es la metodología de desarrollo de bases de datos basada en tres etapas fundamentales [Calero *et al.* 2006]: *Diseño conceptual*, *Diseño lógico* y *Diseño físico*. Estas etapas son abordadas utilizando los modelos Entidad Relación y Relacional.

El *diseño conceptual*, es la primera fase del desarrollo de una base de datos y se subdivide en dos etapas: etapa de *análisis de requisitos* y etapa de *generación del esquema conceptual*.

La etapa de *análisis de requisitos*, en general común para datos y procesos, es de percepción, identificación y descripción de los fenómenos del mundo real a analizar.

En el análisis de requisitos se responde a la pregunta “¿qué representar?”. Mediante el estudio de las reglas y procedimientos que rigen el trabajo en la organización (que proveen el marco para el análisis del sistema) y de entrevistas a los usuarios de los diferentes niveles de la organización (que proveen los detalles sobre los datos) se llega a elaborar un esquema descriptivo de la realidad. El esquema descriptivo se representa utilizando el lenguaje natural, y con ello se ayuda a que el problema de comunicación entre usuarios y analistas se reduzca.

En la etapa de *generación del esquema* se transforma el esquema descriptivo, refinándolo y estructurándolo adecuadamente. Esta etapa responde a la pregunta: “¿cómo representar?”.

En esta etapa de conceptualización se busca una representación normalizada, que se apoye en un modelo de datos que cumpla determinadas propiedades (claridad, sencillez, coherencia, plenitud, no redundancia, fidelidad, etc.), para llegar así al denominado esquema conceptual.

Para la representación del esquema conceptual, se usa el Modelo Entidad Relación (Modelo E/R).

Al generar el esquema conceptual es necesario interpretar las frases del lenguaje natural en el que está descrito el esquema percibido en el análisis, convirtiéndolas en elementos del Modelo E/R.

El Modelo E/R utiliza fundamentalmente dos elementos, para describir las asociaciones que existen entre las categorías de datos de un sistema de información: *Entidad* y *Relación*. [Soto & Rodríguez 2000].

Entidad: “Cualquier ente, real o abstracto (tangible o intangible), distinguible en la organización, sobre el cual se desea almacenar datos”. Se representa como un rectángulo dentro del cual se muestra una palabra que es el nombre de la entidad. Cada rectángulo representa todas las ocurrencias o presencias de la entidad respectiva. Por ejemplo "Paciente" representa a todos los pacientes del CESFAM.

Una entidad se describe a través de características llamadas "atributos" (*Fig. 4*), que van tomando distintos valores posibles para cada ocurrencia de la entidad. Debe existir al menos un atributo de datos que tome un valor único para cada ocurrencia, es decir, no pueden existir dos ocurrencias con igual valor para ese atributo. Dicho atributo se denomina clave o identificador.

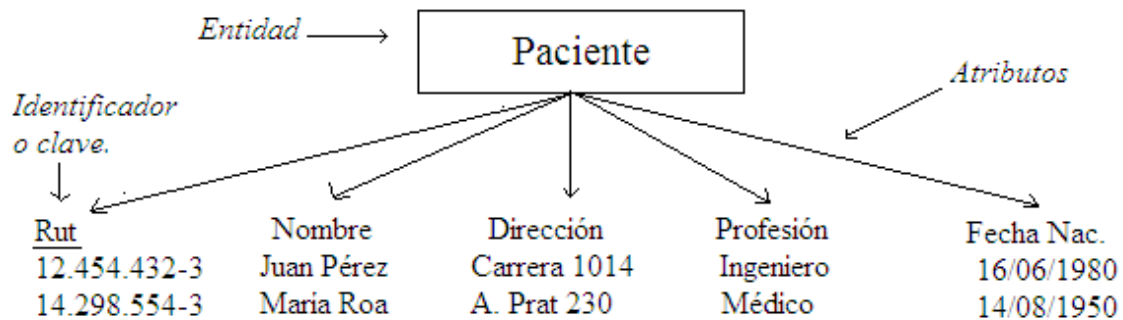


Fig. 4: Conceptos de Entidad, Atributo y Clave en el modelo E/R.

Relación: Es la asociación natural que existe entre una o más entidades”. También se define como las actividades o sucesos que unen a dos o más entidades entre sí”. Se representa como rombo que conecta las entidades que relaciona. Se debe nominar mediante un verbo o una frase verbal, por ejemplo: un "Paciente" retira una "Receta" (ver Fig. 5).

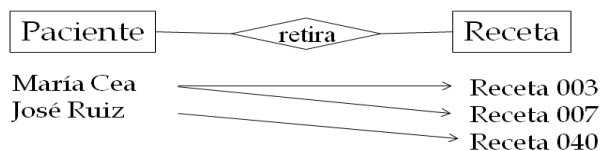


Fig. 5: Relación entre Entidades en el Modelo E/R.

Dos entidades pueden tener una o más relaciones entre sí como se aprecia en la Fig. 6.

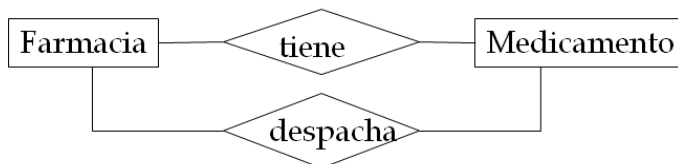
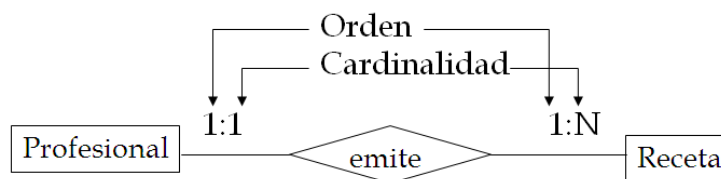


Fig. 6: Entidades con múltiples relaciones entre sí en el Modelo E/R.

Por otra parte, existen dos reglas que definen una relación: *orden* y *cardinalidad* (ver Fig. 7).

Orden: Define si la relación entre las entidades es obligatoria u opcional, es decir, el número mínimo de ocurrencias de una entidad con respecto a la otra.

Cardinalidad: Define el número máximo de ocurrencias de una entidad para una única presencia de la entidad relacionada.



Se dice: Un profesional emite una o más recetas.
Una receta es emitida por un solo profesional.

Fig. 7: Orden y Cardinalidad en el Modelo E/R.

1.4.4 Patrones de Diseño Abstract Factory, Data Access Object y Data Transfer Object.

En el diseño de software existen soluciones probadas a problemas que ocurren repetidamente, cuando se trata de estructurar la interacción entre los componentes del sistema y delegar responsabilidades específicas a determinadas partes del sistema en construcción, para abstraerse de detalles que deben analizarse e implementarse en distintos niveles para lograr una arquitectura robusta y sistemas que sean fáciles de entender y mantener.

Un patrón describe los elementos que forman el diseño, sus relaciones, responsabilidades y colaboraciones. No describe un diseño o implementación concreta particular, debido a que es una solución general que puede ser aplicada en muchas situaciones diferentes.

A continuación se presenta una breve explicación de los patrones Data Access Object (DAO), Abstract Factory y Transfer Object, que serán útiles en el diseño de las soluciones que se plantean en el proyecto.

a) **Patrón Data Access Object (DAO):**

El problema a solucionar consiste en que una aplicación puede estar estructurada de tal manera que la lógica del negocio y la lógica de persistencia estén entremezcladas. Si por alguna razón cuando la aplicación está en funcionamiento en una organización, se requiere cambiar el motor de base de datos que subyace, habrá que reemplazar todas las partes de la aplicación que son llamadas específicas a ese motor de base de datos, con el consiguiente problema para reestructurar la aplicación.

Al utilizar el patrón DAO, se crea una capa adicional para entregar los servicios relacionados con la persistencia a los demás componentes del sistema, separando esta capa de la lógica del negocio, con lo cual si es necesario cambiar el motor de la base de datos o acceder dos o más distintos desde la aplicación, sólo bastará con crear los nuevos componentes, que contengan los métodos y llamadas específicas para leer, escribir, modificar o eliminar datos para el motor de base de datos requerido, todo desde la capa de persistencia y no en la capa lógica.

Este patrón DAO surge históricamente de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además ocultar la forma de acceder a los datos. Se trata de que el software cliente se centre en los datos que necesita y no en cómo se realiza el acceso a los datos o de cuál es la fuente de almacenamiento.

b) Patrón Abstract Factory:

Implementa una fábrica abstracta de objetos. El problema que resuelve consiste en que se tienen diferentes tipos de objetos concretos relacionados entre sí, que deben ser creados con una lógica compleja la cual quiere separarse y encapsularse, además dichos objetos poseen operaciones comunes como podrían ser *buscar*, *leer*, *escribir*, *eliminar* pero que se implementan en forma distinta por cada tipo de objeto. Frente a este problema el patrón factory implementa un objeto abstracto que es capaz de crear los distintos tipos de objetos y de esta forma en el sistema sólo se hacen las peticiones de creación a un solo tipo de objeto que decide cual crear de acuerdo a parámetros que se le entregan. El patrón factory también se implementa en combinación con el patrón “Singleton”, que sirve para crear instancias únicas de objetos en un momento dado, de manera que si se trata de crear un objeto del mismo tipo mientras la instancia única existe se retornará la misma instancia única. Es útil cuando se realizan operaciones en que se requiere acceso exclusivo a un recurso que es utilizado en forma concurrente por varios procesos, donde se necesita mantener la coherencia producto de las operaciones, por ejemplo en una base de datos.

c) Patrón Data Transfer Object:

El patrón Data Transfer Object (DTO) responde a la necesidad de transportar los datos entre las diversas capas de la aplicación, para evitar que sean accedidos de cualquier forma y alterados, por ejemplo cuando los datos van desde la base de datos hasta la interfaz del usuario, o si se trata de aplicaciones en ambiente distribuido donde los datos deben viajar de una máquina física a otra. La forma como soluciona el problema este patrón consiste en crear un objeto en el cual se encapsulan los datos que sólo pueden ser accedidos a través de los métodos que el objeto de transferencia provee.

Herramientas de Desarrollo.

1.5.1 Lenguaje de Programación Java.

Java es una plataforma virtual de desarrollo de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales o plataformas. Esta plataforma cuenta con los siguientes componentes: [González, R. 2006]

- El lenguaje de programación Java.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
- El API Java, una biblioteca estándar para el lenguaje.

El lenguaje mismo se inspira en la sintaxis de C++, pero su funcionamiento es más similar al de Smalltalk. Incorpora sincronización y manejo de tareas en el lenguaje mismo e incorpora interfaces como un mecanismo alternativo a la herencia múltiple de C++.

Java fue seleccionado para el desarrollo del proyecto ya que es un lenguaje orientado a objetos por excelencia y el más popular para tal efecto en nuestros días. Este lenguaje se diseñó con las siguientes características:

- Simple: Elimina la complejidad de los lenguajes como C, mantiene las facilidades básicas del lenguaje y proporciona un gran número de extras con las librerías de clases.
- Familiar: La sintaxis de Java es muy similar al lenguaje C++, además se han eliminado ciertas características, como los punteros.
- Robusto: Java maneja la memoria del computador por lo que no es necesario preocuparse por apuntadores, memoria que no se esté utilizando, etc.

- Seguro: Java tiene ciertas políticas que evitan que se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no se puede hacer con los recursos críticos de un computador.
- Portable: Como el código compilado de Java –conocido como byte code- es interpretado, un programa compilado de Java puede ser utilizado por cualquier computador que tenga implementado el interprete de Java.
- Multithreaded: El lenguaje soporta la concurrencia a través de hilos, es decir, puede ejecutar varios procedimientos en paralelo.
- Interpretado y compilado a la vez: Java es compilado, en la medida que su código fuente se transforma en una especie de código de máquina o byte-codes los cuales son semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los byte-codes se pueden ejecutar directamente sobre cualquier máquina a la cual se haya portado el intérprete.
- Dinámico: Java no requiere compilar todas las clases de un programa para que funcione, si se realiza una modificación a una clase, Java se encarga de realizar un Dynamic Binding o un Dinamic Loading para encontrar las clases.

1.5.2 Servlets.

Los Servlets son la respuesta de la tecnología Java a la programación CGI. Son programas que se ejecutan en un servidor Web y construyen páginas Web. Construir páginas Web al vuelo es útil (y comúnmente usado) por un número de razones:

1) La página Web está basada en datos enviados por el usuario. Por ejemplo, las páginas de resultados de los motores de búsqueda se generan de esta forma, y los programas que procesan pedidos desde sitios de comercio electrónico también.

2) Los datos cambian frecuentemente. Por ejemplo, un informe sobre el tiempo o páginas de cabeceras de noticias podrían construir la página dinámicamente, quizás devolviendo una página previamente construida y luego actualizándola.

3) Las páginas Web que usan información desde bases de datos corporativas u otras fuentes. Por ejemplo, usaríamos esto para hacer una página Web en una tienda on-line que liste los precios actuales y el número de artículos en stock.

1.5.3 Java Server Pages JSP.

Java Server Pages (JSP) es una tecnología que permite mezclar código en lenguaje HTML (Hiptertext Markup Language) estático, con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas CGI son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los servlets, hacen que se genere la página completa mediante un programa propio, incluso aunque la mayoría de ella sea siempre lo mismo. JSP permite crear dos partes de forma separada. [HALL2006]

Ventajas de JSP.

- Respecto de Active Server Pages (ASP): ASP es una tecnología similar de Microsoft. Las ventajas de JSP están duplicadas. Primero, la parte dinámica está escrita en Java, no en Visual Basic, otro lenguaje específico de Microsoft, por eso es mucho más poderosa y fácil de usar. Segundo, es portable a otros sistemas operativos y servidores Web.
- Respecto de los Servlets: JSP no proporciona nada que no se pudiese realizar en principio con un servlet. Pero es mucho más conveniente escribir y modificar HTML normal que tener que realizar muchas sentencias “println()” que generen HTML. Además, separando el formato del contenido se puede poner diferentes personas en diferentes

tareas: los expertos en diseño de páginas Web pueden construir el HTML, dejando espacio para que los programadores de servlets inserten el contenido dinámico.

- Respecto de Server-Side Includes (SSI): SSI es una tecnología ampliamente soportada que incluye piezas definidas externamente dentro de una página Web estática. JSP es mejor porque permite usar servlets en vez de un programa separado para generar las partes dinámicas. Además, SSI, realmente está diseñado para inclusiones sencillas, no para programas "reales" que usen formularios de datos o hagan conexiones a bases de datos, etc.
- Respecto de JavaScript: JavaScript puede generar HTML dinámicamente en el navegador cliente. Esta es una capacidad útil, pero sólo maneja situaciones donde la información dinámica está basada en el entorno del cliente. Con la excepción de las cookies, el HTTP y el envío de formularios no están disponibles con JavaScript. Como se ejecuta en el cliente, JavaScript no puede acceder directamente a los recursos en el lado del servidor, como bases de datos, catálogos, información de precios, etc.

1.5.4 Framework Struts.

Struts es un framework para aplicaciones Web java que implementa el modelo MVC. Realmente lo que provee es un conjunto de clases y librerías que conforman el Controlador, permitiendo la integración con el modelo o lógica de negocio y facilitando la construcción de vistas. Naturalmente, el modelo o lógica de negocio es la parte que corresponde desarrollar. Por eso Struts es una plataforma sobre la que se puede montar la lógica de negocio, y además esta plataforma permite dividir la lógica de la presentación entre otras cosas.

Los diferentes componentes del patrón son implementados en struts como se describen a continuación [Valdés, C. y Vásquez, H. 2004]:

Modelo: Struts no provee ningún componente en particular para la capa de negocios.

Vista: Comprende las Java Server Pages (principalmente) y los servlets involucrados en la generación de la interfaz de usuario o con otros Sistemas. Struts provee además de soporte para construir aplicaciones multi-idioma, interacción con formularios y otras utilidades mediante la utilización de Tag Libs especiales.

Controlador: El componente Controlador es el soporte principal de toda aplicación Web basada en Struts. Este es implementado usando un servlet llamado “org.apache.struts.action.ActionServlet”. Este recibe todas las peticiones de los clientes, y delega el control a una clase “org.apache.struts.action.Action” definida por el usuario.

En la figura *Fig. 8* se puede apreciar la forma en que Struts implementa el patrón MVC:

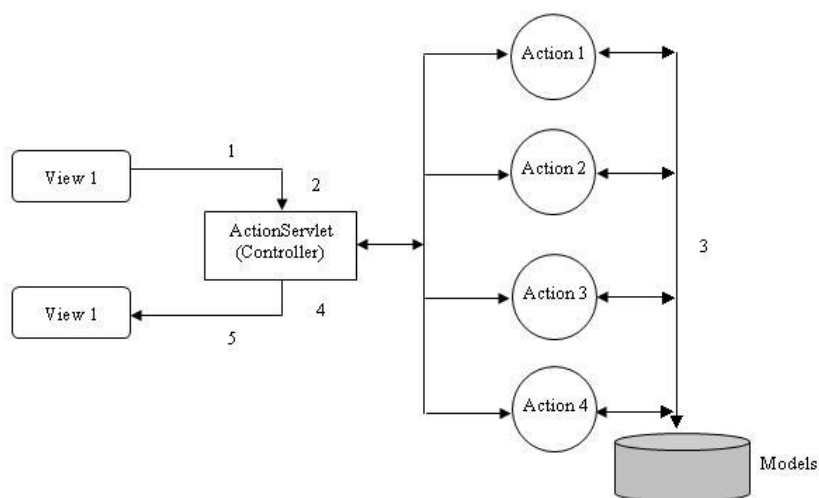


Fig. 8: Implementación del modelo MVC por Struts [GOODWILL, J. 2002].

1.5.5 Middleware Hibernate.

Hibernate es un servicio de consulta y persistencia, para bases de datos relacionales y orientadas a objeto, de alto rendimiento que se instala y actúa como mediador entre la aplicación y la base de datos.

Permite desarrollar clases persistentes usando lenguaje orientado a objetos, incluyendo encapsulamiento, polimorfismo, composición y colecciones. Hibernate permite además expresar consultas en su propia extensión del lenguaje SQL llamada HQL, o bien en SQL nativo, o con un criterio orientado a objetos.

Hibernate implementa la técnica de realizar la transición desde una representación de los datos en el modelo relacional a un modelo orientado a objetos y viceversa, lo que se conoce como ORM (Object Relational Mapping). Mediante un archivo de configuración (hibernate.cfg.xml) se configuran los datos de conexión a la base de datos, un archivo de mapeo (*.hbm.xml) describe los atributos de la base de datos. Una clase Service, que actúa basándose en la información de esos archivos, es la intermediaria entre la aplicación y la base de datos.

1.5.6 Servidor de Aplicaciones Apache Tomcat.

Apache Tomcat es un software servidor de aplicaciones para ambiente Web, que crea un contenedor o ambiente para ejecutar aplicaciones estructuradas en el modelo cliente-servidor sobre redes de área local (LAN), de área extendida (WAN), o Internet. Las aplicaciones residen en una máquina servidor, y funcionan en base a peticiones enviadas desde un programa cliente (navegador web), las cuales son procesadas por el motor de apache Tomcat, para generar el código a ejecutar y luego de ejecutarlo enviar la respuesta de vuelta al programa cliente vía protocolos de red (HTTP, HTTPS).

Específicamente, Apache Tomcat puede procesar aplicaciones que contienen Servlets y páginas JSP. En su versión 6.0 el servidor de aplicaciones implementa las especificaciones de Servlets 2.5 mediante el motor denominado “Catalina” y JSP 2.1 mediante el motor denominado “Jasper”. Por este motivo Apache Tomcat puede ser usado como plataforma para desarrollar y explotar aplicaciones web y servicios web.

Posee características como soporte de conexiones a distintos motores de base de datos como MySQL, Oracle, PostgreSQL a través de drivers configurables para su uso. También se puede configurar para que actúe como servidor “proxy” de aplicaciones, lo que significa que puede recibir todas las peticiones de los clientes en una sola dirección web y encargarse de distribuirlas a las aplicaciones específicas.

La característica de “clustering” consiste en que, configurado debidamente puede operar instalado en varias máquinas físicamente distintas y unidas por red, actuando como si se tratase de un sólo servidor, recibiendo peticiones desde clientes y distribuyéndolas coordinadamente entre las distintas máquinas (nodos) del cluster de forma balanceada. Esta forma de instalación permite soporte de redundancia que protege ante caídas de Apache Tomcat, puesto que cuando un nodo cae, inmediatamente las peticiones son transferidas a otro nodo del cluster que esté en condiciones de responder, también esta forma de responder lo hace más robusto ante periodos de sobrecarga de peticiones.

Posee características de “hosting virtual” al ser configurable para actuar como servidor de sitios web, es decir, las webapps pueden ser invocadas desde los clientes bajo *distintos nombres de servidor* siendo una única máquina y una sola instancia de Apache Tomcat instalado, simplemente configurando el parámetro “<engine/>” en el archivo “server.xml” donde se indican los nombres de host. Las múltiples webapps,

residen físicamente en distintos directorios dentro del directorio principal de Apache Tomcat, definido por la variable de entorno \$CATALINA_HOME.

1.5.7 Sistemas Gestores de Bases de Datos (SGBD) MS-Access 97 y MySQL 5.0.

1) SGBD Microsoft Access 97:

Access 97 es un programa comercial que forma parte de la suite de programas Office 97, de la empresa Microsoft Corporation. Se trata de un sistema gestor de bases de datos diseñado para ser utilizado en computadores personales. Aunque no está considerado como un sistema para alto desempeño con grandes volúmenes de datos, lo cierto es que permite realizar las mismas tareas que los servidores de bases de datos tradicionalmente considerados robustos, como puede ser Oracle o MySQL. A diferencia de ellos, Access puede funcionar en sistemas de bajo costo, y es mucho más asequible desde este punto de vista que Oracle. Además, dispone de un entorno gráfico amigable (al estilo Microsoft Windows). La instalación completa de Access proporciona una serie de herramientas de apoyo al usuario y al programador que pueden facilitar la creación y administración de una base de datos.

Como ya se dijo, Access 97 no es para gestionar bases de datos de gran tamaño, de hecho la barrera en tamaño para un archivo de Access 97 es 1 Gigabyte en disco. Una vez alcanzado el límite el SGBD comienza a rechazar las operaciones de inserción y a lanzar advertencias al usuario.

El archivo de una base de datos Access 97 tiene extensión "MDB" (Microsoft Data Base). Además de este archivo, puede existir otro con extensión "LDB". Este archivo "LDB" contiene información de bloqueo de registros, que permite el acceso seguro de múltiples usuarios de forma concurrente. El archivo de la base de datos permanece abierto mientras se trabaja con los datos, y únicamente se cierra cuando se cierra la base de

datos de forma explícita, con la opción correspondiente. Si por cualquier problema, no se realiza un cierre ordenado, la base de datos puede dañarse.

2) SGBD MySQL 5.0:

Es un sistema gestor de bases de datos relacionales (SGBDR o RDBMS), creado por la empresa MySQL AB, fundada por Michael Widenius. En sus comienzos (1996) MySQL era considerado por muchos como sólo orientado para aplicaciones y sitios web que no fueran de misión crítica para la administración de datos, por ser un proyecto de código abierto, es decir, al cual cualquier persona con los conocimientos necesarios podía aportar ideas y desarrollarlas para mejorar el producto. Con el tiempo ha ido evolucionando y creciendo en funcionalidad para diversos entornos de aplicaciones, conservando la velocidad de respuesta rápida que tiene desde el comienzo.

MySQL compite con sistemas SGBDR propietarios (comerciales) conocidos, como son Oracle, SQL Server y DB2. MySQL se ve potenciado en el mercado de las bases de datos, también por el hecho de que "Apache", con el cual trabaja óptimamente, es el producto de código abierto dominante en el mercado de servidores de aplicaciones Web y por la presencia en el mercado, de varios sistemas operativos de código abierto (como distribuciones de Linux y FreeBSD) que ganan terreno cada día.

MySQL 5.0 posee lenguaje de consulta estructurado (SQL), a través del cual permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos. También posee un sistema de administración de permisos flexible y seguro, además de incorporar dos tipos distintos de motores de almacenamiento, InnoDB y MyIsam. Algunas de las ventajas de MySQL son las siguientes: [Gilfillan2004]

- 1) Costo: El costo de MySQL es gratuito para la mayor parte de los usos, en su versión community.
- 2) Accesibilidad: El software se puede descargar directamente desde su sitio web oficial <http://www.mysql.com>.
- 3) Asistencia: La empresa creadora "MySQL AB", ahora parte de "Sun Microsystems", ofrece contratos de asistencia a precios razonables de mercado y también existe una diversa y activa comunidad de desarrolladores MySQL.
- 4) Velocidad: MySQL es mucho más rápido que la mayor parte de sus rivales.
- 5) Portabilidad: MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- 6) Facilidad de uso: MySQL resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.
- 7) Compatibilidad con Lenguajes de Programación: MySQL posee interfaces de programación de aplicaciones (API) para distintos lenguajes de programación, como C, C++, Eiffel, Java, Perl, PHP, Python, Rubi y Tcl, lo que permite que al escribir una aplicación en cualquiera de estos lenguajes, se pueda interactuar con el motor de base de datos aprovechando todo su potencial.
- 8) Escalabilidad y Límites: Es capaz de manipular grandes bases de datos, de alrededor de 50 millones de registros, o bases de datos con 60 mil tablas y alrededor de 5000 millones de filas. El tamaño en disco que puede llegar a tener una tabla en MySQL está determinado por los límites que le impone el sistema operativo, por ejemplo sobre

un sistema operativo "Linux" con kernel 2.4 y usando el sistema de archivos "ext3" puede ser 4TB, y sobre un sistema operativo "Windows XP Professional" con sistema de archivos "NTFS" puede llegar a un tamaño máximo de 2TB aproximadamente.

1.5.8 Entorno de Desarrollo Eclipse.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Ingeniería de Modelado.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios [Eclipse2008].

Capítulo 2: Planteamiento de Soluciones.

2.1 Objetivo General.

Desarrollar los módulos de administración para las tareas de exportación, importación, backup y restauración, los cuales son nuevas funcionalidades del “Sistema de Farmacia 2.0” con tecnología Web, para lograr una exitosa implantación de dicho sistema en el “CESFAM Violeta Parra” de Chillán.

2.2 Objetivos Específicos.

- a) Desarrollar y ejecutar tareas de exportación y mapeo de datos desde la base de datos del antiguo “Sistema de Farmacia 1.0” (sistema de información heredado o SIH) en “MS-Access 97” a la nueva base de datos con distinta estructura y con formato de base de datos “MySQL 5.0”.
- b) Desarrollar y ejecutar tareas de importación de datos desde la base de datos de la “Plataforma Virtual del CESFAM Violeta Parra” (Sistema SOME) para chequear la consistencia e integridad de la base de datos del Sistema de Farmacia 1.0, antes de migrarlos al nuevo Sistema de Farmacia 2.0 e integrar la base de datos de este último con el Sistema SOME.
- c) Desarrollar una funcionalidad que permita a los usuarios el cierre anual de transacciones de dispensación a pacientes, pedidos a abastecimiento, distribución entre farmacias, inventario de medicamentos, bincards de productos, moviendo los datos respectivos a tablas historial de la base de datos.
- d) Desarrollar una funcionalidad que permita la creación de respaldos (backup) de la base de datos del Sistema de Farmacia 2.0, que se ejecute desde la aplicación misma.
- e) Desarrollar funcionalidad para la restauración de la base de datos del “Sistema de Farmacia 2.0”, que se ejecute en forma independiente.

2.3 Estrategias para implementar soluciones.

La estrategia para solucionar la problemática de la migración de los datos desde el “Sistema Heredado de Farmacia 1.0” al nuevo “Sistema de Farmacia 2.0” se muestra en la figura (ver *Fig. 9*) y se aborda en detalle en los capítulos 5 y 6. En general la estrategia consiste en aplicar elementos de la metodología de migración “Butterfly” vista en el marco teórico; que se implementa a través de almacenes temporales de datos y una aplicación que transforma los datos del modelo heredado al nuevo modelo. En cuanto a los módulos de administración para backup, restauración y cierre de transacciones anuales, la solución es abordada a través del ciclo de desarrollo de aplicaciones iterativo incremental, y arquitectura de diseño MVC.

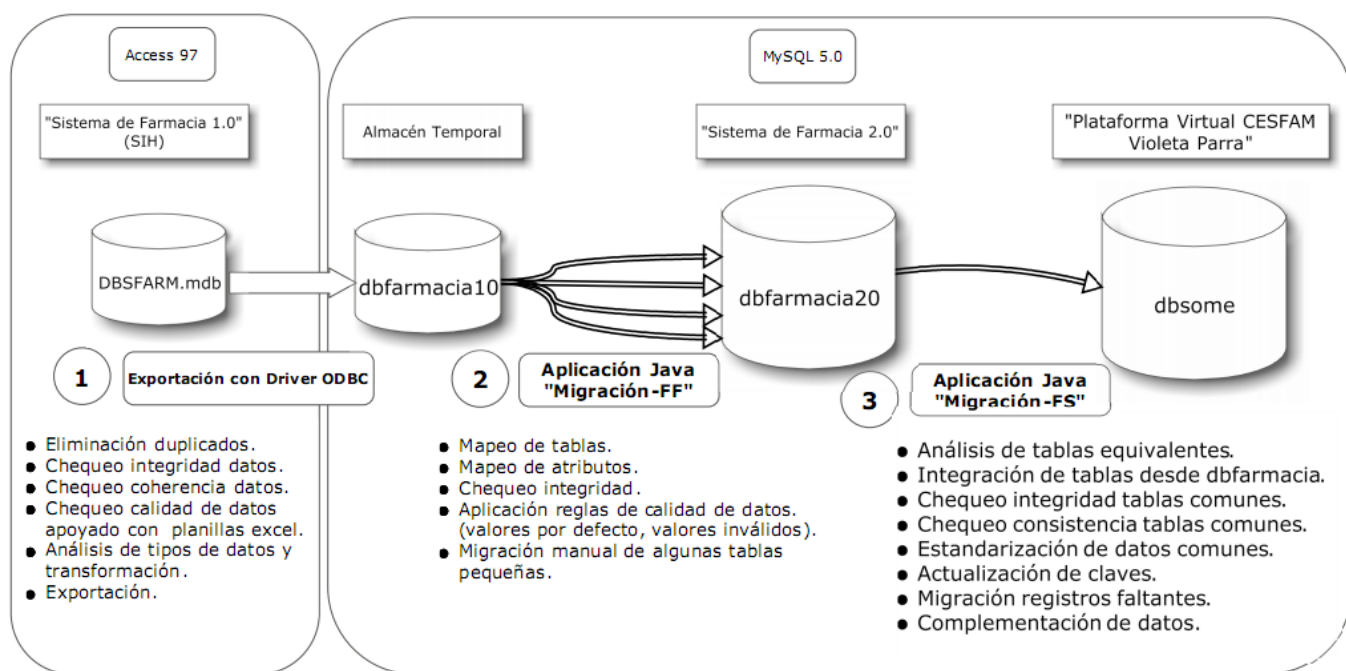


Fig. 9: Estrategia de migración de datos Sistema de Farmacia.

Capítulo 3: Estudio de Factibilidad.

Antes de ejecutar un proyecto, sea cual sea su naturaleza, es necesario revisar diferentes alternativas de implementación considerando los recursos que involucra cada una de ellas. De esta manera se puede contrastar entre propuestas y con los recursos disponibles en la organización. Finalmente con estos elementos se llega a determinar con base cierta los beneficios que se obtienen al implementar cada alternativa y el impacto del proyecto en la organización medido en los beneficios, para decidir si se lleva a cabo la ejecución y a través de cual propuesta.

3.1 Factibilidad Técnica.

Las farmacias del CESFAM cuentan con los equipos computacionales necesarios para la instalación del sistema. Dichos equipos están actualizados, ya que fueron adquiridos dentro del plan de renovación de equipos que se ejecutó en 2008, ya que la organización pasó de ser un Consultorio de atención primaria a ser un CESFAM, lo que trajo consigo un aumento en el presupuesto de inversión en infraestructura y equipos de distinto tipo.

El CESFAM también cuenta con una red LAN, con nodos en distintas dependencias incluidas las farmacias donde opera el Sistema Heredado de Farmacia 1.0.

También se adquirió un equipo servidor para la instalación y funcionamiento de los sistemas informáticos del CESFAM. En dicho equipo servidor está instalado y en funcionamiento el sistema SOME o “Plataforma Virtual del CESFAM” y se incluye la instalación y puesta en marcha del “Sistema de Farmacia 2.0”.

El software necesario para el proyecto, en la fase de desarrollo de los módulos de administración, y la disponibilidad en el CESFAM es el siguiente:

1. Sistema Operativo para Computador Servidor de Desarrollo y Pruebas. (Licencia Disponible Windows XP Professional).
2. Software Navegador Mozilla Firefox 3.0 para Win32. (Disponible).

3. Entorno de Desarrollo Eclipse 3.0 para Windows. (Software Libre).
4. Lenguaje de Programación JAVA J2SE5 para Windows. (Software Libre).
5. Software Servidor de Base de Datos MySQL 5.0 NT. (Software Libre).
6. Software Servidor de Aplicaciones Web Tomcat 6.0. (Software Libre)
7. Lenguajes de Programación Interfaz Web (HTML, JSP) (Software Libre).
8. SGBD MySQL Administrator 1.2 para Windows. (Software Libre).
9. SGBD MySQL-Front 3.2 versión de prueba. (Software de Prueba Gratuito).
10. SGBD Microsoft Access 97. (Licencia Disponible).

En la fase de explotación del “Sistema de Farmacia 2.0”, una vez que haya sido puesto en marcha, el software necesario es el mismo de la lista anterior, excepto el de los números 3, 8, 9 y 10.

En cuanto al mantenimiento a futuro del sistema, el establecimiento de salud cuenta con los servicios de un Ingeniero de Ejecución en Computación e Informática, que está encargado del mantenimiento y desarrollo de la “Plataforma Virtual del CESFAM Violeta Parra”, de la que pasa a formar parte el “Sistema de Farmacia 2.0”.

El impacto esperado tras la puesta en marcha del “Sistema de Farmacia 2.0” y la implementación de sus módulos de administración, es que mejoraría la disponibilidad de la información que gestiona el Departamento de Farmacia, al integrarla con información de otros sistemas existentes, lo que ayuda a que se puedan tomar mejores decisiones a todo nivel en la organización. Además se soluciona el problema de caídas que sufre el actual “Sistema de Farmacia 1.0” causado por el desbordamiento del tamaño de la base de datos Microsoft Access 97, la cual no puede gestionar bases de datos de tamaño mayor a 1 Gigabyte en disco. También será posible conectar más computadores clientes para tener más

puestos de trabajo disponibles para la atención a pacientes, lo que impacta positivamente en la reducción del tiempo de espera de los últimos.

3.2 Factibilidad Operativa.

En cuanto a la factibilidad operativa, se cuenta con el compromiso y disponibilidad del personal de farmacia y de la dirección del CESFAM, quienes han estado apoyando desde el inicio del diseño del sistema de farmacia que será puesto en marcha y están llanos a aprender la utilización del sistema, al que por otra parte se le ha conservado gran parte de la funcionalidad original, agregando nuevas funcionalidades para mejorar el trabajo en farmacia y que posee interfaces de usuario similares al sistema original de farmacia, para lograr una mejor adaptación de los usuarios. Sumado a ello se cuenta con las herramientas de hardware y software para llevar a cabo la migración de los datos, y el chequeo de integridad, como de coherencia.

El actual sistema en uso, “Sistema de Farmacia 1.0”, cubre gran parte de las necesidades de información de las farmacias del CESFAM, pero por tratarse de un software que lleva 9 años en explotación, se visualiza que está llegando al final de su ciclo de vida, ya que es cada vez más costoso hacerle mantenciones, debido a que la base de datos en formato Microsoft Access 97, llega constantemente a su tamaño límite de 1 Gigabyte en disco duro, lo que ocasiona caídas del sistema y deterioro en la calidad de la atención de farmacia por los tiempos de espera que esto causa a los pacientes y funcionarias.

3.3 Factibilidad Económica.

Los costos de desarrollo de la migración que incorpora software, hardware y servicios de ingeniería de ejecución informática, son los siguientes:

Detalle	Costo Unitario (\$)	Sub-Total
Hardware		
1 Computador Servidor	\$850.000.-	\$850.000.-
4 Estaciones de Trabajo (PC)	\$400.000.-	\$1.600.000.-
Software		
5 Sistemas Operativos Windows XP Professional	\$90.000.-	\$450.000.-
5 Software Navegador Mozilla Firefox 3.0 para Win32	\$0.-*	\$0.-
1 IDE Desarrollo Eclipse 3.0 para Windows	\$0.-*	\$0.-
Lenguaje de Programación JAVA J2SE5 para Win	\$0.-*	\$0.-
1 Software Servidor de Base de Datos MySQL 5.0 NT	\$0.-*	\$0.-
1 Software Servidor de Aplicaciones Web Tomcat 6.0	\$0.-*	\$0.-
Lenguajes de Programación Webapp (HTML, JSP)	\$0.-*	\$0.-
SGBD MySQL Administrator 1.2 para Windows	\$0.-*	\$0.-
SGBD Microsoft Access 97	\$120.000.-	\$120.000.-
Recursos Humanos		
1 Servicios Ingeniero de Ejec. En Comp. E Informática	\$6.960.000.-	\$6.960.000.-
	TOTAL	\$9.980.000.-

(*)Por tratarse de software libre.

Tabla 2: Costos del Proyecto Implantación Sistema de Farmacia 2.0 y Desarrollo de Módulos de Administración bajo Sistema Operativo Windows XP Professional.

El equipamiento de hardware, ya existe en el CESFAM como anteriormente se mencionó. El software requerido, SGBD Access 97, los sistemas operativos Windows XP Professional de las estaciones de trabajo y del computador servidor, ya poseen licencia. El software servidor de base de datos MySQL 5.0, el servidor de aplicaciones Web Apache Tomcat 6.0 y el SGBD MySQL Administrator 1.2, están instalados previamente para la ejecución del sistema “Plataforma Virtual del CESFAM Violeta Parra”. El software que tiene costo cero, posee licencias de tipo GPL (General Public License) y LGPL (Legal General Public License) que pueden adquirirse en forma gratuita, descargando el software vía internet, para usarlo sin costo para la organización.

También es posible revisar los costos en que se incurriría, con otra alternativa de sistemas operativos, como es “Linux Ubuntu” de la empresa “Canonical Ltda.”.

Detalle	Costo U. (\$)	Sub-Total
Hardware		
1 Computador Servidor	\$850.000.-	\$850.000.-
4 Estaciones de Trabajo (PC)	\$400.000.-	\$1.600.000.-
Software		
1 Sistema Operativo Linux Ubuntu Server 8.04 64-bits	\$0.-	\$0.-
4 Sistemas Operativos Linux Ubuntu Desktop 8.04 32-bits	\$0.-	\$0.-
5 Software Navegador Mozilla Firefox 3.0 para Linux	\$0.-	\$0.-
1 IDE Desarrollo Eclipse 3.0 para Linux	\$0.-	\$0.-
Lenguaje de Programación JAVA J2SE5 para Linux	\$0.-	\$0.-
1 Software Servidor de Base de Datos MySQL 5.0 Linux	\$0.-	\$0.-
1 Software Servidor de Aplicaciones Web Tomcat 6.0	\$0.-	\$0.-
Lenguajes de Programación Webapp (HTML, JSP)	\$0.-	\$0.-
SGBD MySQL Administrator 1.2	\$0.-	\$0.-
SGBD Microsoft Access 97	\$120.000.-	\$120.000.-
Recursos Humanos		
1 Servicios Ingeniero de Ejec. En Comp. E Informática	\$6.960.000.-	\$6.960.000.-
	TOTAL	\$9.530.000.-

Tabla 3: Costos del Proyecto Implantación Sistema de Farmacia 2.0 y Desarrollo de Módulos de Administración bajo Sistema Operativo Linux Ubuntu.

Los beneficios obtenidos con la ejecución del proyecto, son los siguientes:

Liberar de horas de trabajo en análisis, diseño y programación al Ingeniero Informático del CESFAM, haciendo posible que dedique dichas horas de trabajo a la construcción y mejora de otros sistemas informáticos en el establecimiento asistencial, lo que se traduce en ahorro de recursos económicos.

Ahorro de recursos económicos, tiempo y aseguramiento de la integridad de los datos al automatizar la migración de la información, prescindiendo de la contratación de servicios de digitadores para el ingreso de datos históricos de dispensación y datos personales de pacientes.

Servir de base para nuevos proyectos que requieran migración de datos heredados a nuevos sistemas en el CESFAM.

Utilizar un motor de base de datos (MySQL) más robusto que el utilizado hasta ahora (MS-Access 97), lo que permite almacenar mayor cantidad de datos y evitar

realizar eliminación de registros cuando la base de datos alcanza 1Gb de tamaño en disco duro.

Posibilitar la puesta en marcha del “Sistema de Farmacia 2.0”, con tecnología Web, apoyando de forma directa la disponibilidad de información de farmacia, como costo y gasto asociado a consumo de medicamentos, para seguimiento y construcción de información relevante para la toma de decisiones presupuestarias y de compra de medicamentos e insumos en la Jefatura de Farmacia, así como decisiones de gestión en el nivel de Dirección del CESFAM.

Es posible realizar una valoración económica de las horas de trabajo invertidas en la realización del proyecto y ahorradas al Ingeniero Informático del CESFAM, junto con el ahorro de recursos al no tener que contratar los servicios de digitadores como se dijo anteriormente. La tabla muestra dicha valoración económica.

Cantidad	Item	Total
1392	Horas (8 hrs. diarias en días hábiles, por espacio de 8 meses), de 1 Ingeniero Informático dedicado al análisis, diseño y programación del proyecto.(\$5.000 por hora)	\$6.960.000.-
1125	Horas de trabajo de digitación repartidas entre 5 digitadores, 5 horas al día, por 45 días para ingreso de 695000 registros históricos de farmacia a \$3000.- la hora.	\$3.375.000.-
	Total ahorro	\$10.335.000.-

Tabla 4: Valor total de horas de trabajo ahorradas al ejecutar proyecto.

Con los datos anteriores se elabora el flujo de caja del proyecto.

Item	Mes 0	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7
Ahorros (ver Tabla 4)	-	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.426
Inversión (ver Tabla 2)	(9.980.000)							
Flujo Neto de Caja	(9.980.000)	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.426

Tabla 5: Flujo de caja del proyecto evaluando solución bajo sistema operativo Windows XP.

Item	Mes 0	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7
Ahorros (ver Tabla 4)	-	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.426
Inversión (ver Tabla 2)	(9.530.000)							
Flujo Neto de Caja	(9.530.000)	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.429	1.476.426

Tabla 6: Flujo de caja del proyecto evaluando solución bajo sistema operativo Linux Ubuntu.

El Valor Actual Neto (VAN), se calcula de acuerdo a la siguiente fórmula.

$$VAN = -I + \sum_{n=1}^N \frac{Q_n}{(1+i)^n}$$

Fórmula del valor actual neto (VAN).

Donde: I = Inversión en el proyecto.

N = Número de periodos.

Q_n = Flujos de caja estimados.

i = Interés.

Fórmula 1: Cálculo del Valor Actual Neto (VAN)

Calculando el VAN con los resultados estimados en cada mes de los flujos de caja de las tablas 5 y 6, con una tasa de interés del 0,8% realizando la evaluación a 7 meses, se obtienen los siguientes resultados.

$$\text{VAN (0,8\%)} = \frac{1.476.429}{(1+0,008)^1} + \frac{1.476.429}{(1+0,08)^2} + \frac{1.476.429}{(1+0,08)^3} + \frac{1.476.429}{(1+0,08)^4} + \frac{1.476.429}{(1+0,08)^5} + \frac{1.476.429}{(1+0,08)^6} + \frac{1.476.429}{(1+0,08)^7} - 9.980.000$$

$$\text{VAN} = 32.061,353$$

Fórmula 2: Aplicación de la fórmula del VAN al proyecto considerando solución bajo sistema operativo Windows XP.

$$\text{VAN (0,8\%)} = \frac{1.476.429}{(1+0,008)^1} + \frac{1.476.429}{(1+0,08)^2} + \frac{1.476.429}{(1+0,08)^3} + \frac{1.476.429}{(1+0,08)^4} + \frac{1.476.429}{(1+0,08)^5} + \frac{1.476.429}{(1+0,08)^6} + \frac{1.476.429}{(1+0,08)^7} - 9.530.000$$

$$\text{VAN} = 482.061,353$$

Fórmula 3: Aplicación de la fórmula del VAN al proyecto considerando solución bajo sistema operativo Linux Ubuntu.

Como se aprecia en las fórmulas 2 y 3, el resultado de la evaluación del proyecto a 7 meses con la fórmula VAN y un interés del 0,8 % arroja un resultado positivo en ambos casos de implementación, lo que indica que el proyecto es económicamente viable en cualquiera de los dos casos, más aún considerando que gran parte del software y hardware requerido está disponible en la organización. Por otra parte es probable que se opte por la alternativa bajo servidor con sistema operativo Windows XP, ya que se encuentra disponible y en funcionamiento en el CESFAM.

Como conclusión del estudio de factibilidad, se puede afirmar que técnicamente los equipos y software necesarios para el proyecto están disponibles en la organización, operativamente existe compromiso de la organización, sumado a la necesidad de la

puesta en marcha del nuevo "Sistema de Farmacia 2.0". Por lo tanto el estudio indica que el proyecto es viable y puede cubrir las necesidades del CESFAM.

Capítulo 4: Estudio del Sistema Informático que se Pondrá en Marcha.

En todo proceso de migración de sistemas de información heredados (SIH), un factor de éxito determinante, es la comprensión que se logra del SIH, por parte de las personas que ejecutan el proceso. Lo anterior incluye el entendimiento tanto del modelo de datos, como de la lógica del negocio, para poder cubrir todos los requisitos en la nueva aplicación que se desarrolla, incorporar nuevas funcionalidades y migrar la base de datos correctamente.

En el caso de la migración que se ha de realizar, es igualmente importante el factor comprensión, dado que se están migrando datos, el estudio está centrado en el modelo de datos, el que se aborda en el análisis y diseño de las aplicaciones para migración en los capítulos 4 y 5; sin descuidar la lógica del negocio, que viene a explicar el “para que” de muchas tablas y atributos de la base de datos.

4.1 Requerimientos Funcionales y No Funcionales del Sistema de Farmacia 2.0 del CESFAM Violeta Parra.

Los requerimientos funcionales de un sistema, se definen como “lo que el sistema habrá de hacer”, es decir, el detalle de todas las funcionalidades que el software debe poseer para cubrir las necesidades del trabajo para el que es requerido.

El siguiente listado muestra los requerimientos funcionales, que cubre el “Sistema de Farmacia 2.0” del CESFAM “Violeta Parra”.

Ref.	Función
R1	Permitir ingresar y registrar datos de medicamentos e insumos administrados por el departamento de farmacia.

R2	<p>Registrar las recetas y prescripciones dispensadas a los pacientes del CESFAM y pacientes en tránsito o de paso, que acuden a las farmacias de morbilidad, crónicos y servicio SAPU, además de registrar los profesionales que emiten cada receta y prescripción.</p>
R3	<p>Clasificar a los pacientes que acuden a farmacia de acuerdo al tipo de atención médica que se les brindó, a la enfermedad que se les diagnosticó y el programa de tratamiento que les haya asignado el médico tratante o profesional de la salud, además de actualizar estos y otros datos del paciente en la base de datos integrada con la “Plataforma Virtual del CESFAM Violeta Parra” (Sistema SOME).</p>
R4	<p>Controlar y registrar la frecuencia de asistencia de los pacientes crónicos a farmacia y la fecha de citación al siguiente control médico para dispensación de medicamentos.</p>
R5	<p>Administrar la tarjeta de dispensación de farmacoterapia prolongada de pacientes crónicos, registrando las entregas de medicamentos e insumos que se realizan en detalle.</p>
R6	<p>Generar reportes sobre qué medicamentos e insumos están siendo consumidos, por cuál paciente y en qué cantidad.</p>
R7	<p>Controlar y registrar stock, ingresos, despachos, dispensaciones y generar bindcards con totales por periodo solicitado o detalle de movimientos si se desea, de los insumos y medicamentos que poseen cada una de las farmacias del CESFAM (Morbilidad, Crónicos, SAPU).</p>

R8	<p>Gestionar y registrar en línea, los pedidos, la recepción y las devoluciones de insumos y medicamentos, desde farmacia general al departamento de abastecimiento, indicando las causas de cada operación, teniendo la posibilidad de agregar nuevas causas.</p>
R9	<p>Registrar las operaciones de Distribución del stock de insumos y medicamentos desde farmacia general, hacia las farmacias externas al CESFAM, como postas y otras que soliciten artículos.</p>
R10	<p>Registrar los fármacos sometidos a control legal como psicotrópicos y estupefacientes de uso médico que se almacenan y despachan en las farmacias del CESFAM, avisando en caso de que se esté entregando en segunda o posterior oportunidad el mismo medicamento, a un mismo paciente y en un mismo día. Además para este tipo de medicamentos se debe registrar el nombre y RUN tanto del Profesional que lo prescribe como del Paciente que lo recibe, las cantidades recetadas y entregadas, restando de estas últimas las dosis no entregadas por atraso del paciente en la fecha de retiro.</p>
R11	<p>Registrar los medicamentos a dispensar que deben ser cancelados en Recaudación por los pacientes cuya previsión lo requiera. (Particulares, FFAA, Gendarmería y Carabineros), cerrando mensualmente el registro de estas prestaciones y anulando las dispensaciones que no hayan sido entregadas efectivamente a los pacientes, al momento del cierre.</p>
R12	<p>Generar informes sobre prescripciones, consumo y stock de medicamentos, diagnósticos de enfermedades y censos de pacientes por diferentes criterios.</p>

R13	Poseer distintos perfiles de usuario, para controlar las funcionalidades a las que tiene acceso cada persona que utiliza el sistema.
-----	--

Los requisitos no funcionales se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar software [Bredemeyer, 2001].

En cuanto a los requerimientos no funcionales del Sistema de Farmacia 2.0 son los siguientes:

Atributos del Sistema	Detalles y Restricción
Facilidad de uso.	Interfaz de usuario semejante al Sistema Heredado de Farmacia 1.0, que ya es conocido.
Tiempo de Respuesta.	Máximo 5 segundos desde que se hace petición hasta que se muestra primer resultado en pantalla.
Soporte Sistema Operativo.	Microsoft Windows XP Profesional o Linux (Kernel v2.6).
Soporte Servidor de Aplicaciones.	Apache Tomcat Versión 6.0.16 para aplicaciones web Java JSP y J2EE.
Soporte Servidor de Base de Datos	MySQL Server Versión 5.0.24a Community
Soporte Máquina Virtual JAVA.	JAVA J2SE Versión 5 ó 6.
Metáfora de Interfaz del Sistema.	Interfaz Web con gráficos.

4.2 Casos de Uso del Sistema de Farmacia 2.0.

En el proceso de desarrollo de software, específicamente durante la etapa de análisis, se utilizan una serie de técnicas y herramientas para la captura de requisitos como el modelo de casos de uso.

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Cada forma en que se usará el sistema se representa con un caso de uso. Los casos de uso son fragmentos de funcionalidad del sistema y en forma más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con los usuarios, incluyendo acciones alternativas dentro de la secuencia. Por lo tanto a través de los casos de uso es posible conocer a fondo el modelo de sistema que se está construyendo o que está construido como en el caso del “Sistema de Farmacia 2.0”.

Los casos de uso que constituyen el Sistema de Farmacia 2.0 son los siguientes mostrados en la tabla:

Casos de Uso “Sistema de Farmacia 2.0”	
1	Ingresar Datos de Autenticación.
2	Mantener Datos Consultorio.
3	Mantener Datos Nuevo Usuario del Sistema.
4	Mantener Datos Farmacia General.
5	Mantener Datos Farmacia Crónicos.
6	Mantener Datos Farmacia Morbilidad.
7	Mantener Datos Farmacia SAPU.
8	Mantener Datos Farmacia Externa.
9	Mantener Paciente Crónico.
10	Mantener Paciente Morbilidad.
11	Mantener Grupo Etario.
12	Mantener Programa de Salud.

13	Enviar Pedido de Insumos y Medicamentos a Abastecimiento.
14	Recibir Pedido de Insumos y Medicamentos.
15	Ingresar Pedido de Insumo o Medicamento Especial.
16	Ingresar Identificación de Nuevo Insumo o Medicamento a Inventario.
17	Distribuir Insumos y Medicamentos a Farmacias del CESFAM.
18	Registrar Despacho de Insumos y Medicamentos a Farmacia Externa.
19	Dar de Baja Insumo o Medicamento.
20	Devolución de Insumo o Medicamento.
21	Ingresar Receta para dispensación desde Farmacia Morbilidad.
22	Ingresar Receta para dispensación desde Farmacia Crónicos.
23	Ingresar Receta para dispensación desde Farmacia SAPU.
24	Generar Informe de Stock, Entrada y Salida de Artículos por Farmacia.
25	Generar Informe de Stock, Entrada y Salida de Artículos en General.
26	Visualizar Bindcard de Insumo o Medicamento.
27	Solicitar Informe de Entregas de Medicamentos Controlados.
28	Solicitar Informe de Insumos y Medicamento Devueltos.
29	Solicitar Informe de Pacientes Inasistentes por Periodo.
30	Solicitar Informe de Consumo por Familia.
31	Solicitar Informe de Consumo por Paciente.
32	Solicitar Informe de Medicamentos Prescritos por Profesional.
33	Solicitar Informe Censo de Pacientes por Grupo Etario y Tipo de Atención.
34	Solicitar Informe de Pacientes con Diagnóstico Climaterio.

La lista de casos de uso, nos da una idea sobre cada funcionalidad del “Sistema de Farmacia 2.0”, y como el proceso de desarrollo de dicho sistema no es el objetivo del presente trabajo, se incluye la especificación detallada de casos de uso en la sección anexos.

4.3 Diagrama de Clases.

Como se describe en el marco teórico (1.3.3) el enfoque de desarrollo orientado a objetos, permite modelar sistemas, representando objetos del universo

del problema y sus interacciones. Este modelamiento puede especificarse y hacerse visual, mediante diagramas UML, como es el caso de un diagrama de clases que especifica cuales objetos se implementan en el software y cómo interactúan entre sí.

En la figura siguiente se muestra el diagrama de clases del “Sistema Heredado de Farmacia 1.0”, que sirvió como base para el desarrollo del “Sistema de Farmacia 2.0” que es el sistema en proceso de ser implantando.

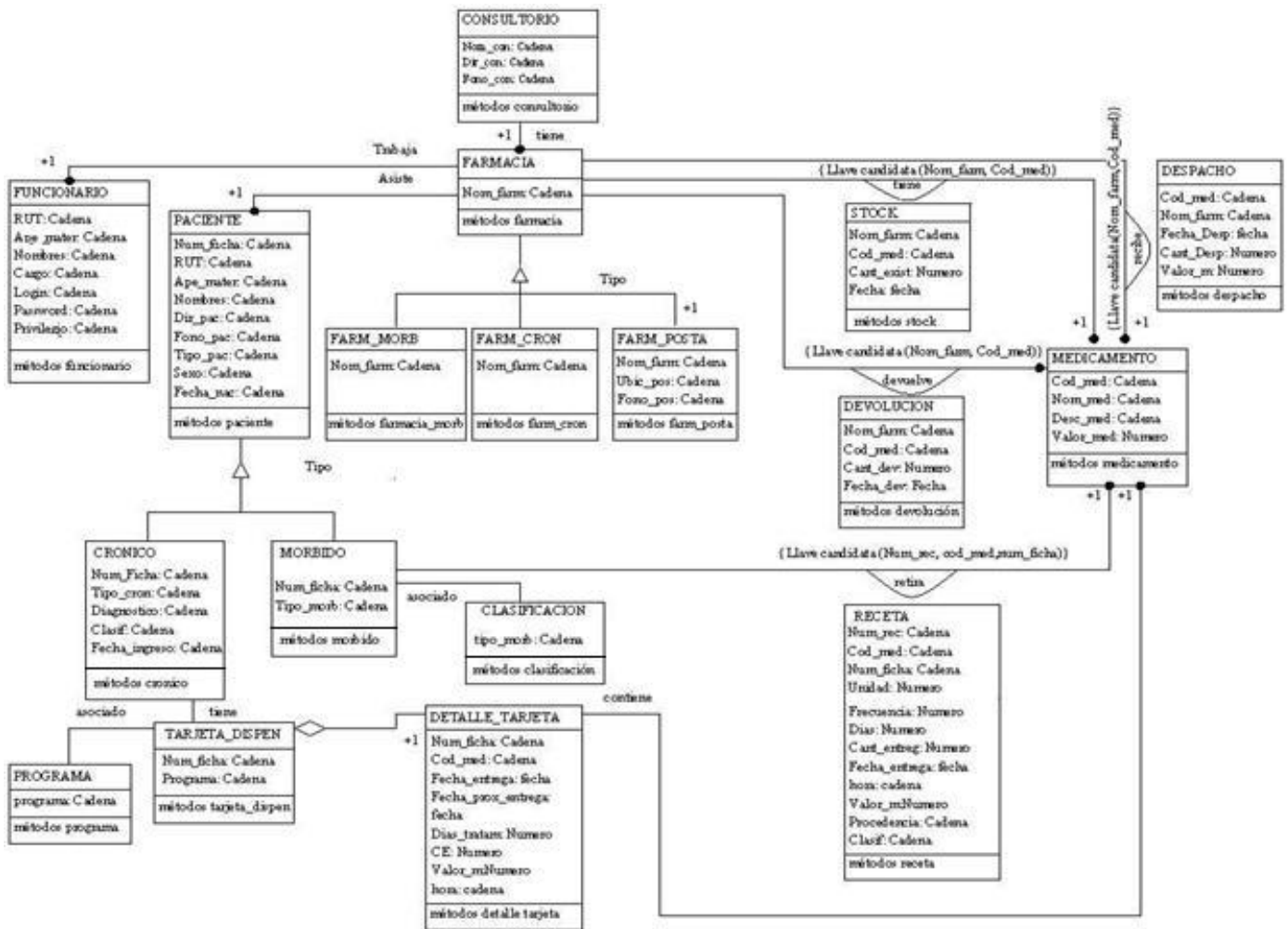


Fig. 10: Diagrama de clases de diseño Sistema de Farmacia 1.0.[Hermosilla 2000].

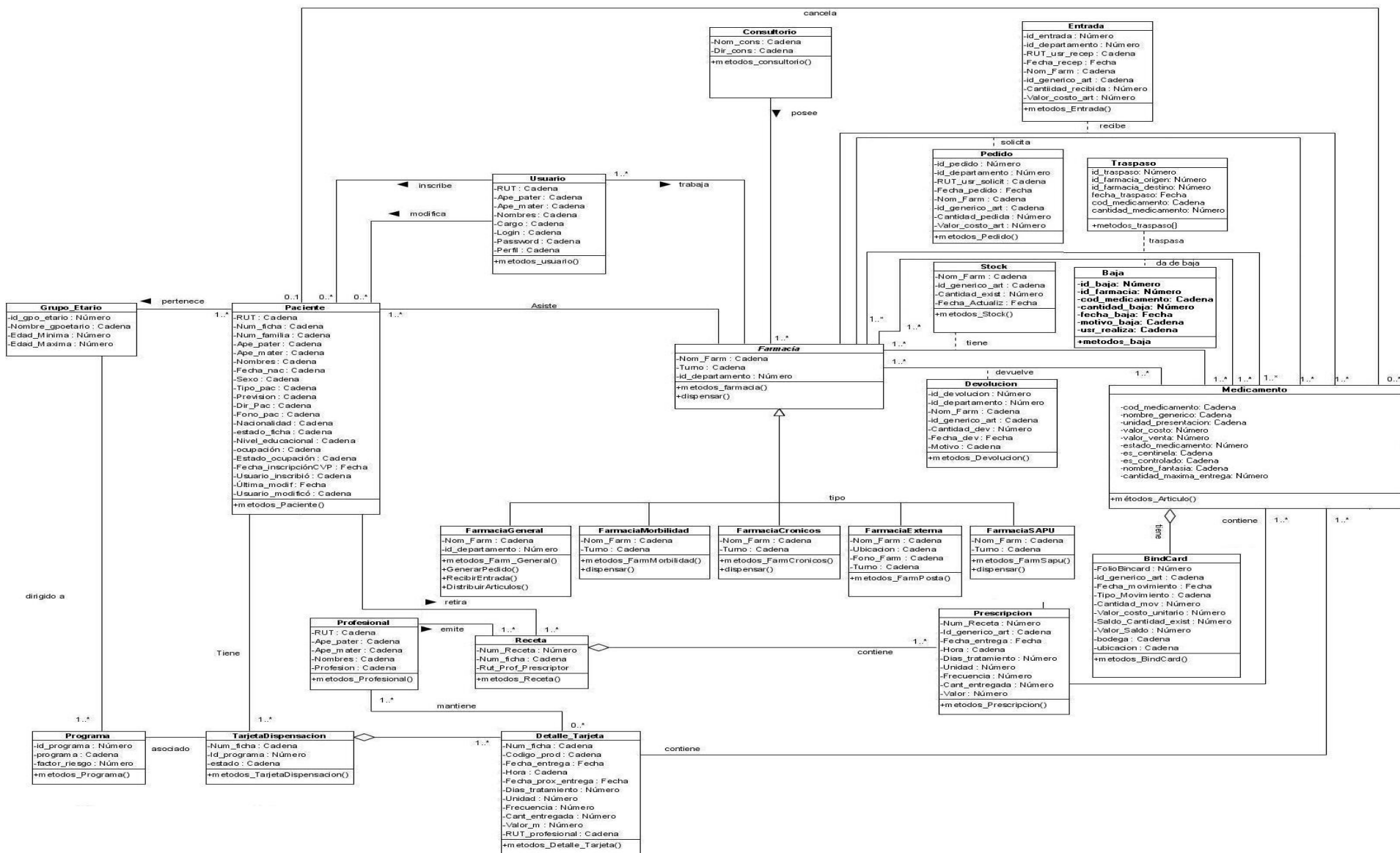


Fig. 11: Diagrama de Clases del Sistema de Farmacia 2.0.

4.4 Modelo de Datos.

El modelo de datos es fundamental para entender el sistema en estudio, y lograr una exitosa migración de los datos desde el sistema heredado. En primer lugar se muestra la descripción de las tablas de la base de datos “dbfarmacia20” y a continuación se presenta el diagrama Modelo Entidad Relación evolucionado, donde se muestran las tablas, relaciones y relaciones tablas del sistema.

La siguiente es la descripción de las tablas de la Base de Datos “dbfarmacia 2.0” implementada en MySQL 5.0.

Arancel	
<i>Atributo</i>	<i>Tipo</i>
Código	varchar(11)
id_prevision	int(11)
Detalle	varchar(550)
Precio	int(11)

La tabla arancel almacena los valores de las prestaciones de salud (insumos y medicamentos) que los pacientes deben cancelar de acuerdo a su previsión.

Baja	
<i>Atributo</i>	<i>Tipo</i>
id_baja	int(10)
id_farmacia	int(10)
cod_medicamento	varchar(20)
cantidad_baja	int(10)
fecha_baja	Datetime
motivo_baja	varchar(45)
usr_realiza	varchar(45)

La tabla Baja guarda los detalles de las transacciones cuando se da de baja un insumo o medicamento, desde el inventario de farmacia.

Beneficiario_Prog	
<i>Atributo</i>	<i>Tipo</i>
run_paciente	varchar(20)
id_programa	int(11)
fecha_ingreso	varchar(40)
fecha_egreso	varchar(40)
estado_paciente	int(11)
usr_crea	varchar(50)
fecha_crea	varchar(40)
usr_modifica	varchar(50)
fecha_modifica	varchar(40)

La tabla Beneficiario_Prog relaciona un paciente con los diferentes programas de salud de los que es beneficiario.

Bincard	
<i>Atributo</i>	<i>Tipo</i>
folio_movimiento	int(10)
cod_medicamento	varchar(20)
tipo_movimiento	varchar(10)
fecha_mov	Datetime
cantidad_mov	int(11)
id_farmacia	int(10)
detalle_movimiento	varchar(255)
usr_crea	varchar(255)
fecha_crea	datetime
usr_modif	varchar(255)
fecha_modif	datetime

La tabla bincard almacena los movimientos (entradas y salidas) en cantidad de unidades, que se producen en inventario para los insumos y medicamentos.

Cesfam	
<i>Atributo</i>	<i>Tipo</i>
id_cesfam	int(11)
nombre_cesfam	varchar(120)
Director	varchar(120)
Dirección	varchar(120)
Fono	varchar(15)
Email	varchar(120)

La tabla CESFAM almacena la información básica del establecimiento de salud, utilizada para mostrarla en informes del sistema.

Cobertura_prog	
<i>Atributo</i>	<i>Tipo</i>
id_grupo_etario	int(11)
id_programa	int(11)

La tabla Cobertura_prog asocia los grupos etarios con programas de salud que están dirigidos a las personas de esos grupos etarios.

Consultorio_Procedencia	
<i>Atributo</i>	<i>Tipo</i>
Id	int(11)
Nombre	varchar(150)

La tabla Consultorio_Procedencia almacena los nombres de distintos consultorios de la ciudad, desde donde eventualmente, pueden provenir algunos pacientes, o pertenecen algunas farmacias externas que son abastecidas.

Ocupacion	
<i>Atributo</i>	<i>Tipo</i>
id_ocupacion	int(10)
nombre_ocupacion	varchar(80)

La tabla Ocupación define la lista de posibles actividades laborales en las que puede trabajar un paciente.

Despacho	
<i>Atributo</i>	<i>Tipo</i>
id_despacho	int(10)
id_farmacia_origen	int(10)
id_farmacia_destino	int(10)
fecha_despacho	varchar(40)
cod_medicamento	varchar(20)
cantidad_medicamento	int(11)
usr_crea	varchar(50)

La tabla despacho almacena las transacciones de los envíos de insumos y medicamentos, hacia farmacias externas al CESFAM.

Detalle_Tarjeta	
<i>Atributo</i>	<i>Tipo</i>
id_detalle_tarjeta	bigint(15)
run_paciente	varchar(20)
fecha_entrega	Date
id_despacho	int(10)
cod_medicamento	varchar(20)
hora_entrega	Time
fecha_prox_entrega	Datetime
dias_tratamiento	int(10)
Dosis	Double
Frecuencia	int(10)
cant_entregada	int(10)
valor_medicamentos	int(10)
Procedencia	varchar(1)
run_profesional	varchar(11)
id_diagnostico	int(11)
id_programa	int(11)
Sector	int(11)
es_ges	varchar(1)
es_controlado	varchar(2)
es_centinela	varchar(2)
usr_crea	varchar(20)
fecha_crea	varchar(40)

La tabla Detalle_Tarjeta almacena los detalles de las transacciones de dispensación de insumos y medicamentos a los pacientes crónicos del CESFAM.

Diagnostico_Comun	
<i>Atributo</i>	<i>Tipo</i>
id_diagnostico_comun	int(11)
descripcion_diag_comun	varchar(100)

La tabla Diagnostico_Comun contiene la nómina de enfermedades diagnosticadas comúnmente, y es usada por otras tablas.

Devolución	
<i>Atributo</i>	<i>Tipo</i>
id_devolucion	int(10)
id_farmacia	int(10)
cod_medicamento	varchar(20)
fecha_devolucion	Datetime
cantidad_devuelta	int(10)
motivo_devolucion	varchar(120)

La tabla Devolución almacena las transacciones de insumos o medicamentos devueltos al departamento de abastecimiento por diversos motivos, como vencimiento, prohibiciones sanitarias, mal estado, entre otras.

Diagnostico_Ges	
<i>Atributo</i>	<i>Tipo</i>
id_diagnostico_ges	int(11)
diagnostico_ges	varchar(250)
nombre_corto_ges	varchar(80)

La tabla Diagnostico_Ges contiene la nómina de enfermedades diagnosticadas comúnmente y que son parte del “Plan de Garantías Explícitas en Salud” del Estado de Chile.

Distribución	
<i>Atributo</i>	<i>Tipo</i>
id_distribucion	int(11)
fecha_distribucion	Datetime
cod_medicamento	varchar(20)
cant_medicamento	int(11)
id_farmacia_destino	int(10)

La tabla Distribución contiene las operaciones de cantidad de insumos o medicamentos transferidos, desde el inventario de farmacia general hacia el inventario de otras farmacias internas del CESFAM, como farmacia morbilidad y farmacia crónicos.

Escolaridad	
<i>Atributo</i>	<i>Tipo</i>
id_escolaridad	int(10)
descrip_escolaridad	varchar(50)

La tabla escolaridad almacena información que define el nivel educacional alcanzado por un paciente, por ejemplo: Enseñanza Básica, Enseñanza Media, etc.

Especialidad	
<i>Atributo</i>	<i>Tipo</i>
id_especialidad	int(11)
desc_especialidad	varchar(50)

La tabla especialidad contiene la nómina de especialidades que puede poseer un profesional de la salud. Ejemplo: Traumatología.

Especializacion	
<i>Atributo</i>	<i>Tipo</i>
run_profesional	varchar(11)
id_especialidad	int(11)

La tabla especialización asocia un profesional de la salud con una o más especialidades.

Estado_Paciente	
<i>Atributo</i>	<i>Tipo</i>
id_estado_paciente	int(10)
estado_paciente	varchar(50)

La tabla Estado_Paciente contiene la definición de los estados del registro de un paciente, que pueden ser: Activo, Pasivo, Traslado, etc. Es usada por el atributo "estado" de la tabla "Paciente".

Farmacia	
<i>Atributo</i>	<i>Tipo</i>
id_farmacia	int(10)
nombre_farmacia	varchar(100)
ubicacion_farmacia	varchar(100)
fono_farmacia	varchar(20)
Encargado	varchar(100)
id_consultorio	int(11)

La tabla Farmacia contiene la información básica de cada unidad de farmacia interna del CESFAM.

Farmacia_Ext	
<i>Atributo</i>	<i>Tipo</i>
id_farmacia	int(10)
nombre_farmacia	varchar(100)
ubicacion_farmacia	varchar(100)
fono_farmacia	varchar(20)
Encargado	varchar(100)
id_consultorio	int(11)

La tabla Farmacia_Ext contiene información básica de las farmacias externas al CESFAM, a las que se les despachan insumos o medicamentos.

Grupo_Etario	
<i>Atributo</i>	<i>Tipo</i>
id_grupoetario	int(11)
edad_min	int(11)
edad_max	int(11)
nombre_gpo_etario	varchar(45)

La tabla Grupo_Etario contiene la definición de los rangos de edad definidos por el Ministerio de Salud para clasificar a los pacientes. Ejemplo: Niños de 0 a 9 años.

Historial_Diagnostico	
<i>Atributo</i>	<i>Tipo</i>
id_historial_diagnostico	int(10)
fecha_historial_diag	datetime
es_ges	int(11)
id_diagnostico	int(11)
id_diagnostico_ges	int(11)
usr_crea	varchar(50)
run_paciente	varchar(20)

La tabla Historial_Diagnostico contiene los registros de los diagnósticos de los pacientes y las fechas en que han sido establecidos.

Medicamento	
<i>Atributo</i>	<i>Tipo</i>
cod_medicamento	varchar(20)
nombre_generico	varchar(80)
unidad_presentacion	varchar(50)
valor_costo	int(11)
valor_venta	int(11)
estado_medicamento	int(11)
es_centinela	varchar(2)
es_controlado	varchar(2)
nombre_fantasia	varchar(50)
cantidad_maxima_entrega	int(11)

La tabla Medicamento contiene la descripción y las características de cada insumo o medicamento que se entrega en las farmacias del CESFAM. Además indica si el medicamento está sujeto a control legal (es controlado) y si es un medicamento centinela, es decir, que su presencia en inventario está garantizada.

Pedido_Detalle	
<i>Atributo</i>	<i>Tipo</i>
id_detalle_pedido	int(11)
id_pedido	int(10)
cod_medicamento	varchar(20)
cantidad_pedida	int(10)
cantidad_entregada	int(10)
cod_alternativo_med	varchar(20)
visto_recepcion	varchar(45)
nombre_alternativo_med	varchar(250)
valor_costo_medicam_pedido	int(10)
valor_costo_medicam_entregado	int(10)

La tabla Pedido_Detalle contiene la lista detallada de insumos y medicamentos, pedidos al departamento de abastecimiento. Además contiene atributos para registrar la operación de respuesta desde abastecimiento al pedido.

Ocupacion_Estado	
<i>Atributo</i>	<i>Tipo</i>
id_ocup_estado	int(10)
descrip_ocup_estado	varchar(50)

La tabla Ocupacion_Estado contiene la definición de los posibles estados en que puede encontrarse la situación ocupacional de un paciente.

Prevision	
<i>Atributo</i>	<i>Tipo</i>
Id	int(11)
Nombre	varchar(55)

La tabla Previsión almacena la definición de los nombres que tienen los distintos tipos de previsión que poseen los pacientes. Ej.: FONASA A, FONASA B.

Paciente	
<i>Atributo</i>	<i>Tipo</i>
Run	varchar(20)
numFicha	varchar(20)
ficha_maternal	varchar(15)
numFamilia	varchar(9)
Nombres	varchar(205)
apellido_paterno	varchar(24)
apellido_materno	varchar(25)
Previsión	int(3)
Sexo	varchar(10)
Fono	varchar(10)
Dirección	varchar(70)
Fnacimiento	Date
tipoPaciente	varchar(20)
Estado	int(1) unsigned
codigo_consultorio	int(11)
funcionarioCSVP	varchar(2)
id_escolaridad	int(10)
id_ocupacion	int(10)
id_estado_ocup	int(10)
Nacionalidad	varchar(50)
usr_crea	varchar(20)
fecha_crea	datetime
usr_modifica	varchar(20)
ultima_actualizacion	date
causal Eliminacion	varchar(105)

La tabla Paciente contiene la información de datos personales de identificación y de otros datos de control, de los pacientes del CESFAM.

Pedido	
<i>Atributo</i>	<i>Tipo</i>
id_pedido	int(10)
id_farmacia	int(10)
run_solicitante	varchar(11)
fecha_solicitud	datetime
run_despachador	varchar(11)
fecha_recepcion	datetime
run_receptor	varchar(11)
estado_pedido	varchar(45)
orden_chilecompra	varchar(45)
observaciones	varchar(255)
id_despacho_abast	int(10)

La tabla Pedido contiene, la información general de definición de una transacción de pedido al departamento de abastecimiento, por lo que le corresponden registros desde la tabla Detalle_Pedido.

Prescripcion	
<i>Atributo</i>	<i>Tipo</i>
id_prescripcion	int(11)
num_receta	varchar(20)
tipo_atencion	varchar(20)
cod_medicamento	varchar(20)
fecha_hora_entrega	datetime
Dosis	double
Frecuencia	int(11)
Días	int(11)
cant_entregada	int(11)
costo_prescripcion	int(11)
valor_venta_prescripcion	int(11)
es_ges	varchar(2)
es_controlado	varchar(2)
es_centinela	varchar(2)
usr_crea	varchar(20)
fecha_crea	datetime
usr_modif	varchar(20)
fecha_modif	datetime

La tabla prescripción contiene todos los insumos y medicamentos entregados a los pacientes de atención morbilidad. Almacena los registros que detallan las recetas definidas en la tabla de ese mismo nombre.

Profesion	
<i>Atributo</i>	<i>Tipo</i>
id_profesion	int(5)
Nombre	varchar(55)

La tabla Profesión contiene la nómina con la definición de las diferentes profesiones que pueden poseer los profesionales de la salud que atienden en el CESFAM.

Profesional	
<i>Atributo</i>	<i>Tipo</i>
Run	varchar(11)
apellido_paterno	varchar(50)
apellido_materno	varchar(50)
Nombres	varchar(200)
id_profesion	int(5)
Estado	varchar(1)

La tabla Profesional contiene los datos básicos de los Profesionales de la salud que trabajan en el CESFAM y pueden prescribir medicamentos o insumos médicos.

Programa_Salud	
<i>Atributo</i>	<i>Tipo</i>
id_programa	int(11)
Programa	varchar(80)

La tabla Programa_Salud contiene la definición de los programas de salud para los pacientes del CESFAM.

Receta	
<i>Atributo</i>	<i>Tipo</i>
num_receta	varchar(20)
fecha_receta	datetime
run_paciente	varchar(20)
run_profesional	varchar(11)
id_diagnostico	int(11)
es_controlado	varchar(2)
es_centinela	varchar(2)

La tabla Receta contiene la información general de las recetas entregadas por profesionales autorizados.

Sector	
<i>Atributo</i>	<i>Tipo</i>
Id	int(11)
Nombre	varchar(255)
id_consultorio	int(11)

La tabla Sector contiene la nómina de las áreas en que se divide el territorio de donde provienen los pacientes del CESFAM. Ej.: "Sector 1".

Stock	
<i>Atributo</i>	<i>Tipo</i>
cod_medicamento	varchar(20)
id_farmacia	int(10)
cantidad_existente	int(11)
stock_maximo	int(11)
stock_minimo	int(11)
stock_critico	int(11)
cant_max_dispensar	int(11)
fecha_actualizacion	datetime
usr_actualizo	varchar(50)

La tabla Stock contiene la información de cantidades existentes de cada insumo y medicamento en las farmacias del CESFAM, incluyendo parámetros de control del inventario como cantidad mínima y cantidad máxima.

Tarjeta_Dispensacion	
<i>Atributo</i>	<i>Tipo</i>
run_paciente	varchar(20)
fecha_ingreso	datetime
estado_cronico	int(10)
usr_crea	varchar(20)
usr_modifica	varchar(20)
fecha_modifica	Datetime

La tabla Tarjeta_Dispensacion contiene la información general de cada tarjeta de control de farmacoterapia prolongada de pacientes crónicos, información de entregas de medicamentos que es detallada en la tabla Detalle_Tarjeta.

Traspaso	
<i>Atributo</i>	<i>Tipo</i>
id_traspaso	int(10)
id_farmacia_origen	int(10)
id_farmacia_destino	int(10)
fecha_traspaso	Datetime
cod_medicamento	varchar(20)
cantidad_medicamento	int(10)

La tabla Traspaso contiene el registro de las operaciones de traspaso de unidades de insumos o medicamentos entre los inventarios de las distintas farmacias internas del CESFAM.

Usuario	
<i>Atributo</i>	<i>Tipo</i>
Rut	varchar(11)
nombre_completo	varchar(60)
Cargo	varchar(50)
Clave	varchar(128)
Perfil	varchar(50)
estado_usr	varchar(50)
usr_crea	varchar(50)
fecha_crea	datetime
usr_modif	varchar(50)
fecha_modif	datetime
id_consultorio	int(11)
ultima_visita	timestamp

La tabla Usuario contiene los datos personales y de acceso de todos los usuarios del sistema de Farmacia 2.0.

Capítulo 5: Tareas de Exportación y Mapeo de Datos desde “Sistema Heredado de Farmacia 1.0”.

5.1 Exportación desde Base de Datos del “Sistema Heredado de Farmacia 1.0” (DBSFARM.mdb) en MS-Access 97 al formato de “dbfarmacia 2.0” en MySQL.

En este punto se desarrolla la primera etapa de la estrategia de migración de los datos entre planteada en el capítulo 2.

El “Sistema Heredado de Farmacia 1.0” posee una base de datos denominada “DBSFARM.mdb” creada en formato del Sistema Gestor de Bases de Datos (SGBD) “Microsoft Access 97”, la que debe ser migrada al formato del SGBD “MySQL 5.0”, ya que, es este último el formato en que se encuentra implementada la base de datos del “Sistema de Farmacia 2.0”, que es el objetivo de la puesta en marcha. Al exportar los datos del “Sistema Heredado de Farmacia 1.0” a MySQL, se podrán tener las tablas y atributos en el mismo formato del nuevo sistema y será más transparente la lectura de datos para la aplicación encargada de migrar, chequear integridad y consistencia de datos, que se verá en el punto 4.3.

Previamente a exportar los datos es preciso chequear la consistencia e integridad de estos, entre las tablas de la misma base de datos “DBSFARM.mdb”, utilizando “Microsoft Access 97” para crear consultas de búsqueda de tuplas no coincidentes, por ejemplo, los registros de la tabla “CRONICO” que no existen en la tabla “PACIENTE” y también los registros de la tabla “MORBIDO” que no están en la tabla “PACIENTE”. Estas dos consultas, en rigor, no deberían arrojar tuplas como resultado, dado que las tablas “CRONICO” y “MORBIDO”, guardan información específica de los pacientes que buscan atención de morbilidad y de los pacientes enfermos crónicos, respectivamente, por lo que los registros de ambas tablas deberían existir en la tabla donde se almacenan los datos generales de todos los pacientes del SIH “Sistema de Farmacia 1.0” que es la tabla “PACIENTE”. Sin embargo, al ejecutar estas consultas, se encontraron algunas tuplas, lo que significa que efectivamente, la base de datos “DBSFARM.mdb” tiene algún grado de inconsistencia, probablemente debido a la edición directa que se hace de los datos, usando “Access 97”, sin considerar las restricciones de integridad al prescindir de la

aplicación encargada de administrar la base de datos: el “Sistema de Farmacia 1.0” y además porque la implementación de “DBSFARM.mdb” no contiene en sí misma, las restricciones de integridad, puesto que se validan sólo a través de la lógica del programa ejecutable “farmacia.exe” que es el “Sistema de Farmacia 1.0”.

El mismo tipo de consulta se debe realizar entre las tablas “PACIENTE” Y “TARJETA_DISPEN”; y también entre “TARJETA_DISPEN” y “DETALLE_TARJETA”. Estas dos últimas, guardan los datos de control de las tarjetas de dispensación de farmacoterapia prolongada de pacientes crónicos.

Otras consultas del mismo tipo son las siguientes:

- Tuplas de “RECETA” que no coinciden con “PACIENTE”.
- Tuplas de “RECETA” que no coinciden con “MORBIDO”.

Una vez que se ha chequeado esta parte de la consistencia e integridad interna de la base de datos, se continúa con la tarea de exportación.

La exportación de datos se realiza a través de un driver ODBC (Open Database Connectivity) desde el Sistema Gestor de Bases de Datos (SGBD) “Microsoft Access 97”, a través de una conexión configurada con el driver ODBC hacia una base de datos temporal llamada “dbfarmacia10” que fue creada con el SGBD “MySQL Administrator versión 1.2.12” para formato de base de datos MySQL 5.0 como se ilustra en la *Fig. 13*.

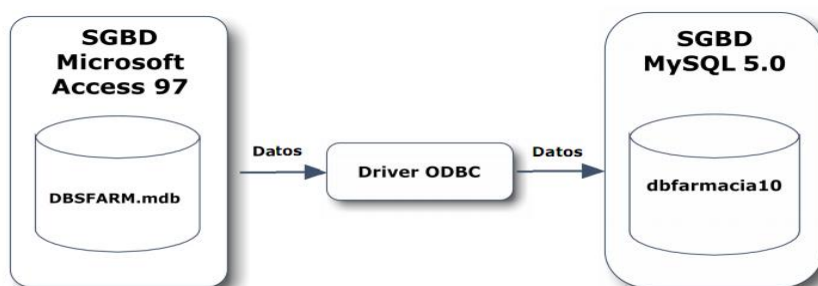


Fig. 13: Diagrama del proceso de exportación de datos de Access 97 a MySQL 5.0.

MS-Access 97 provee la posibilidad de exportar las tablas de una base de datos que está en formato propietario “mdb”, hacia un archivo o base de datos externa en otro formato distinto, como por ejemplo: “dbf” (base de datos Dbase), “xls” (Planilla Excel), “txt” (archivo de texto), bases de datos ODBC, entre otros; a fin de permitir la portabilidad de los datos entre aplicaciones. La funcionalidad para exportar a bases de datos ODBC, no reside completamente en el SGBD MS-Access 97, si no que es implementada al crear una conexión hacia otra aplicación, por medio de un software denominado driver ODBC, que a su vez implementa ese estándar creado por la empresa Microsoft Corporation, con el objetivo de “hacer posible acceder a cualquier dato desde cualquier aplicación sin importar el SGBD que los creó y que los almacena”. [Wikipedia 2009].

El driver es una capa intermedia entre la aplicación y el SGBD. Dicha capa se encarga de la traducción de las consultas de datos (lectura, inserción, eliminación, etc.) de la aplicación para que el SGBD las entienda. En el caso de esta exportación se utilizó el driver ODBC “MySQL Connector ODBC 5.1” fabricado por MySQL AB.

Los pasos y herramientas para realizar la exportación de los datos están determinados además por el sistema operativo del computador donde se llevan a cabo. En este caso, las aplicaciones están instaladas en una máquina con el sistema operativo “Microsoft Windows XP Professional Service Pack 3”, por lo que es necesario configurar un origen de datos de sistema o DSN (Data Source Name). A través de este DSN de sistema configurado, se indica, entre otros parámetros, cual es el driver que se utilizará para la conexión y cuál es la base de datos a la que se podrá acceder. Esto es aplicable también a otros sistemas operativos de la Empresa Microsoft como Windows 95, NT, 98, ME, 2000, 2003 Server y versiones de Windows Vista.

La configuración del “DSN de sistema” se realiza en el “Panel de Control” de Windows, específicamente en el “Administrador de Orígenes de Datos” y se crea siguiendo los siguientes pasos:

1. Previamente a crear el “DSN de Sistema” deben estar instalados: el software servidor de bases de datos “MySQL 5.0” en el computador local, el SGBD

“MySQL Administrator” y debe estar creado el esquema que guardará las tablas que serán exportadas, en este caso el esquema se denomina “dbfarmacia10” que es una base de datos que se crea vacía, sin tablas. Además debe estar instalado el SGBD “MS-Access 97” y se debe haber almacenado una copia de la base de datos “DBSFARM.mdb” desde el computador servidor de farmacia, en la máquina local, donde se hará la operación de exportación.

2. Se instala el driver ODBC para acceso a bases de datos MySQL, denominado “MySQL Connector/ODBC 5.1”, el cual puede ser obtenido en forma gratuita desde el sitio web oficial de la empresa MySQL AB: <http://www.mysql.net>.
3. Terminada la instalación del punto anterior, se abre el “Panel de Control de Windows” y desde éste se inicia el “Administrador de Orígenes de Datos”.
4. Se escoge la pestaña correspondiente a “DSN de Sistema” y se hace click en el botón “Agregar...”. Ésta acción mostrará un formulario para crear un nuevo origen de datos, donde se debe elegir un driver o controlador para el cual se definirá el nuevo origen de datos. En este caso se elige el driver instalado anteriormente “MySQL ODBC Driver 5.1”. (**Fig. 14**).

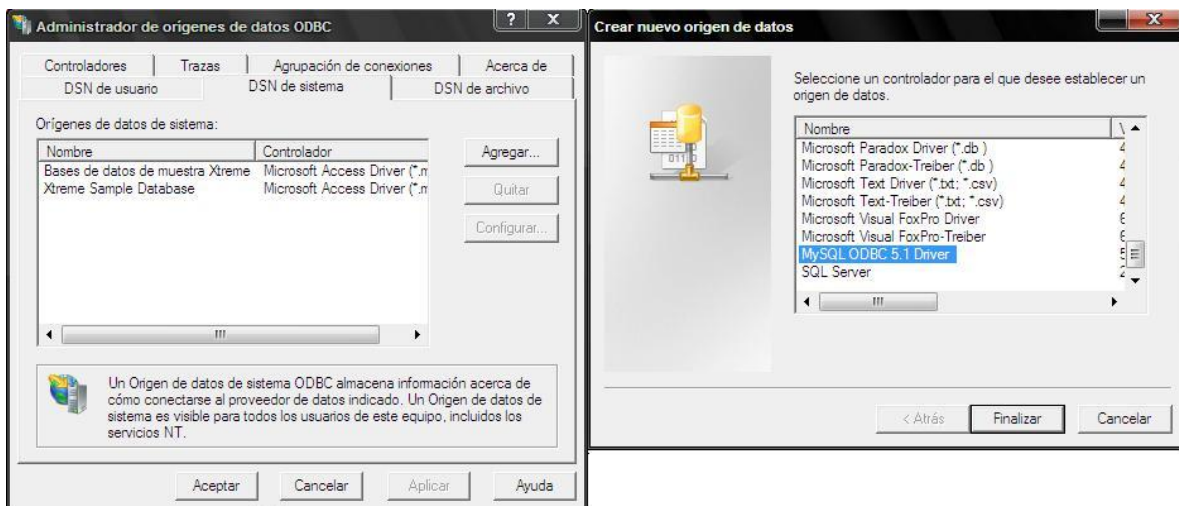


Fig. 14: Agregar un DSN (Origen de Datos) de Sistema.

5. A continuación se mostrará un formulario para configurar la conexión del driver ODBC elegido (**Fig. 15**), donde se ingresa un nombre para el DSN de sistema que está siendo creado, esto para que pueda ser identificado en el sistema por cualquier aplicación capaz de trabajar con orígenes de datos ODBC. Además se debe ingresar una descripción para el DSN (opcional, sirve para mayor información al usuario), el nombre o dirección IP del servidor (localhost o 127.0.0.1 en este caso por estar en la máquina local), el puerto de conexión con el servidor MySQL de destino (por defecto es el puerto 3306), el nombre de usuario (root), el password de acceso a la base de datos que se definió cuando se instaló el servidor MySQL 5.0 y el nombre de la base de datos (esquema) creado con el SGBD “MySQL Administrator versión 1.2.12”.

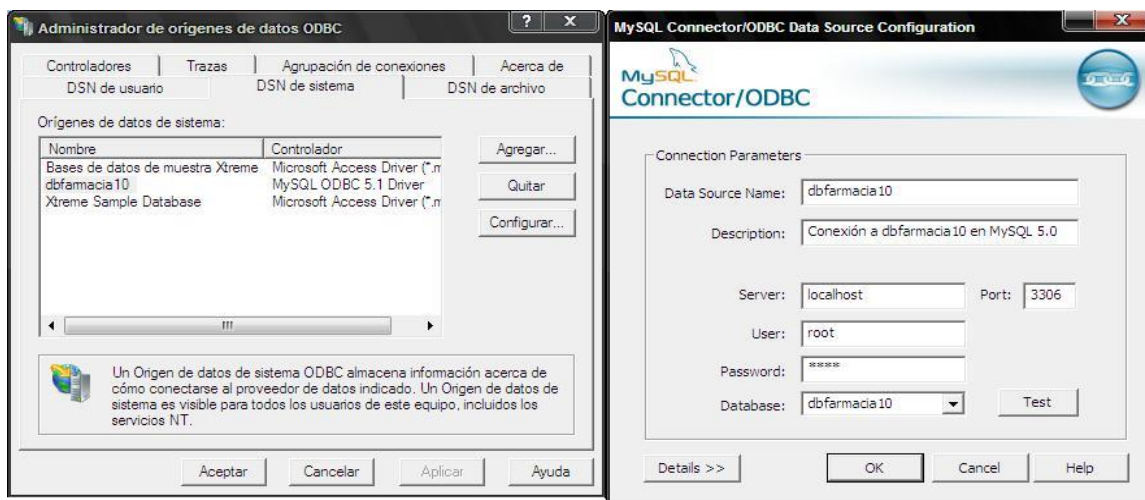


Fig. 15: Creación de DSN de Sistema con MySQL ODBC Driver.

6. Una vez completo el formulario con los parámetros requeridos, se puede probar si la conexión que se está creando funciona haciendo click en el botón “Test”, el que dará el mensaje confirmando la conexión exitosa. Luego se debe presionar “OK” en el formulario y “Aceptar” en la ventana del “Administrador de Orígenes de Datos”, para dejar creado y grabado el DSN de Sistema y posteriormente poder utilizarlo.

Ya creado el DSN de sistema, se está en condiciones de exportar los datos, utilizando el SGBD Access 97. Esto se realiza de la siguiente manera:

- a) Se abre el archivo de base de datos DBSFARM.mdb con el SGBD Access 97.
- b) Desde la pestaña “Tablas” se pueden ver los nombres de todas las tablas que conforman la base de datos. Haciendo un click sobre una tabla en particular se la puede seleccionar.
- c) Una vez seleccionada la tabla que se desea exportar, se debe ir al menú “Archivo”.
- d) Seleccionar "Guardar como o exportar...".
- e) En este paso elegir "En un archivo o una base de datos externos".
- f) En este paso la ubicación del archivo no se indica, pues está dada en el DSN de sistema que se creó previamente, el cual indica que la ubicación es la base de datos de destino “dbfarmacia10”. El DSN se indicará en un próximo paso.
- g) En el parámetro "Nombre de Archivo" el nombre de la tabla se deja sin cambios para evitar confusiones posteriores. (Este parámetro "Nombre de archivo" vuelve a ser requerido dos pasos después).
- h) En el parámetro "Guardar como tipo" se indica "Bases de datos ODBC()".
- i) En el cuadro de diálogo que aparecerá a continuación, se indica el nombre que tendrá la tabla en la base de datos de destino. También aquí se deja el nombre sin cambios para evitar confusiones posteriores, y se hace click en "Aceptar".
- j) Seguidamente aparece el cuadro de diálogo "Seleccionar origen de datos", donde se debe elegir el DSN creado antes, el cual usa el driver ODBC y establece la conexión con el servidor de datos MySQL y la base de datos "dbfarmacia10". Lo anterior se realiza haciendo un click en la pestaña "Origen de datos de equipo" y haciendo otro click sobre el nombre del DSN creado para

el efecto, llamado igualmente "dbfarmacia10". Finalmente para ejecutar la exportación se hace click en “Aceptar” y comenzará el traspaso de datos.

Es importante tener en cuenta que en la base de datos de destino, no debe existir previamente ninguna tabla con el mismo *nombre de la tabla* que se está exportando, es decir, si se va a exportar "paciente", no debe existir previamente ninguna tabla que se llame “paciente” en la base de datos de destino, así se evitará un fallo en la operación.

Los pasos para exportar, se deben realizar para cada tabla de la base de datos origen (DBSFARM.mdb), una a una, hasta completarlas todas. De esta manera se exportan todos los datos y se tienen ahora en la base de datos “dbfarmacia10”, conservando la misma estructura de atributos, de las tablas originales y los mismos tipos de datos, salvo en el caso de los atributos de tipo “fecha” de Access 97, que son convertidos automáticamente a tipo “string”, por lo que se deben cambiar nuevamente a tipo “Date” de MySQL, con el SGBD “MySQL Administrator”, editando la estructura de la tabla en cuestión.

5.2 Mapeo de Datos entre Bases de Datos dbfarmacia 1.0 y dbfarmacia 2.0.

Una vez que se exportaron los datos desde el formato en que estaba “DBSFARM.mdb” al formato MySQL 5.0 con el nombre de “dbfarmacia10”, se está en condiciones de realizar el mapeo de los datos requeridos desde cada tabla hacia su correspondiente en la base de datos “dbfarmacia20” del nuevo “Sistema de Farmacia 2.0”. Estas actividades forman parte de la segunda etapa de la estrategia de migración de los sistemas planteada en el capítulo 2.

El mapeo comienza con el análisis de las tablas de ambas bases de datos, para identificar las *tablas de origen y destino* que almacenan la misma información o semejante. Además aquí se debe ver los *atributos coincidentes de origen y destino*, donde se moverán los datos, y los *atributos nuevos*, que previamente no existen en “dbfarmacia10”, los cuales se completarán en otra etapa del proceso, en forma automática, con valores por defecto. Estas tareas también forman parte del análisis de la aplicación que se desarrolla para migrar los datos en el punto 5.3.

La Fig. 16: Mapeo de tablas para migración desde "dbfarmacia10" a "dbfarmacia20"., muestra las tablas que contienen ambas bases de datos y la correspondencia de cada cual para la migración de los datos:

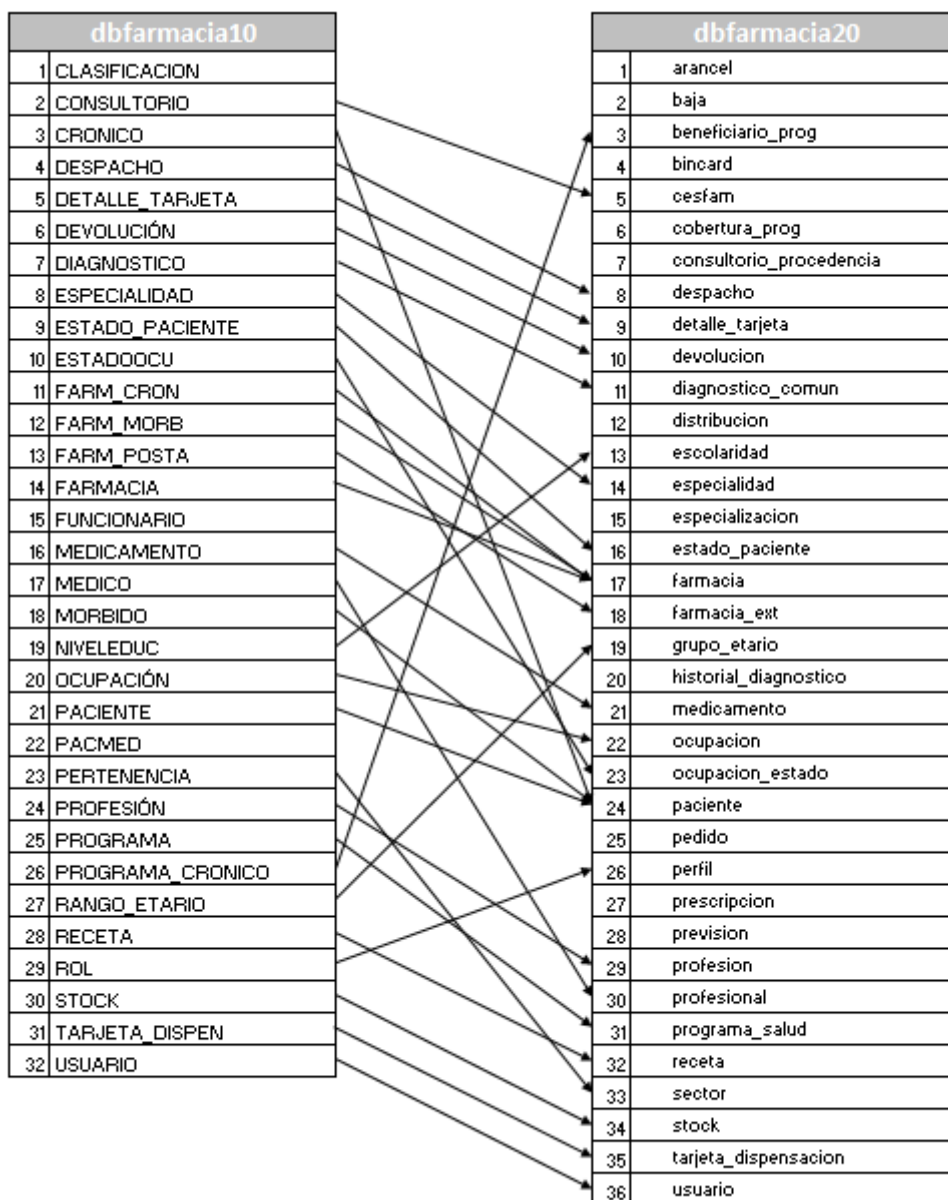


Fig. 16: Mapeo de tablas para migración desde "dbfarmacia10" a "dbfarmacia20".

Las tablas de "dbfarmacia10", "CLASIFICACION" y "FUNCIONARIO", que no indican mapeo hacia alguna tabla en "dbfarmacia20", son tablas cuya información no se utilizará en el nuevo sistema, debido a que son datos duplicados

o datos utilizados antes, en el “Sistema heredado de Farmacia 1.0”, para una funcionalidad de administración de perfiles de usuario que ha sido reemplazada en el nuevo “Sistema de Farmacia 2.0”.

5.3 Análisis, Diseño y Pruebas de la Aplicación para Migración, Chequeo de Integridad y Consistencia de Datos de “dbfarmacia10” y “dbfarmacia20” (Migración-FF).

En este punto se desarrolla el análisis y diseño de la aplicación que se ha denominado “Migración-FF” (entre bases de datos de farmacia) para continuar ejecutando la segunda etapa de la estrategia de migración de datos desde el “Sistema de Farmacia 1.0” al “Sistema de Farmacia 2.0”.

5.3.1 Análisis de la Aplicación “Migración-FF”.

Para la etapa de análisis de la aplicación “Migración-FF”, ha sido necesario el análisis previo de los datos heredados de la base de datos “dbfarmacia10”, tal como se plantea en el estudio de migración de los SIH visto en el marco teórico para conocer los datos y sus relaciones, además de aplicarles preprocesamiento, lo que se logró con herramientas como planillas de cálculo y consultas de Access 97 donde se buscaron, como ya se dijo en el punto 5.1, las siguientes situaciones: valores nulos, inválidos, o fuera de rango en campos claves y no claves; registros faltantes entre tablas que tienen restricciones de integridad en ciertos campos (ej.: si un “Num_Ficha” está en la tabla “tarjeta_dispen”, entonces también debe estar en la tabla “pacientes”); patrones de asociación entre valores de campos distintos relacionados por las reglas de negocio para descubrir valores erróneos o situaciones excepcionales de uso del sistema. Esto último podría considerarse como una especie de minería de datos, para descubrir la calidad de los datos antes de migrarlos.

Es importante mostrar la relación de esta aplicación, “Migración-FF”, con las estrategias de migración vistas en el marco teórico. La aplicación de migración de datos “Migración-FF”, toma un elemento de la estrategia “Butterfly”, en la que se tiene una etapa destinada a crear un “*Data Transformer*” o “Transformador de

Datos”, la aplicación “Chrysaliser”, especializada para mapear los datos desde la base de datos heredada (con datos heredados) pasándolos a almacenes temporales con formato de la base de datos del nuevo sistema de información, para luego aplicarles reglas de transformación que los llevan a tener la misma estructura conceptual que el sistema heredado, pero ahora bajo el modelo de datos del sistema nuevo. Análogamente “Migración-FF” toma los datos heredados almacenados temporalmente en “dbfarmacia10” con el formato nuevo MySQL 5.0, pero aún bajo el modelo de datos heredado, los transforma y mapea, llevándolos a “dbfarmacia20” con el mismo nuevo formato MySQL 5.0 y ahora bajo el modelo de datos del “Sistema de Farmacia 2.0”.

La problemática de la migración de datos desde “dbfarmacia10” a “dbfarmacia20” queda en parte explicada por el diagrama anterior de la Fig. 16, donde se muestra la similitud entre ambas bases de datos y como se mueven los datos entre tablas.

Adicionalmente es necesario precisar por cada tabla, como se moverán los datos entre atributos de tablas de ambas bases de datos. Esto dará origen a las reglas de transformación que se aplicarán a los datos, y si es preciso cambiar los tipos de datos, por ejemplo de tipo “cadena” a tipo “fecha”.

El orden en que se ejecutará la migración de los datos también es importante, ya que cada tabla tiene restricciones de integridad, que obligan a que los datos de un atributo de una determinada tabla existan previamente en otra. De esta manera se deben migrar primero algunas tablas y otras posteriormente cuando aquellas de las que dependen están completas, por ejemplo migrar la tabla “paciente” antes que la tabla “tarjeta_dispen” y migrar la tabla “medicamentos” antes que la tablas “despacho” y “stock” que hacen referencia a “medicamentos”.

Es del caso también hacer mención al hecho de que la migración de datos se hará en la mayoría de los casos de las tablas, en forma automatizada por la aplicación en análisis, sin embargo existen tablas que poseen muy pocos registros dado que definen algunos parámetros como por ejemplo la tabla “ESTADOOUCU”

que define los estados ocupacionales de un paciente, que son 8 registros (activo, jubilado, cesante, etc.); en tales casos la migración no se hará por la aplicación sino que en forma manual, dado que es mucho más sencillo para el logro del objetivo.

A continuación se presentan los requerimientos funcionales para la aplicación:

5.3.1.1 Requerimientos Funcionales de la Aplicación.

Ref.	Función.
RF1	Migrar datos desde las tablas de la base de datos “dbfarmacia10” a las tablas correspondientes en la base de datos “dbfarmacia20”, de acuerdo al mapeo establecido en el análisis de ambos modelos de datos y el estudio de los sistemas de farmacia 1.0 y 2.0.
RF2	Mapear el traspaso de los datos desde cada atributo de las tablas en “dbfarmacia10” al correspondiente atributo y tabla de “dbfarmacia20”.
RF3	Establecer el orden en que se migran las tablas de acuerdo a las restricciones de integridad de la base de datos “dbfarmacia20”.
RF4	Mostrar al usuario, por cada tabla migrada, la cantidad de registros leídos en “dbfarmacia10”, la cantidad de registros migrados a “dbfarmacia20”.

5.3.1.2 Requerimientos No Funcionales de la Aplicación.

Atributos del Sistema	Detalles y Restricciones
Facilidad de uso	El usuario puede indicar el inicio de la ejecución de la migración de las tablas y esperar el resultado de la operación.

Tiempo de Respuesta.	La migración consiste en la consulta y escritura de aproximadamente 650.000 registros en 5 a 7 minutos.
Soporte Sistema Operativo.	Microsoft Windows XP Profesional o Linux (Kernel v2.6).
Soporte Servidor de Base de Datos	MySQL Server Versión 5.0.24a Community
Soporte Máquina Virtual JAVA.	JAVA J2SE Versión 5 ó 6.
Driver JAVA para Base Datos MySQL	Mysql-connector-java-5.0.3
Metáfora de Interfaz del Sistema.	Interfaz Web con gráficos.

5.3.2 Diseño de la Aplicación “Migración-FF”.

La aplicación “Migración-FF” utiliza como base los modelos de datos de los sistemas involucrados en la migración. Además dichos modelos serán accedidos por la aplicación utilizando la arquitectura MVC y los patrones de diseño DAO, Abstract Factory y Data Transfer Object. Además desde el punto de vista de la arquitectura del sistema, la aplicación “Migración-FF” puede clasificarse como “sistema de procesamiento de datos”. “Las aplicaciones de procesamiento de datos son aplicaciones conducidas por los datos. Procesan datos por lotes sin intervenciones explícitas del usuario durante el procesamiento. Las acciones específicas tomadas por la aplicación dependen de los datos que se están procesando. Los sistemas de procesamiento por lotes se usan normalmente en aplicaciones de negocio en donde se realizan operaciones similares sobre grandes cantidades de datos”. [Sommerville 2005]. Por este motivo la aplicación no está diseñada para interactuar mayormente con el usuario y sólo requiere su intervención para llamarla desde el menú del “Sistema de Farmacia 2.0”, indicarle que tablas se desea migrar y luego dar la orden de comienzo de la ejecución de la migración.

5.3.2.1 Modelo de Datos

Un modelo de datos para la aplicación “Migración-FF” está dado por la relación que enlaza los modelos de datos del “Sistema Heredado de Farmacia 1.0” y “Sistema de Farmacia 2.0”, que están descritos con anterioridad en el capítulo 4 y en la Fig. 16 del capítulo 5, sobre el mapeo de datos entre tablas de ambas bases de datos. Esencialmente cada tabla en “dbfarmacia10” tiene su tabla destino en “dbfarmacia20”, salvo algunos casos que ya fueron descritos en el análisis de la aplicación.

La correspondencia entre tablas incluye el mapeo de atributos, para transformar desde un modelo de datos a otro, por esta razón ha de realizarse para cada tabla de “dbfarmacia10” que será migrada. En la Fig. 17 se muestran como ejemplo las tablas detalle_tarjeta y su correspondencia de atributos.

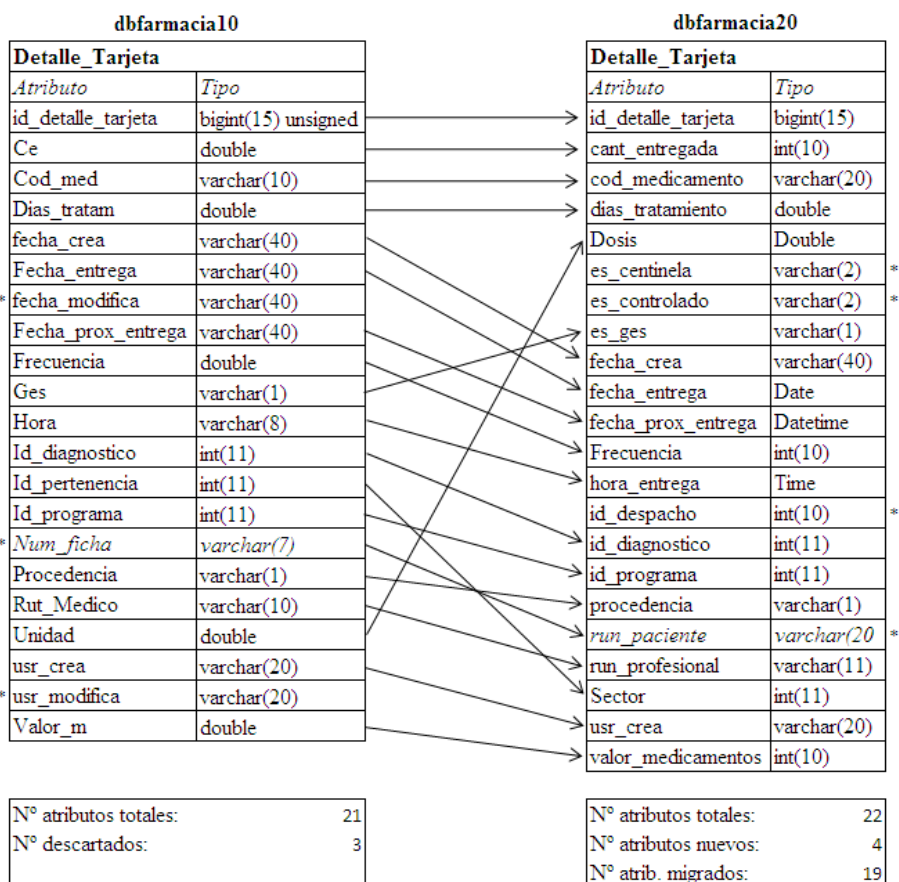


Fig. 17: Mapeo de atributos entre tabla “detalle_tarjeta” para migración.

En la figura anterior (ver Fig. 17), se destacan con asteriscos al lado izquierdo, los atributos de las tablas que son descartados en la migración desde la base de datos legada (dbfarmacia10) a la nueva (dbfarmacia20) por no ser utilizados en el nuevo modelo, y por otra parte se marcaron con asterisco al lado derecho, los atributos nuevos en el modelo de la nueva base de datos (dbfarmacia20). Estos atributos son: es_centinela, es_controlado e id_despacho. La forma en que se asignará el valor a estos atributos en el caso de la tabla detalle_tarjeta, es el siguiente:

Atributo	Valor asignado inicialmente
es_centinela	NULL
es_controlado	NULL
id_despacho	Valor numérico autoincremental por cada registro.

Tabla 7: Valores asignados por defecto en atributos nuevos de dbfarmacia20.

Cabe señalar que en la Tabla 7 se les asigna valor nulo a algunos de los atributos porque por acuerdo con el usuario, no se exige dichos valores a los registros heredados y sólo se asigna un valor cuando sea necesario para los nuevos registros que se vayan creando, cuando el sistema está en funcionamiento.

Un desafío al migrar la tabla “detalle_tarjeta” es con respecto al atributo “Num_ficha” en dbfarmacia10, porque es el único dato del que se dispone para identificar a que paciente pertenece el registro, por lo que en el momento de realizar el mapeo es necesario que este atributo ya haya sido preprocesado y contenga el run del paciente, así se evita tener que realizar la búsqueda del run correspondiente en la tabla “paciente” de dbfarmacia10 por cada registro que se quiere migrar y así extender demasiado el tiempo de ejecución del mapeo. Como necesariamente el preproceso de esta tabla debe realizarse, se prepara un método que lo lleve a cabo y se ejecuta dicho método antes de realizar la migración.

El preproceso de la tabla “detalle_tarjeta” de “dbfarmacia10”, para insertar el Rut de los pacientes en cada registro en lugar de los números de ficha se realiza siguiendo los siguientes pasos:

- a) Se agrega el atributo “run_paciente” a la tabla “detalle_tarjeta”, usando el editor de bases de datos del SGBD “MySQL 5.0”, para posteriormente insertar el valor de Rut que corresponda desde la tabla “paciente”.
- b) Por cada registro en la tabla “paciente” se recorre la tabla “detalle_tarjeta” buscando por “Num_ficha” e insertando el valor del atributo “Rut” cuando corresponda. El diseño de este método en pseudocódigo se muestra a continuación en la Fig. 18:

```

1 INICIO
2 MIENTRAS ( paciente !EOF ) HACER{
3
4     LEER registro_paciente
5
6     SI( registro_paciente.runpaciente != NULL )ENTONCES{
7
8         encontrado =FALSE
9         distinto = FALSE
10
11         ABRIR( detalle_tarjeta )
12         MIENTRAS ( detalle_tarjeta !EOF AND !distinto) HACER{
13
14             LEER registro_detalle_tarj
15
16             SI (encontrado) ENTONCES
17                 encontrado = FALSE
18                 SI (numficha_anterior != registro_detalle_tarj.numficha) ENTONCES
19                     distinto = TRUE
20
21             SI ( registro_paciente.numficha == registro_detalle_tarj.numficha ) ENTONCES{
22
23                 encontrado = TRUE
24                 numficha_anterior = registro_detalle_tarj.numficha
25                 registro_detalle_tarj.runpaciente = registro_paciente.runpaciente
26
27             }
28         }
29         CERRAR( detalle_tarjeta )
30     }
31 }
32 FIN
33

```

Fig. 18: Diseño en pseudocódigo de método para preprocesar tabla "detalle_tarjeta"

Otra muestra de cómo se realiza el mapeo entre tablas se visualiza en la Fig. 19, donde el atributo “cant_max_med” de la tabla medicamento de “dbfarmacia10” es mapeado hacia dos atributos. Esta situación está dada por una regla de negocio, ya que es necesario establecer un valor máximo para la cantidad de medicamento que se entrega en una transacción, para evitar errores de digitación y para normar la dispensación. La doble migración es debido a que en la tabla “stock” de destino se necesita un *valor de referencia* para las farmacias internas, el que pueden modificar posteriormente y tener sus propias cantidades máximas de entrega para un determinado medicamento; y en la tabla medicamento este atributo cumple la misma función pero sólo para la farmacia general.

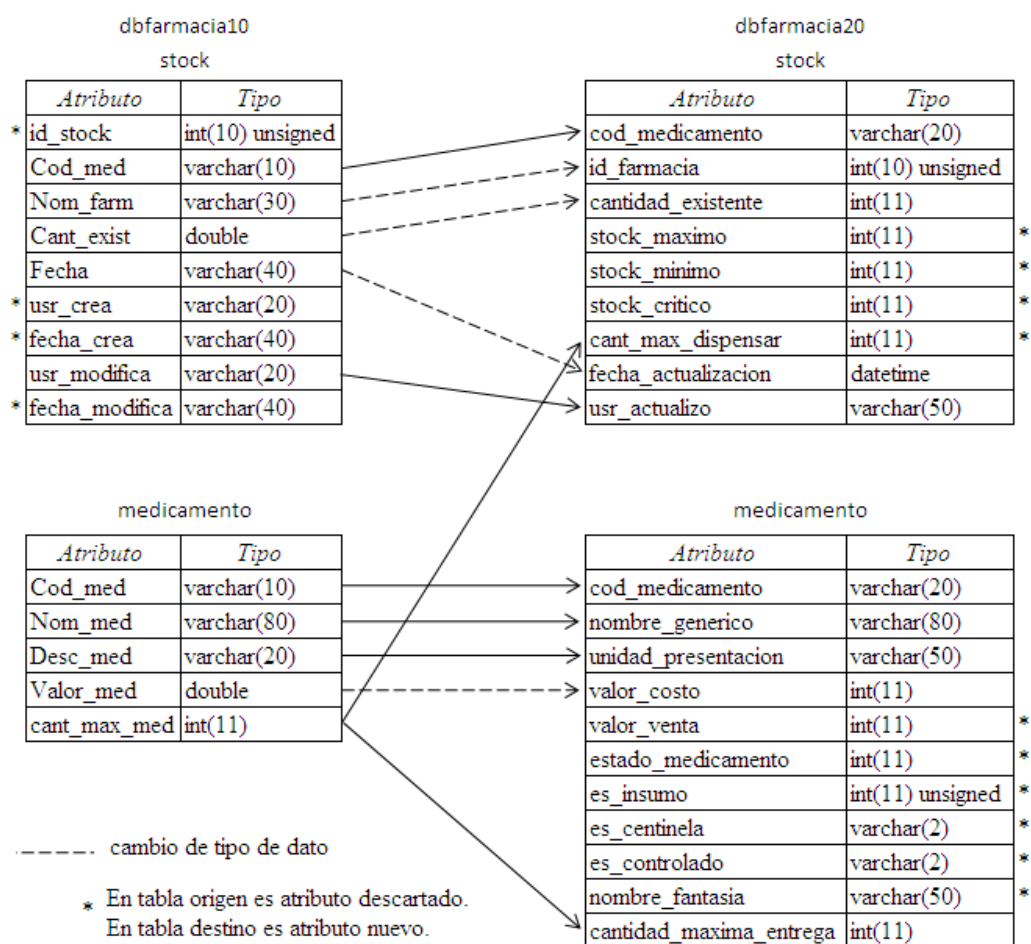


Fig. 19: Mapeo de atributos entre tablas “stock” y “medicamento” para migración

5.3.2.2 Diagrama de Clases.

En el diagrama de clases de la Fig. 20: Diagrama de clases para *tablas* “paciente”, *módulo* de aplicación Migración-FF., se presentan también las clases del framework Struts (Controlador y Vista) y las clases asociadas a la capa de persistencia (DAO) para tener una clara idea de cómo se relacionan e interactúan. Por simplicidad el diagrama muestra sólo las clases asociadas a la migración de la tabla “pacientes”, pero se debe considerar que de manera análoga es el diseño de las clases asociadas a la migración de las otras tablas de la base de datos.

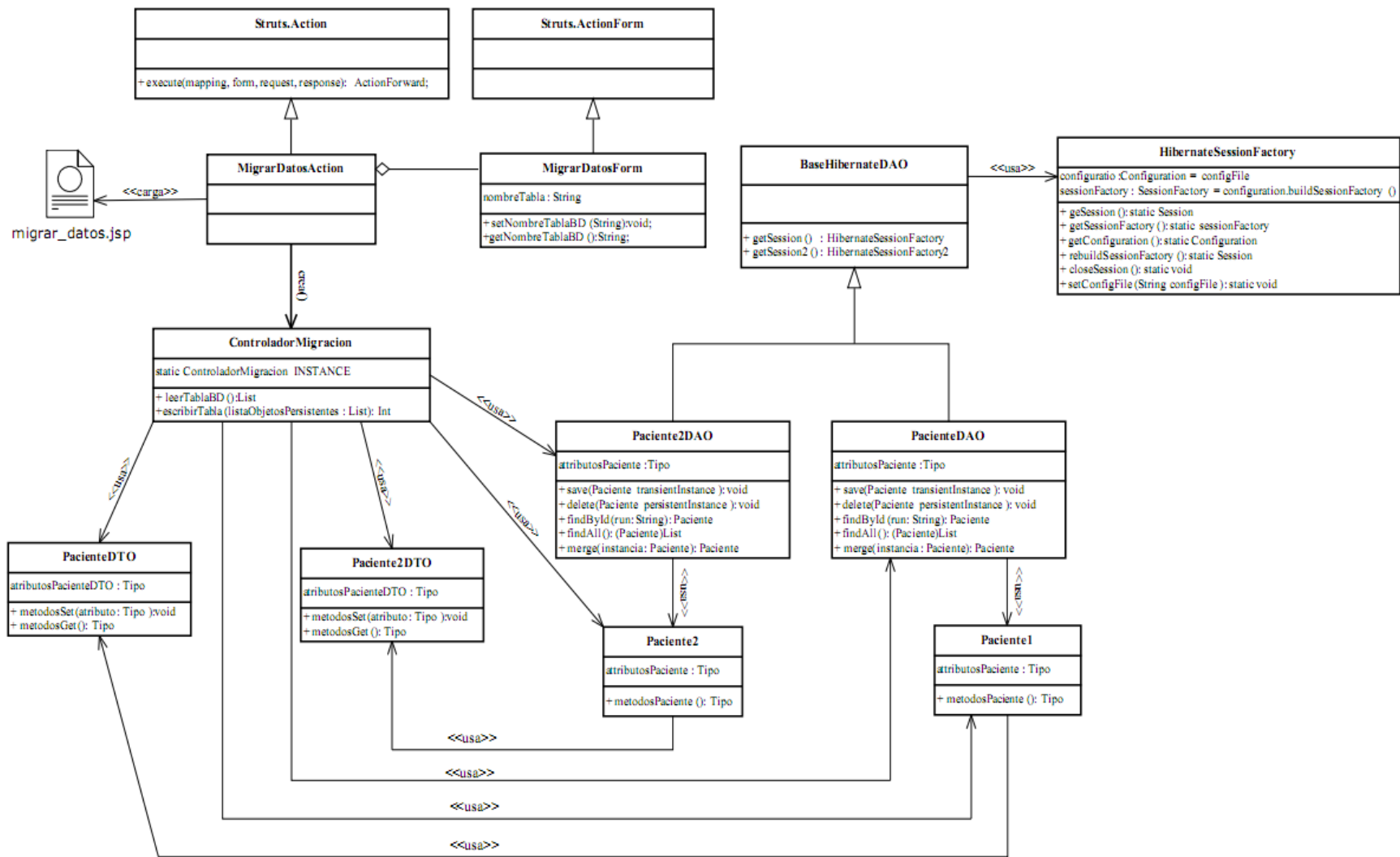


Fig. 20: Diagrama de clases para tablas “paciente”, módulo de aplicación Migración-FF.

5.3.2.3 Diagrama de Colaboración

Una colaboración es una descripción de una colección de objetos que interactúan para implementar un cierto comportamiento dentro de un contexto. Describe una sociedad de objetos cooperantes unidos para realizar un cierto propósito. [Rumbaugh *et al.* 2007].

Un diagrama de colaboración muestra la forma en que los objetos colaboran entre sí. Muestra los objetos junto con los mensajes que se envían entre ellos. Los diagramas de colaboración son semánticamente equivalentes a los diagramas de secuencia, con la diferencia que los de colaboración ponen énfasis en mostrar el contexto de los objetos y su organización general, es decir, de acuerdo al espacio y los diagramas de secuencia destacan la sucesión en que ocurren las interacciones entre objetos.

Para presentar los diagramas de colaboración se ha pensado conveniente hacer algunos alcances, ya hechos en un trabajo similar por González, [González, R. 2006], que ayudan a su mejor comprensión.

En el diagrama de colaboración que se presenta más adelante se incluyen los principales objetos del framework “Struts” organizados en una capa. En la capa Struts de cada diagrama el controlador *Struts*, visto como caja negra, crea una instancia de una clase “*Action*” la cual redirigirá las acciones de acuerdo a lo recibido en las peticiones a Struts. Las instancias de las clases “*Action*” se identifican por tener el sufijo del mismo nombre. (Ejemplo.: *MigrarAction*). De igual manera, las instancias de las clases de tipo “*Form*” creadas por Struts en los cuales se carga la información recibida en los formularios se identifican por el sufijo del mismo nombre. (Ejemplo.: *MigrarForm*). Los diagramas de colaboración se inician por una petición realizada por el usuario a través un formulario JSP y finalizan con una respuesta que es recibida también

en un formulario JSP. Los nombres asignados a las clases *Action* y *Form* se relacionan con la función que cumplen y no representan métodos.

Los objetos presentados en la capa *Struts* interactúan con servicios propios de ese framework y se representan con nombres sencillos para efectos de otorgar un mayor entendimiento para su interpretación. Estos métodos son: *enviarFormulario*, *cargar*, *redirige*, *cargarFormulario* y *resultado*. La descripción de cada método se explica a continuación:

a) *enviarFormulario*: Este método representa el envío de una petición desde el cliente (programa navegador Web) al controlador Struts.

b) *cargar*: Este método representa la acción que realiza el controlador Struts, de cargar en el navegador Web del cliente una determinada página.

c) *redirige*: Este método representa la acción de parte del controlador Struts de invocar al respectivo objeto *Action* para que procese la petición recibida de parte del cliente. En algunos diagramas de colaboración este método puede incluir algún parámetro. La incorporación de parámetros es para representar que el controlador Struts necesita que el objeto *Action* procese la petición con el parámetro enviado.

d) *cargarFormulario*: Método que representa la acción de parte del controlador Struts, de cargar los datos recibidos del formulario de la petición Web realizada por el cliente en un respectivo objeto Form.

e) *resultado*: Flujo que representa el envío de un cierto resultado al controlador Struts desde la capa de objetos de negocio, para que éste decida que respuesta entregar.

5.3.3 Pruebas de la Aplicación “Migración-FF”.

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación [Pressman 2002].

Al nivel del sistema o de validación, los detalles de conexión de clases desaparecen. Así como la prueba de validación convencional, la validación del software orientado a objetos se centra en las acciones visibles al usuario y salidas reconocibles desde el sistema. Los métodos convencionales de prueba de caja negra, pueden usarse para realizar pruebas de validación.[Pressman 2002].

Por la naturaleza de la aplicación "Migración-FF", que es orientada a los datos, los resultados sólo se pueden verificar realizando pruebas de validación con muestras de datos desde la base de datos origen (dbfarmacia10), que contengan un número limitado de registros (se puede controlar a través de condiciones en el código fuente del método “`mapearPacientes()`” de la clase “ControladorMigracion”), para poder verificar su migración por medios manuales, a través de revisión directa en la base de datos con el SGBD “MySQL Administrator” o con el SGBD “MySQL-Front”, por cada tabla por separado. De esta forma se ve si se ha llevado a efecto la migración de todos los registros correctamente.

Al probar la ejecución de "Migración-FF" con un número de registros pequeño o del orden de decenas de miles se puede visualizar en la interfaz del usuario del sistema, la cantidad de registros leídos por la aplicación en la base de datos origen “dbfarmacia10” y después de que la aplicación los ha migrado se puede ver además en ésta misma interfaz de usuario, la cantidad de registros insertados en la tabla correspondiente en la base de datos “dbfamacia20”.

A continuación se muestra algunos de los registros de pruebas realizadas. La plantilla de descripción está estructurada de acuerdo a recomendaciones de Pressman. [Pressman 2002]

Caso de Prueba:	Migración de Tabla "despacho"
<p>1. Propósito de la Prueba:</p> <p>Verificar que la aplicación lee los 10 primeros registros de la tabla original “despacho” de “dbfarmacia10” y los guarda en la tabla “despacho” de la base de datos "dbfarmacia20".</p>	
<p>2. Estado Inicial:</p> <p>La tabla “despacho” de “dbfarmacia10” contiene 28.832 registros y la tabla “despacho” de “dbfarmacia20” no contiene ningún registro.</p>	
<p>3. Estado Final Esperado:</p> <p>La tabla “despacho” de dbfarmacia20 contiene los 10 primeros registros de la tabla "despacho" de "dbfarmacia10".</p>	
<p>4. Mensajes en pantalla a visualizar:</p> <p>En caso de concretarse correctamente la migración de los registros, debe aparecer en pantalla, el número “10”, bajo la columna “dbfarmacia20”, frente al nombre de la tabla “despacho” bajo la columna “Regs. Migrados”; y también debe aparecer el número “10” frente al nombre de la tabla “despacho” de la base de datos “dbfarmacia20”, bajo la columna “Regs. Leídos”</p>	
<p>5. Posibles Excepciones durante la ejecución:</p> <ul style="list-style-type: none"> a) No se puede conectar a la base de datos. b) Registro no se puede insertar porque la clave principal está duplicada. c) No se puede insertar registro porque la clave del registro a insertar no existe en la tabla padre (Restricción de integridad). 	

6. Condiciones Externas para Funcionamiento:

Servidor de Bases de Datos MySQL 5.0 funcionando correctamente, Servidor de Aplicaciones Apache Tomcat funcionando correctamente, Navegador Web cliente funcionando correctamente.

7. Resultados:

Los 10 registros fueron copiados a la tabla “despacho” de “dbfarmacia20”. Se visualizan físicamente en la base de datos “dbfarmacia20”, a través del SGBD MySQL-Front..

8. Secuencia de Prueba:

1. Se leyeron los 10 primeros registros de “despacho” en “dbfarmacia10”, a través del método `findAll()` de la clase `Despacho2DAO` y se cargaron en una lista de objetos `Despacho2DTO`. (*)
2. Se mapearon los 10 elementos de la lista anterior, transformándolos en objetos `DespachoDTO`. (*)
3. Se obtuvo la lista con elementos `DespachoDTO` y se insertó cada uno de ellos en la tabla “despacho” de la base de datos “dbfarmacia20” a través del método `save()` de la clase `DespachoDAO`. (*)(**)
4. Se enviaron a la interfaz del usuario (página `migrarDatos.jsp`) el número de registros leídos y el número de registros migrados (insertados) en la prueba.

(*) Se verificó con herramientas de depuración de entorno de desarrollo Eclipse, insertando puntos de detención en la ejecución y examinando los valores contenidos en las instancias de los objetos en ejecución.

(**) Se verificó la existencia de los registros insertados en la tabla “despacho” de

dbfarmacia20, ejecutando la consulta a la base de datos con el SGBD MySQL-Front.

Caso de Prueba:	Migración de Tabla "stock"
<p>1. Propósito de la Prueba:</p> <p>Verificar que la aplicación lee los 10 primeros registros de la tabla original “stock” de “dbfarmacia10” y los guarda en la tabla “stock” de la base de datos "dbfarmacia20".</p>	
<p>2. Estado Inicial:</p> <p>La tabla “stock” de “dbfarmacia10” contiene 594 registros y la tabla “stock” de “dbfarmacia20” no contiene ningún registro.</p>	
<p>3. Estado Final Esperado:</p> <p>La tabla “stock” de “dbfarmacia20” contiene los 10 primeros registros de la tabla "stock" de "dbfarmacia10".</p>	
<p>4. Mensajes en pantalla a visualizar:</p> <p>En caso de concretarse correctamente la migración de los registros, debe aparecer en pantalla, el número “10”, bajo la columna “dbfarmacia20”, frente al nombre de la tabla “stock” bajo la columna “Regs. Migrados”; y también debe aparecer el número “10” frente al nombre de la tabla “stock” de la base de datos “dbfarmacia20”, bajo la columna “Regs. Leídos”</p>	
<p>5. Posibles Excepciones durante la ejecución:</p> <p>a) No se puede conectar a la base de datos.</p> <p>b) Registro no se puede insertar porque la clave principal está duplicada.</p> <p>c) No se puede insertar registro porque la clave del registro a insertar no existe en la</p>	

<p>tabla padre (Restricción de integridad).</p>
<p>6. Condiciones Externas para Funcionamiento:</p> <p>Servidor de Bases de Datos MySQL 5.0 funcionando correctamente, Servidor de Aplicaciones Apache Tomcat funcionando correctamente, Navegador Web cliente funcionando correctamente.</p>
<p>7. Resultados:</p> <p>Los 10 registros fueron copiados a la tabla “stock” de “dbfarmacia20”. Se visualizan físicamente en la base de datos “dbfarmacia20”, a través del SGBD MySQL-Front.</p>
<p>8. Secuencia de Prueba:</p> <ol style="list-style-type: none"> 5. Se leyeron los 10 primeros registros de “stock” en “dbfarmacia10”, a través del método <code>findAll()</code> de la clase <code>Stock2DAO</code> y se cargaron en una lista de objetos <code>Stock2DTO</code>. (*) 6. Se mapearon los 10 elementos de la lista anterior, transformándolos en objetos <code>StockDTO</code>. (*) 7. Se obtuvo la lista con elementos <code>StockDTO</code> y se insertó cada uno de ellos en la tabla “stock” de la base de datos “dbfarmacia20” a través del método <code>save()</code> de la clase <code>StockDAO</code>. (**) 8. Se enviaron a la interfaz del usuario (página <code>migrarDatos.jsp</code>) el número de registros leídos y el número de registros migrados (insertados) en la prueba. <p>(*) Se verificó con herramientas de depuración de entorno de desarrollo Eclipse, insertando puntos de detención en la ejecución y examinando los valores contenidos en las instancias de los objetos en ejecución.</p>

(**) Se verificó la existencia de los registros insertados en la tabla “stock” de dbfarmacia20, ejecutando la consulta a la base de datos con el SGBD MySQL-Front.

Capítulo 6: Tareas de Importación de Datos desde “Plataforma Virtual SOME” (dbsome) para Chequeo de Integridad y Consistencia con Sistema Heredado.

En la tercera etapa del proceso de migración de datos del Sistema de Farmacia 1.0 a la versión 2.0, según la estrategia presentada en el capítulo 2 corresponde realizar la integración de todas las tablas de “dbfarmacia20” hacia “dbsome”. Lo anterior supone migrar el modelo de datos completo de “dbfarmacia20” y en el proceso identificar las tablas equivalentes de ambas bases de datos, realizar procesos de chequeo de integridad y consistencia de la información, búsqueda de registros faltantes en una u otra base de datos, complementar y enriquecer los datos, para finalmente de estas tablas equivalentes obtener tablas únicas que serán accedidas por los dos sistemas de información y residirán en “dbsome”, evitando la duplicación de datos. El primer paso de esta etapa será obtener una copia de la base de datos del sistema “Plataforma Virtual del CESFAM Violeta Parra”, en adelante también Sistema SOME, para realizar la integración de tablas y el cruzamiento de los datos.

6.1 Importación de Datos a MySQL 5.0.

Para realizar esta tarea de importación se obtiene una copia de la base de datos del Sistema SOME (archivo dbsome.sql) a través del SGBD “MySQL-Front” y con el mismo software dicha copia, se traspasa al servidor de bases de datos MySQL 5.0 de la máquina en que se está desarrollando la aplicación para realizar la tercera etapa de la migración.

El software “MySQL-Front 3.2” es un SGBD especialmente desarrollado para administrar bases de datos MySQL. Es mantenido por una comunidad de programadores, que ponen a disposición el software a través de su sitio web <http://www.mysqlfront.de>, para la adquisición comercial de licencias y a la vez brindar la posibilidad de ser usado y probado gratuitamente durante 60 días, con la totalidad de su funcionalidad, y de esta manera utilizarlo para las tareas de importación descritas.

La copia de la base de datos “dbsome” se obtiene a través de la función “exportar” de “MySQL-Front 3.2”, la que da como resultado un archivo de extensión “.sql” que contiene sentencias de creación de tablas e inserción de datos en lenguaje SQL. Este archivo copia de la base de datos es llevado al computador donde se están desarrollando los módulos de administración del Sistema de Farmacia 2.0 para importar los datos que se utilizarán en las tareas de puesta en marcha de dicho sistema.

6.2 Tablas Coincidentes de las Bases de Datos “dbfarmacia2.0” y “dbsome”.

En la primera etapa de la estrategia de migración “dbfarmacia10” fue obtenida por exportación de la base de datos original “DBSFARM.mdb a MySQL,

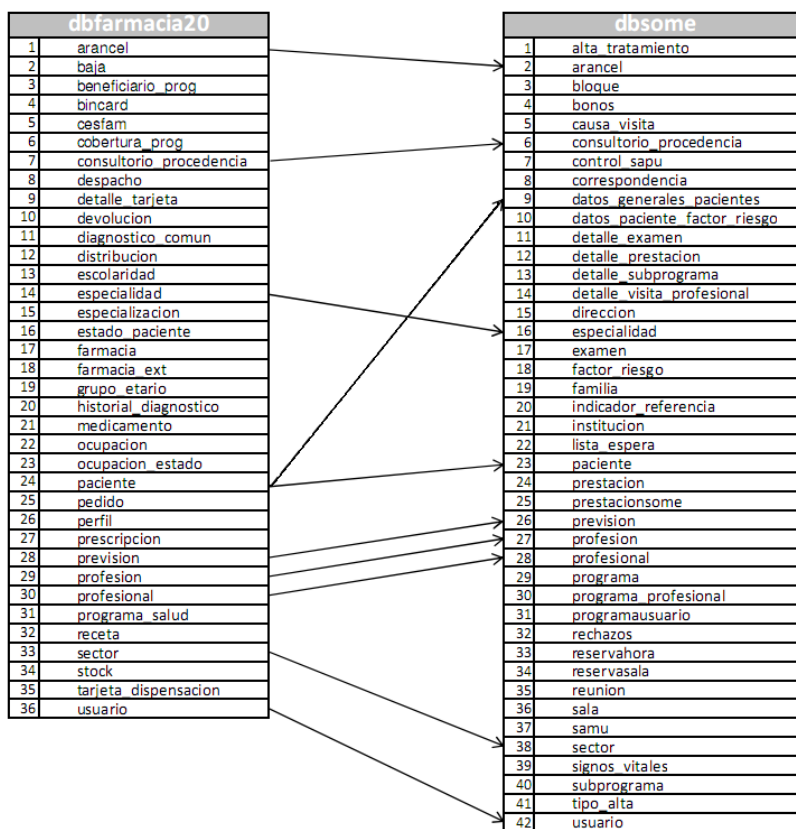


Fig. 22: Tablas equivalentes entre bases de datos "dbfarmacia20" (DBSFARM.mdb), y "dbsome".

luego en la segunda etapa se ha realizado el mapeo entre las tablas de “dbfarmacia10” y “dbfarmacia20” con la aplicación “Migración-FF”. A continuación se presenta el diagrama con la identificación de las tablas equivalentes.

6.3 Análisis, Diseño y Pruebas de la Aplicación para Migración, Chequeo de Integridad y Consistencia de Datos de “dbfarmacia20” y “dbsome” (“Migración-FS”).

En este punto se desarrolla el análisis y diseño y pruebas de la aplicación que se ha denominado “Migración-FS” (entre bases de datos de Farmacia y SOME) para ejecutar la tercera etapa de la migración de datos desde “dbfarmacia20” a “dbsome”.

6.3.1 Análisis de la Aplicación “Migración-FS”.

La aplicación “Migración-FS” tiene como objetivo automatizar parte de las tareas requeridas para la integración de la base de datos “dbfarmacia20” en la base de datos “dbsome”. Estas tareas se ven reflejadas en los requisitos de la aplicación que se presentan a continuación.

6.3.1.1 Requisitos Funcionales de la Aplicación.

Ref.	Función.
RF1	Migrar todas las tablas de la base de datos “dbfarmacia” a la base de datos “dbsome”, incluyendo los datos que contienen las tablas.
RF2	Para tablas que son equivalentes en ambas bases de datos, mapear el traspaso de los datos desde cada atributo de las tablas en “dbfarmacia20” al correspondiente atributo y tabla de “dbsome” decidiendo la edición de los registros de acuerdo a criterios establecidos en el análisis de las tablas.
RF3	Para tablas que no existen previamente en “dbsome” migrarlas y mantener las restricciones de integridad entre tablas.
RF4	Establecer el orden en que se migran las tablas de acuerdo a las restricciones de integridad de las bases de datos “dbfarmacia20” y “dbsome”.

RF5	Mostrar al usuario, por cada tabla migrada, la cantidad de registros leídos en “dbfarmacia20”, la cantidad de registros migrados a “dbsome”.
-----	--

6.3.1.2 Requisitos No Funcionales de la Aplicación.

Atributos del Sistema	Detalles y Restricciones
Facilidad de uso	El usuario puede indicar el inicio de la ejecución de la migración de las tablas y esperar el resultado de la operación.
Tiempo de Respuesta.	La migración consiste en la migración, consulta, aplicación de criterios de actualización y modificación a 10 tablas y aproximadamente 140.000 registros en 5 a 7 minutos.
Soporte Sistema Operativo.	Microsoft Windows XP Profesional o Linux (Kernel v2.6).
Soporte Servidor de Base de Datos	MySQL Server Versión 5.0.24a Community
Soporte Máquina Virtual JAVA.	JAVA J2SE Versión 5 ó 6.
Driver JAVA para Base Datos MySQL	Mysql-connector-java-5.0.3
Metáfora de Interfaz del Sistema.	Interfaz Web con gráficos.

6.3.2 Diseño de la Aplicación “Migración-FS”.

La aplicación “Migración-FS” también utiliza como base los modelos de datos de los sistemas involucrados en la migración. La aplicación utiliza la arquitectura MVC y los patrones de diseño DAO, Abstract Factory y Data Transfer Object. Desde el punto de vista de la arquitectura del sistema, la aplicación “Migración-FS” además puede clasificarse como “sistema de procesamiento de datos”, ya que realiza su cometido sin mayores intervenciones del usuario.

6.3.2.1 Modelo de Datos.

Anteriormente ya se ha mostrado el modelo de datos del “Sistema de Farmacia 2.0”, por lo que aquí se presenta el modelo de datos al cual se integrará la base de datos de dicho sistema. El modelo de datos del “Sistema SOME”, cuenta con 45 entidades, de las cuales 10 son equivalentes a alguna en el modelo de datos del “Sistema de Farmacia 2.0”.

De lo anterior y de los requisitos de la aplicación “Migración-FS”, se desprende la idea de que existen básicamente dos tipos de tareas a realizar en esta etapa de la migración y son a saber: la migración completa de tablas inexistentes en “dbsome” (exclusivas de farmacia) y el mapeo de tablas equivalentes (compartidas con “Sistema SOME”) que luego se integrarán en “dbsome”. Respecto de la conveniencia de integrar las tablas que son exclusivas del “Sistema de Farmacia 2.0” a la base de datos “dbsome”, se analizó y se decidió realizarlo dado que muchas de estas tablas exclusivas de “dbfarmacia20” poseen restricciones de integridad que las asocian con las tablas equivalentes que quedarán definitivamente en “dbsome” y conceptualmente si existe una restricción de integridad entre claves de distintas tablas, significa que ambas tablas forman parte del mismo modelo de datos. Las tablas equivalentes con el Sistema SOME son las siguientes: “arancel”,

“consultorio_procedencia”, “especialidad”, “paciente”, “previsión”, “profesión”, “profesional”, “programa”, “sector” y “usuario”.

A continuación se muestra el Diagrama Entidad Relación de la base de datos del “Sistema SOME” con las entidades más relevantes para el diseño de la migración y para una mejor comprensión del modelo. Se han subrayado las entidades que tienen equivalente en el “Sistema de Farmacia 2.0”.

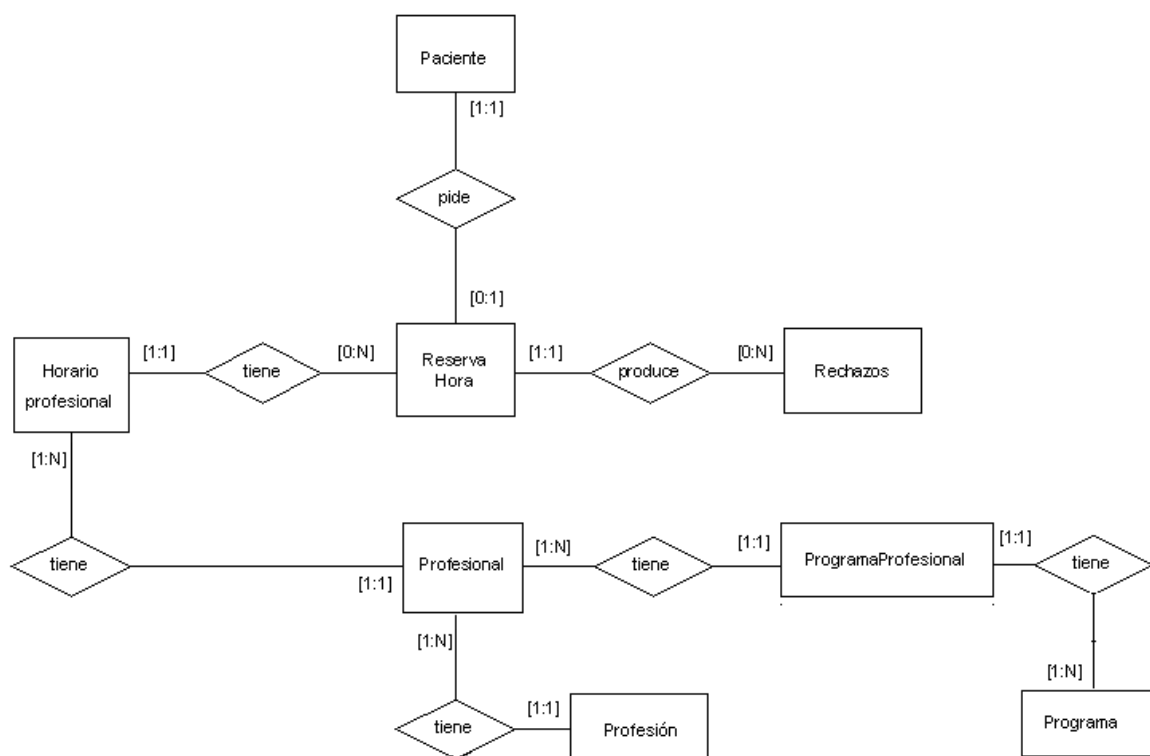


Fig. 23: Diagrama Entidad Relación simplificado del "Sistema SOME" y base de datos "dbsome".

En el punto 6.2, previo al análisis de la aplicación “Migración-FS” se muestra el diagrama con las tablas equivalentes entre las bases de datos “dbfarmacia20” y “dbsome” (Fig. 22). A partir de ese diagrama se tiene una visión general de cuáles son las tablas que han de migrarse completamente porque no existen en “dbsome” y cuáles han de mapearse y cruzar su información con sus equivalentes para decidir la inserción, modificación o descarte de datos.

El mapeo de atributos para las tablas equivalentes se realiza, al igual que en la aplicación “Migración-FF” implementando el patrón de diseño Data Access Object (DAO) y el patrón Abstract Factory. El patrón DAO hace transparente el acceso a las operaciones con la base de datos y el patrón Abstract Factory implementa una interfaz para acceder a las distintas familias de Objetos DAO según estén relacionados con “dbfarmacia” o “dbsome”.

La siguiente figura muestra como son mapeados los atributos de las tablas “paciente” y “datos_generales_paciente”, además para cada caso se indica el criterio que la aplicación usará para operar sobre los datos.

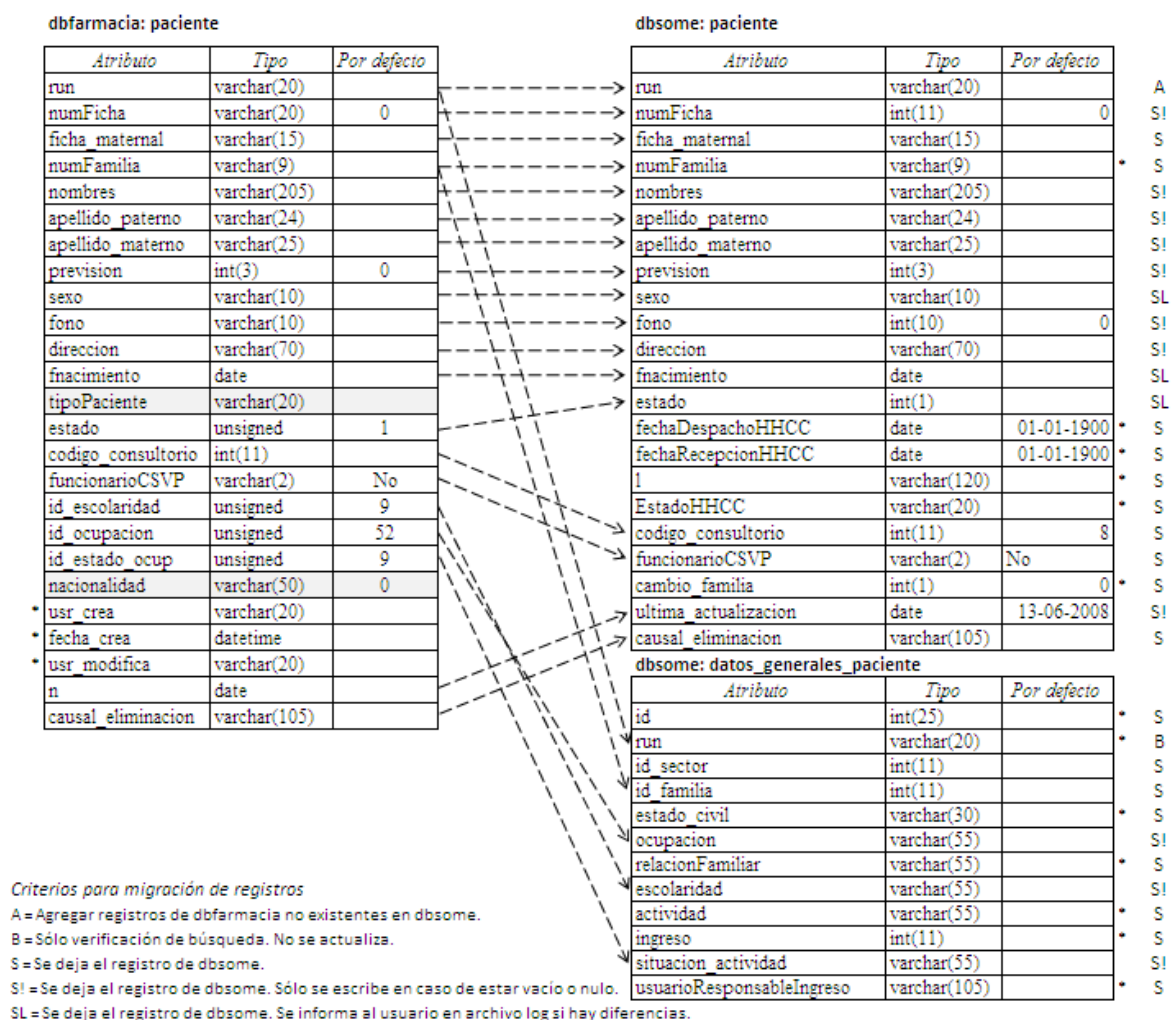


Fig. 24: Mapeo de Atributos y criterios para migrar datos desde "dbfarmacia" a "dbsome".

6.3.2.2 Diagrama de Clases.

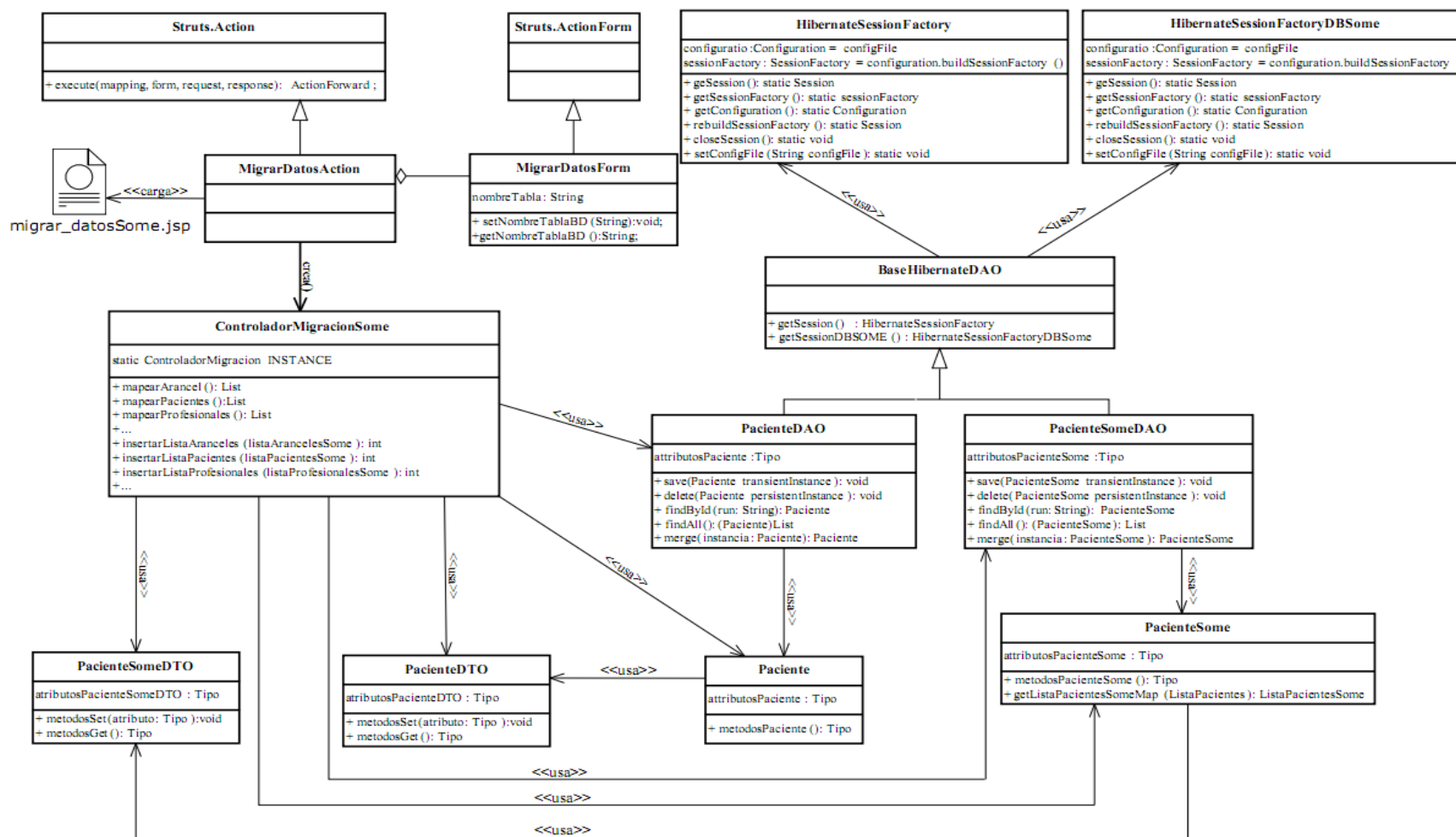


Fig. 25: Diagrama simplificado de Clases de Diseño módulo "Migración-FS"

6.3.3 Pruebas de la Aplicación “Migración-FS”.

Las pruebas de validación realizadas son con el objetivo de determinar si efectivamente se migran las tablas nuevas de “dbfarmacia20” al modelo de “dbsome”, y por otra parte realizar una prueba de migración con mapeo y chequeo de atributos de las tablas “paciente”, con una muestra de registros de la base de datos.

Caso de Prueba:	Migración de Tabla exclusiva de “dbfarmacia20”: "medicamento"
1. Propósito de la Prueba:	
<p>Verificar que la aplicación realiza una copia completa de la tabla “medicamento” desde “dbfarmacia20”.a “dbsome” donde no existe.</p>	
2. Estado Inicial:	
<p>La tabla “medicamento” de “dbfarmacia20” contiene 728 registros.</p> <p>En “dbsome” no existe ninguna tabla llamada medicamento.</p>	
3. Estado Final Esperado:	
<p>La tabla “medicamento” de “dbfarmacia20” ha sido copiada en “dbsome” y contiene los 728 registros de la tabla original.</p>	
4. Mensajes en pantalla a visualizar:	
<p>En caso de concretarse correctamente la migración de la tabla, debe aparecer en pantalla, el número de “728” registros migrados, frente al nombre de la tabla “medicamento”.</p>	
5. Posibles Excepciones durante la ejecución:	
<p>a) No se puede conectar a la base de datos “dbsome”.</p>	

6. Condiciones Externas para Funcionamiento:

Servidor de Bases de Datos MySQL 5.0 funcionando correctamente, Servidor de Aplicaciones Apache Tomcat funcionando correctamente, Navegador Web cliente funcionando correctamente.

7. Resultados:

La tabla con los 728 registros fue creada con éxito en “dbsome”. Se visualizan físicamente los registros en la base de datos “dbsome”, a través del SGBD MySQL-Front.

8. Secuencia de Prueba:

1. Se leyó la tabla “medicamentos” y se generó un archivo en disco con nombre “medicamento.sql” el que se graba en una ubicación predeterminada. (*)
2. El módulo de migración lee el archivo creado en el paso anterior y ejecuta las instrucciones automáticamente en la consola del sistema, enviándolas al servidor de bases de datos que crea la tabla “medicamento” en la base de datos “dbsome” e inserta los 728 registros. (**).
3. Se envía a la interfaz del usuario (página migrarDatosSome.jsp) el número de registros leídos en “dbfarmacia” y el número de registros migrados (insertados) en “dbsome”.

(*) Se verificó la existencia del archivo “medicamento.sql” directamente en el computador, a través del administrador de archivos.

(**) Se verificó la existencia de la tabla “medicamento” y los registros insertados en ella en la base de datos “dbsome”, ejecutando la consulta a la base de datos con el SGBD MySQL-Front.

Capítulo 7: Desarrollo de Módulo de Administración y Mantenimiento para Backup.

Si por alguna razón ocurre un problema con la base de datos del “Sistema de Farmacia 2.0” y se corrompen algunas o todas sus tablas, resultando en un daño de los datos, se estaría ante un grave problema, ya que el sistema de información no podría operar, se perderían datos valiosos de los pacientes, con las consecuencias desastrosas que esto tiene para las farmacias del CESFAM y las personas que solicitan atención.

Es por lo anterior que es necesario adoptar políticas y protocolos para el respaldo y recuperación de datos en las organizaciones. Algunas de estas prácticas ya están implementadas en el CESFAM, respecto del sistema “Plataforma Virtual del CESFAM Violeta Parra”, para el cual se realizan respaldos de la base de datos todos los días al final de la jornada, utilizando la funcionalidad de exportación a archivo “.sql” que posee el SGBD MySQL-Front.

En el presente capítulo se desarrolla un módulo para dotar al Sistema de Farmacia 2.0 de la funcionalidad de creación de archivos de respaldo o backup de la base de datos, necesarios en caso de fallas de la base de datos o pérdida de información.

7.1 Análisis de Requerimientos del Módulo Backup.

7.1.1 Requisitos Funcionales de la Aplicación.

Ref.	Función.
RF1	Crear un archivo de respaldo (backup) completo de todas las tablas que utiliza el “Sistema de Farmacia 2.0”.
RF2	El módulo debe dar la posibilidad de elegir la unidad y ruta en que se guardará el archivo de respaldo y tener asignada una ubicación por defecto en caso de que el usuario no la indique.

RF3	El nombre del archivo generado tendrá nombre estandarizado llevando la fecha, hora, minuto y segundo en que se crea el archivo.
RF4	El archivo backup creado, debe contener tanto la estructura como los datos de la base de datos en caso de se requiera crearla nuevamente.
RF5	Brindar la posibilidad de programar backups automáticos de la base de datos para que sean ejecutados en algún momento sin intervención directa del usuario.

7.1.2 Requerimientos No Funcionales de Módulo de Backup.

Atributos del Sistema	Detalles y Restricciones
Facilidad de uso	El usuario indica que desea realizar el backup desde el menú del sistema.
Tiempo de Respuesta.	Depende del tamaño de la base de datos. Como referencia para 650.000 registros el tiempo de respuesta es de 4 min. 20 segundos.
Soporte Sistema Operativo.	Microsoft Windows XP Profesional o Linux (Kernel v2.6).
Soporte Servidor de Base de Datos	MySQL Server Versión 5.0.24a Community
Soporte Máquina Virtual JAVA.	JAVA J2SE Versión 5 ó 6.
Driver JAVA para Base Datos MySQL	Mysql-connector-java-5.0.3
Metáfora de Interfaz del Sistema.	Interfaz Web con gráficos.

7.2 Diseño del Módulo Backup.

El módulo backup es parte del “Sistema de Farmacia 2.0” por lo que el diseño se basa en arquitectura MVC. A continuación se presenta el diagrama de clases y el diagrama de colaboración.

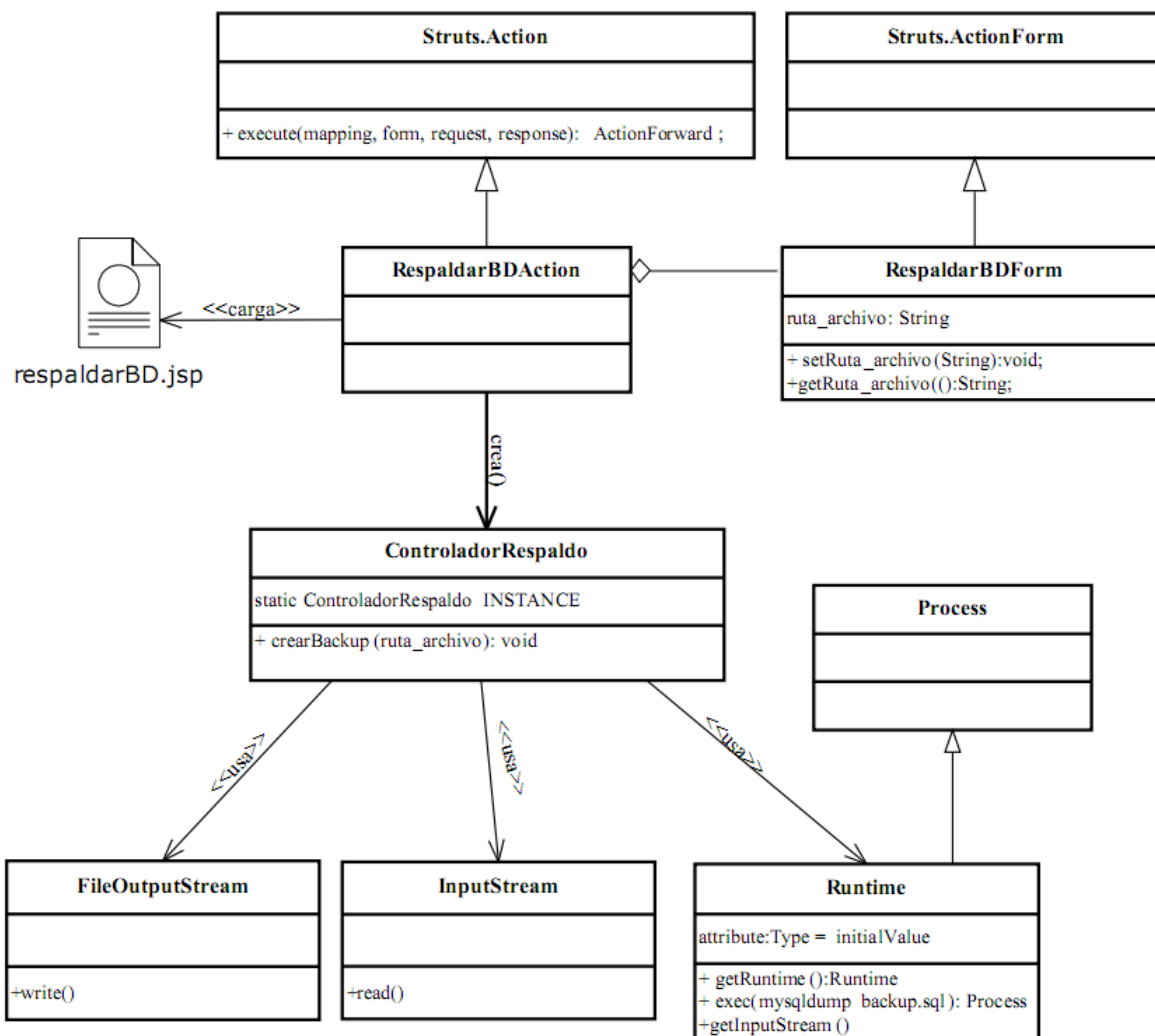


Fig. 27: Diagrama de clases del módulo de administración para crear Backup.

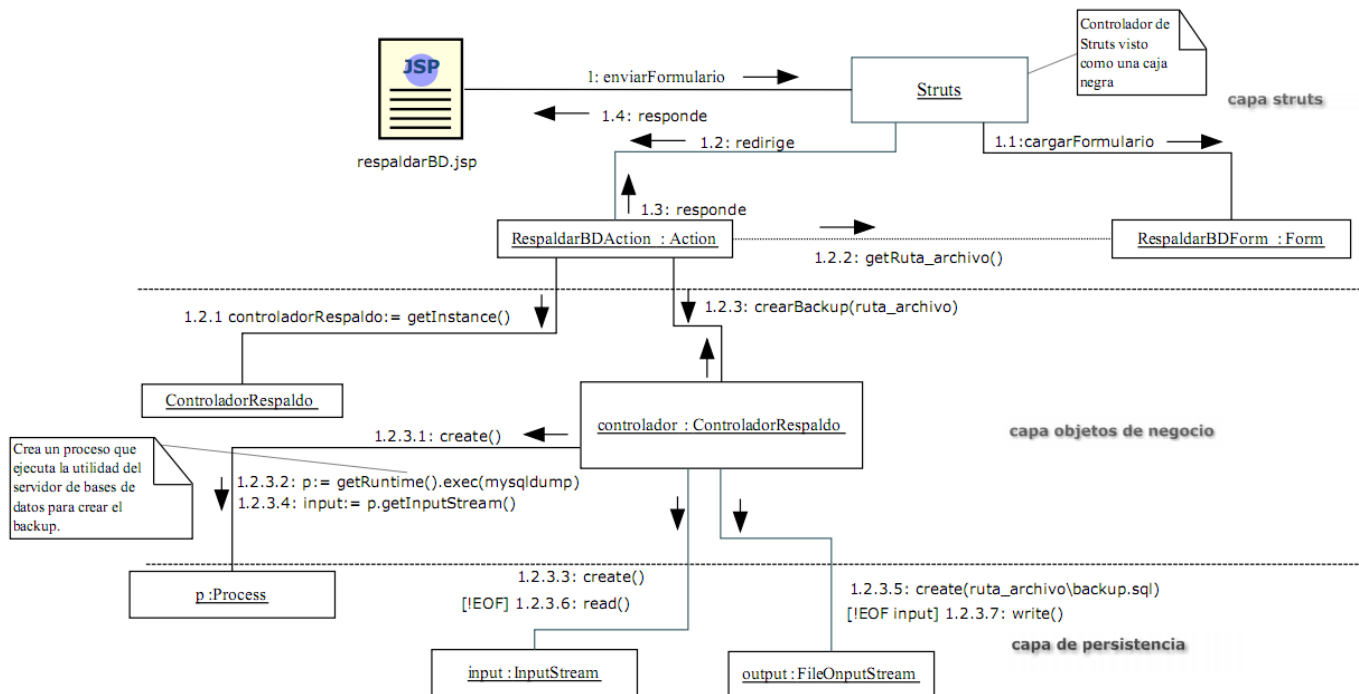


Fig. 28: Diagrama de colaboración del módulo de administración para crear Backup.

7.3 Pruebas del Módulo Backup.

Se realizaron varias pruebas de validación, para verificar si el archivo de respaldo se generaba correctamente.

Caso de Prueba:	Módulo para creación de Backup de la Base de Datos.
1. Propósito de la Prueba:	
<p>Verificar que la aplicación realiza una copia completa de la base de datos “dbsome”, donde residen las tablas y datos de la base de datos del “Sistema de Farmacia 2.0” y del sistema “Plataforma Virtual CESFAM Violeta Parra”</p>	

2. Estado Inicial:

Base de datos “dbsome” con 71 tablas y 1.363.120 registros de datos que corresponden a los sistemas “Plataforma Virtual CESFAM Violeta Parra” y “Sistema de Farmacia 2.0”. (10 tablas son comunes a los dos sistemas).

3. Estado Final Esperado:

Archivo de respaldo creado en disco con el nombre “dbsome_backup_<fecha_hora>.sql” que contiene 71 tablas con datos y 1.363.120 registros de datos.

4. Mensajes en pantalla a visualizar:

El backup se ha creado exitosamente en “<ruta del archivo>\dbsome_backup_fecha_hora.sql”

5. Posibles Excepciones durante la ejecución:

- a) No existe la base de datos.
- b) Error de lectura de la base de datos.
- c) Error de escritura en disco.

6. Condiciones Externas para Funcionamiento:

Servidor de bases de datos “MySQL 5.0” con extensión “mysqldump” funcionando correctamente, servidor de aplicaciones “Apache Tomcat” funcionando correctamente, navegador web cliente funcionando correctamente.

7. Resultados:

Se creó con éxito el archivo “dbsome_backup14-06-2009_02-28-54.sql” en disco, lo que se verificó con el explorador de archivos del sistema operativo. La cantidad de registros se verificó al volver a crear la base de datos a partir del archivo backup, con el SGBD MySQL-Front, el cual generó la base de datos “dbsome” con 71 tablas y 1.363.120 registros de datos.

8. Secuencia de Prueba:

1. Se generó un proceso que inició la ejecución del programa mysqldump, con los parámetros: <computador servidor>, <usuario>, <password>, <archivo backup>
2. Se creó un objeto FileOutputStream que es asociado a un archivo creado en disco llamado “dbsome_backup14-06-2009_02-28-54.sql”(*)
3. Se envía a la interfaz del usuario (página respaldarBD.jsp) el mensaje que se ha creado exitosamente el archivo backup.

(*) Se verificó la existencia del archivo en disco directamente, a través de exploración manual con el administrador de archivos del sistema operativo.

(**) Con la función importación de archivos “.sql” del SGBD MySQL-Front, se procesó el archivo backup creado anteriormente y se verificó que efectivamente fue creada la base de datos “dbsome” y todas sus tablas.

Capítulo 8: Desarrollo Módulo de Administración y Mantenimiento para Restauración.

En el presente capítulo se desarrolla un módulo para dotar al “Sistema de Farmacia 2.0” de la funcionalidad de restauración de la base de datos desde archivos de respaldo o backup creados anteriormente. Este módulo complementa la funcionalidad del módulo para creación de los archivos backup.

8.1 Análisis de Requerimientos del Módulo Restauración.

8.1.1 Requisitos Funcionales de la Aplicación.

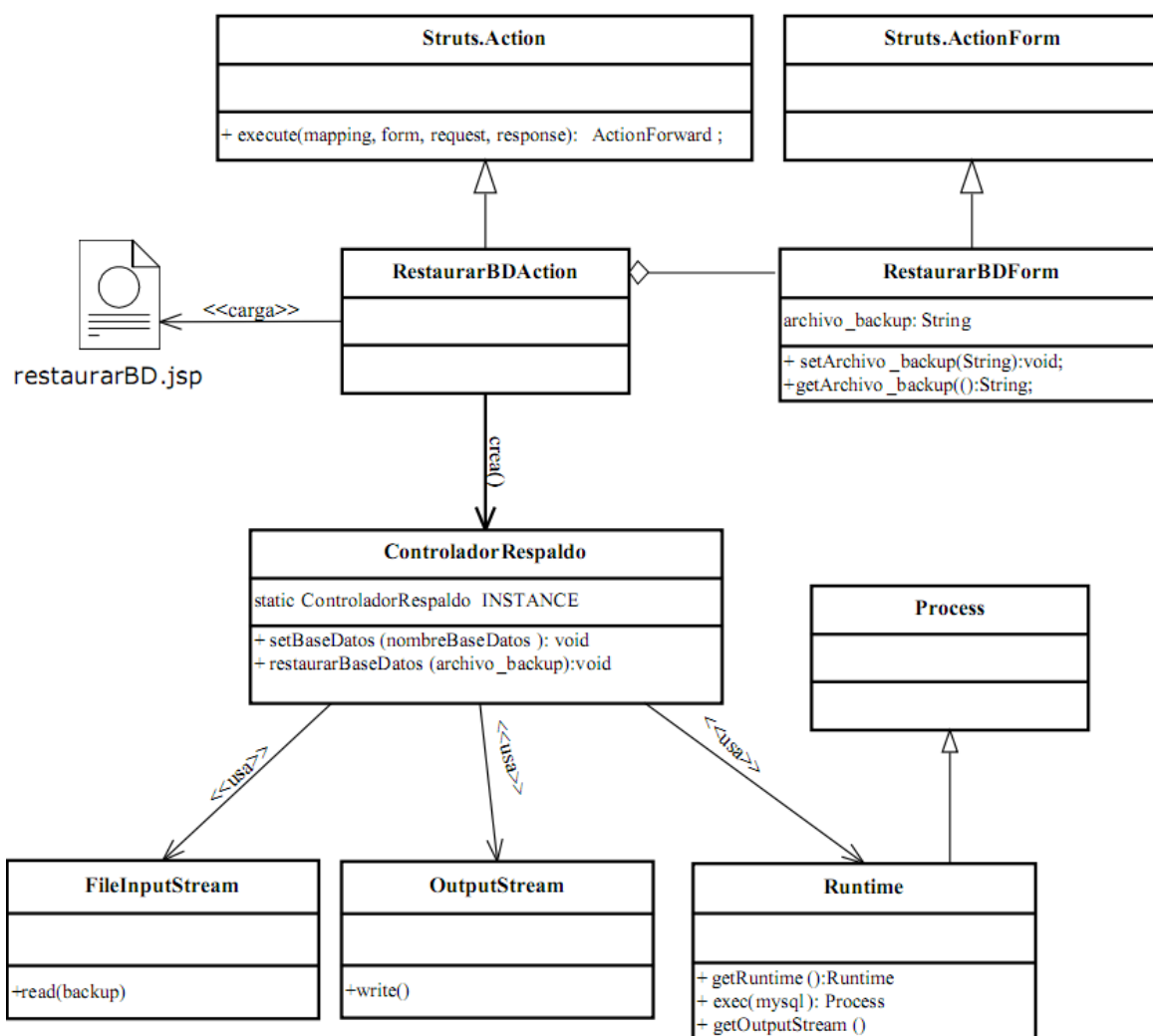
Ref.	Función.
RF1	Restaurar la base de datos del “Sistema de Farmacia 2.0” a partir de un archivo de respaldo (backup) creado anteriormente.
RF2	El módulo debe dar la posibilidad de elegir la unidad y ruta en que se encuentra el archivo de respaldo para poder accesarlo.
RF3	El módulo debe chequear a través del nombre del archivo backup, si corresponde a un archivo válido, para el “Sistema de Farmacia 2.0” y si la fecha de creación es anterior a la actual.
RF4	El archivo backup para restaurar, debe contener tanto la estructura como los datos de la base de datos para crearla y poblarla.

8.1.2 Requerimientos No Funcionales de Módulo de Backup.

Atributos del Sistema	Detalles y Restricciones
Facilidad de uso	El usuario indica que desea realizar la restauración desde un menú de emergencia del sistema, ya que puede darse el caso que no pueda autenticarse normalmente, con lo cual no podría acceder al menú principal del sistema.
Tiempo de Respuesta.	Depende del tamaño de la base de datos a restaurar. Como referencia para 71 tablas y 1.363.120 registros de datos, el tiempo de respuesta es de 7 minutos 07 segundos.
Soporte Sistema Operativo.	Microsoft Windows XP Profesional o Linux (Kernel v2.6).
Soporte Servidor de Base de Datos	MySQL Server Versión 5.0.24a Community
Soporte Máquina Virtual JAVA.	JAVA J2SE Versión 5 ó 6.
Driver JAVA para Base Datos MySQL	Mysql-connector-java-5.0.3
Metáfora de Interfaz del Sistema.	Interfaz Web con gráficos.

8.2 Diseño del Módulo Restauración.

El módulo de restauración de la base de datos, utiliza la misma arquitectura MVC que las demás funcionalidades del “Sistema de Farmacia 2.0”, además su diseño es análogo al diseño del módulo de backup. A continuación se presentan el diagrama de clases y el diagrama de colaboración.



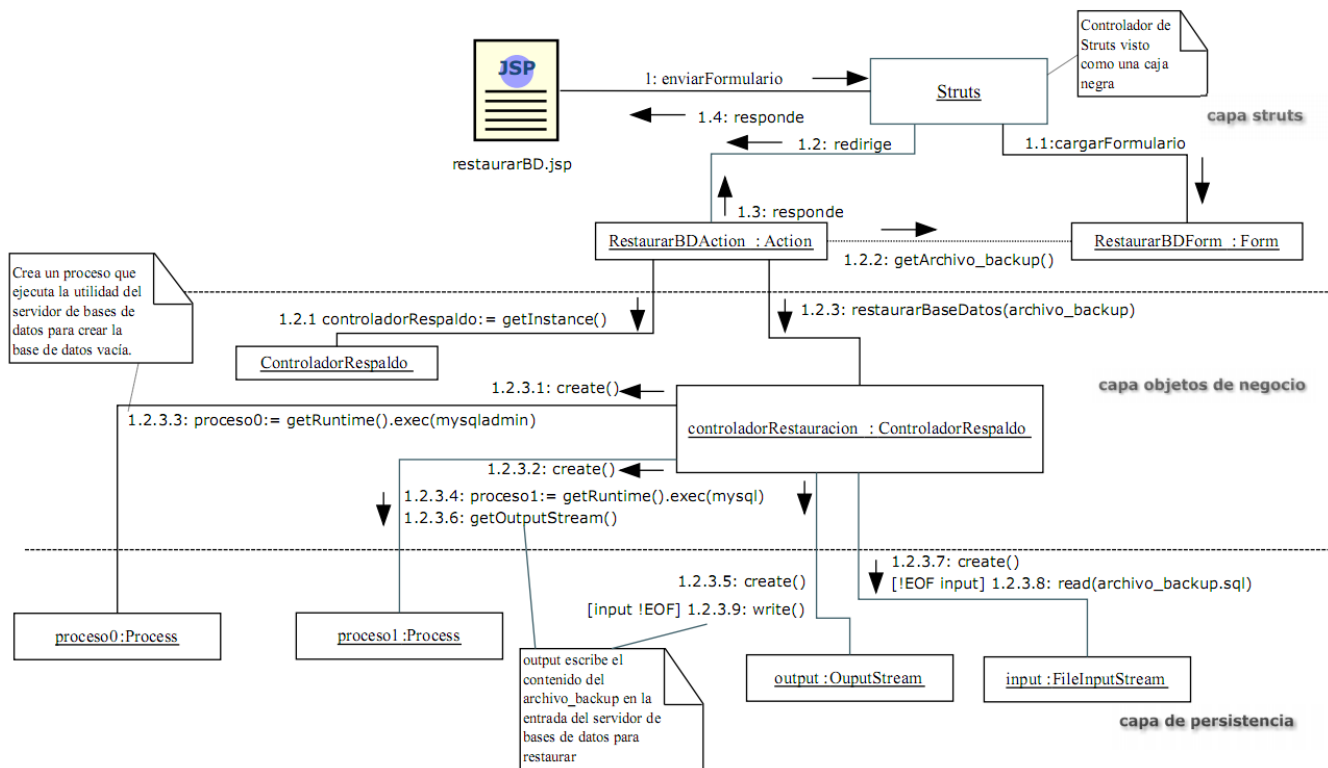


Fig. 29: Diagrama de clases módulo restauración de la base de datos.
 Fig. 30: Diagrama de colaboración módulo restauración de la base de datos.

La clase “ControladorRespaldo” contiene los métodos para respaldar y restaurar la base de datos. Con las clases “Process” y “Runtime” se crea un proceso en el sistema que ejecuta el programa “mysql” que recibe como parámetro el archivo backup desde donde se restaura la base de datos. La clase “FileInputStream” se utiliza para leer el archivo “dbsome_backup_<fecha_hora>.sql” desde el medio de almacenamiento y “OutputStream” para escribirlo en la entrada del programa “mysql” que procesa

las sentencias SQL y genera la base de datos, las tablas, los índices, las claves e inserta los datos.

8.3 Pruebas del Módulo Restauración.

Se realizaron varias pruebas de validación para determinar si efectivamente se logra restaurar completamente la base de datos, utilizando copias de la base de datos conteniendo registros de muestra tomados de la base de datos original.

Caso de Prueba:	Módulo para Restauración de la Base de Datos.
1. Propósito de la Prueba:	
<p>Verificar que la aplicación restaura la base de datos “dbsome”, donde residen las tablas y datos del “Sistema de Farmacia 2.0” y del sistema “Plataforma Virtual CESFAM Violeta Parra”</p>	
2. Estado Inicial:	
<p>No existe la base de datos “dbsome” en el servidor de bases de datos.</p>	
3. Estado Final Esperado:	
<p>Existe la base de datos “dbsome” con 71 tablas y 1.338.803 registros de datos que corresponden a los sistemas “Plataforma Virtual CESFAM Violeta Parra” y “Sistema de Farmacia 2.0”.</p>	
4. Mensajes en pantalla a visualizar:	
<p>La base de datos “dbsome” se ha restaurado exitosamente desde el archivo backup “dbsome_backup_<fecha_hora>.sql”</p>	

5. Posibles Excepciones durante la ejecución:

- a) No se encuentra el archivo “dbsome_backup_<fecha_hora>.sql”.
- b) Error de lectura del archivo backup.

6. Condiciones Externas para Funcionamiento:

Servidor de bases de datos “MySQL 5.0” funcionando correctamente, servidor de aplicaciones “Apache Tomcat” funcionando correctamente, navegador web cliente funcionando correctamente.

7. Resultados:

Se restauró con éxito la base de datos “dbsome” desde el archivo “dbsome_backup29-06-2009_02-28-54.sql” en disco. La base de datos restaurada contiene 71 tablas. Se verificó lo anterior con el SGBD MySQL Administrator.

8. Secuencia de Prueba:

1. Se creó un archivo de respaldo de la base de datos “dbsome” ejecutando el módulo de creación de backup del “Sistema de Farmacia 2.0”.
2. Se eliminó la base de datos “dbsome” con el SGBD MySQL-Front.
3. Se ejecuta el módulo de restauración del “Sistema de Farmacia 2.0”.
 - 3.1 El módulo creó un proceso a través del cual invoca al servidor de bases de datos MySQL y crea una base de datos vacía con nombre “dbsome”
 - 3.2 Se creó otro proceso que invoca al servidor de bases de datos, abriendo una sesión de administrador con la cuenta “root”.
 - 3.3 Se creó objeto FileInputStream para leer desde el archivo “dbsome_backup29-06-2009_02-28-54.sql” y se creó un objeto OutputStream para escribir el contenido del archivo backup en la entrada del programa servidor de bases de datos conectado a la recién creada base de datos “dbsome” para crear las tablas e insertar los datos.

Capítulo 9: Desarrollo Módulo de Administración para Cierre Anual de Transacciones.

En el presente capítulo se muestra el desarrollo del módulo de administración para el cierre anual de transacciones. Este módulo tiene el objetivo de permitir al usuario del “Sistema de Farmacia 2.0”, mover registros de transacciones de cierta antigüedad a tablas historiales cuando el usuario lo solicite. El sistema mueve los registros de las transacciones que fueron realizadas desde una fecha determinada hacia atrás. Esta operación es una forma

de mantenimiento de las tablas de la base de datos que son utilizadas más intensivamente por el “Sistema de Farmacia 2.0”, para evitar su crecimiento excesivo, ya que en cada sesión de usuario existen operaciones repetitivas que deben recorrer dichas tablas y a mayor número de registros mayor será el tiempo de búsqueda y recuperación de los datos. Por otra parte es una forma de archivar datos históricos que cuando alcanzan cierta antigüedad ya no es necesario accesarlos regularmente, sino eventualmente, para lo que siguen disponibles en caso de que el usuario los requiera.

9.1 Análisis de Requerimientos Módulo Cierre Anual de Transacciones.

Los siguientes son los requisitos funcionales y no funcionales del módulo de cierre anual de transacciones.

9.1.1 Requisitos Funcionales de la Aplicación.

Ref.	Función.
RF1	Permitir el cierre de transacciones de los siguientes tipos: Dispensación de medicamentos a pacientes crónicos; Dispensación de recetas a pacientes de morbilidad; Distribución de insumos y medicamentos a otras farmacias; Devolución de insumos y medicamentos; Movimientos de inventario de insumos y medicamentos; Movimientos de pedidos de productos al departamento de abastecimiento,
RF2	Mover a tablas historial, los registros de transacciones pasadas que tengan antigüedad desde una fecha dada por el usuario hacia atrás.

9.1.2 Requerimientos No Funcionales de Módulo de Backup.

Atributos del Sistema	Detalles y Restricciones

Facilidad de uso	El usuario indica que desea realizar el cierre de transacciones de un tipo, a través del menú del sistema e indica la fecha desde la cual hacia atrás se moverán los registros al historial.
Tiempo de Respuesta.	Depende de la cantidad de registros a mover a las tablas historiales. Entre 3 y 10 minutos.
Soporte Sistema Operativo.	Microsoft Windows XP Profesional o Linux (Kernel v2.6).
Soporte Servidor de Base de Datos	MySQL Server Versión 5.0.24a Community
Soporte Máquina Virtual JAVA.	JAVA J2SE Versión 5 ó 6.
Driver JAVA para Base Datos MySQL	Mysql-connector-java-5.0.3
Metáfora de Interfaz del Sistema.	Interfaz Web con gráficos.

9.2 Diseño Módulo Cierre Anual de Transacciones.

El diseño del módulo de cierre anual de transacciones está ligado al modelo actual de la base de datos del “Sistema de Farmacia 2.0”, ya que principalmente su objetivo es mover los registros de transacciones antiguas a tablas distintas de las originales, lo que supone que existan tablas historiales con igual estructura de atributos para almacenar los registros salientes.

9.2.1 Modelo de Datos.

Las tablas de la base de datos que están involucradas en el diseño de este módulo son las siguientes según las transacciones que almacenan:

- a) Dispensación de medicamentos a pacientes crónicos:
Tablas “tarjeta_dispensacion” y “detalle_tarjeta”.
- b) Dispensación de recetas a pacientes de morbilidad:
Tablas “receta” y “prescripción”.
- c) Distribución de insumos y medicamentos a farmacias:
Tablas “distribución” y “despacho”.
- d) Devolución de insumos y medicamentos:
Tabla “devolución”.
- e) Movimientos de inventario de insumos y medicamentos:
Tabla “bincard”.
- f) Movimientos de pedidos de productos al departamento de abastecimiento:
Tablas “pedido” y “pedido_detalle”.

Por cada una de estas tablas, se crea una tabla con igual estructura y al nombre se le agrega el sufijo “historial”, por ejemplo “receta_historial”. Para recordar las relaciones que existen entre las tablas y el objetivo de cada una de ellas, se puede volver a la sección 4.4 donde se muestra el modelo de datos completo y definición de tablas de la base de datos del “Sistema de Farmacia 2.0”

9.2.2 Diagrama de Clases.

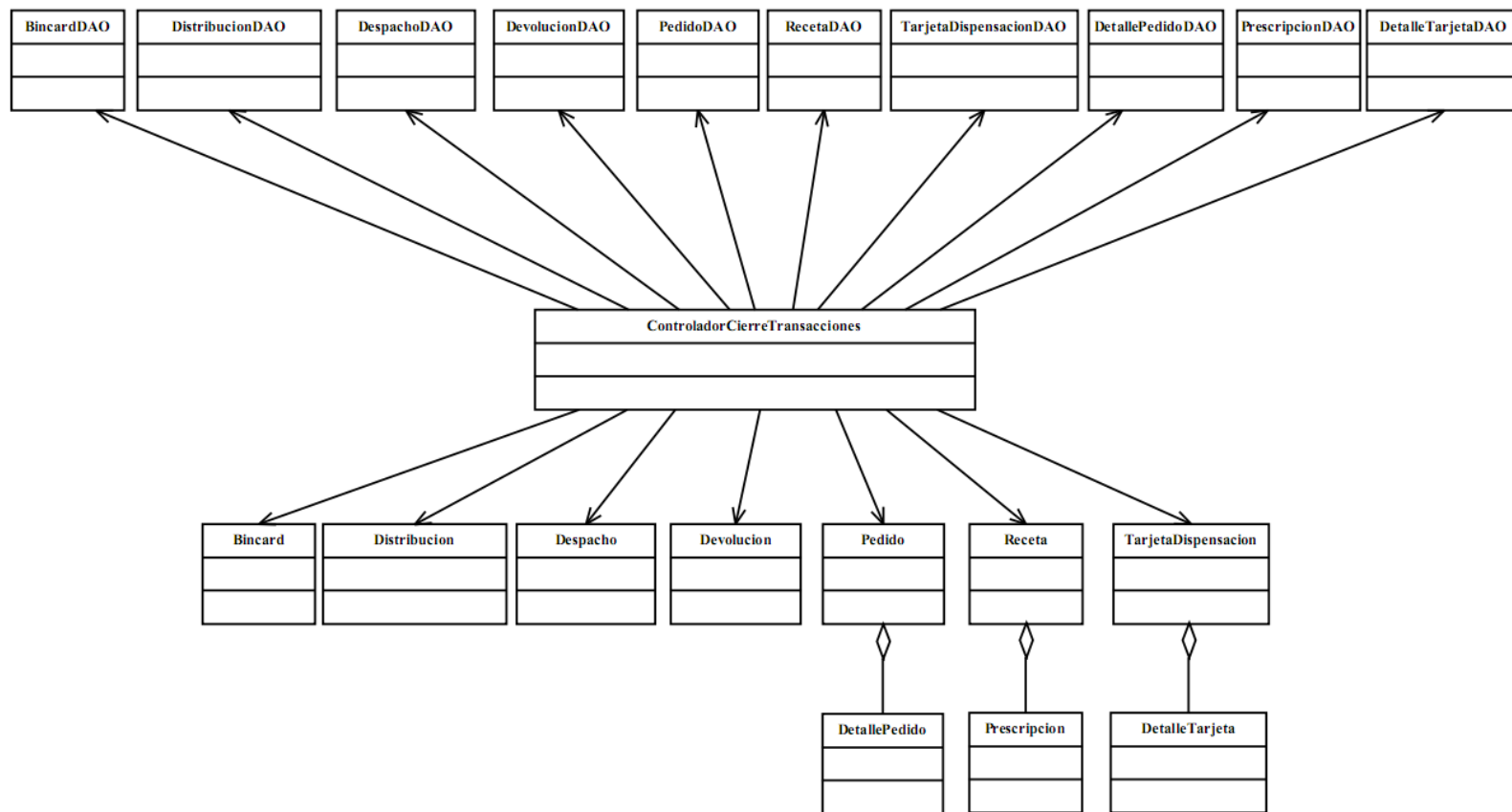


Fig. 31: Diagrama de clases módulo Cierre Anual de Transacciones.

9.2.3 Diagrama de Colaboración.

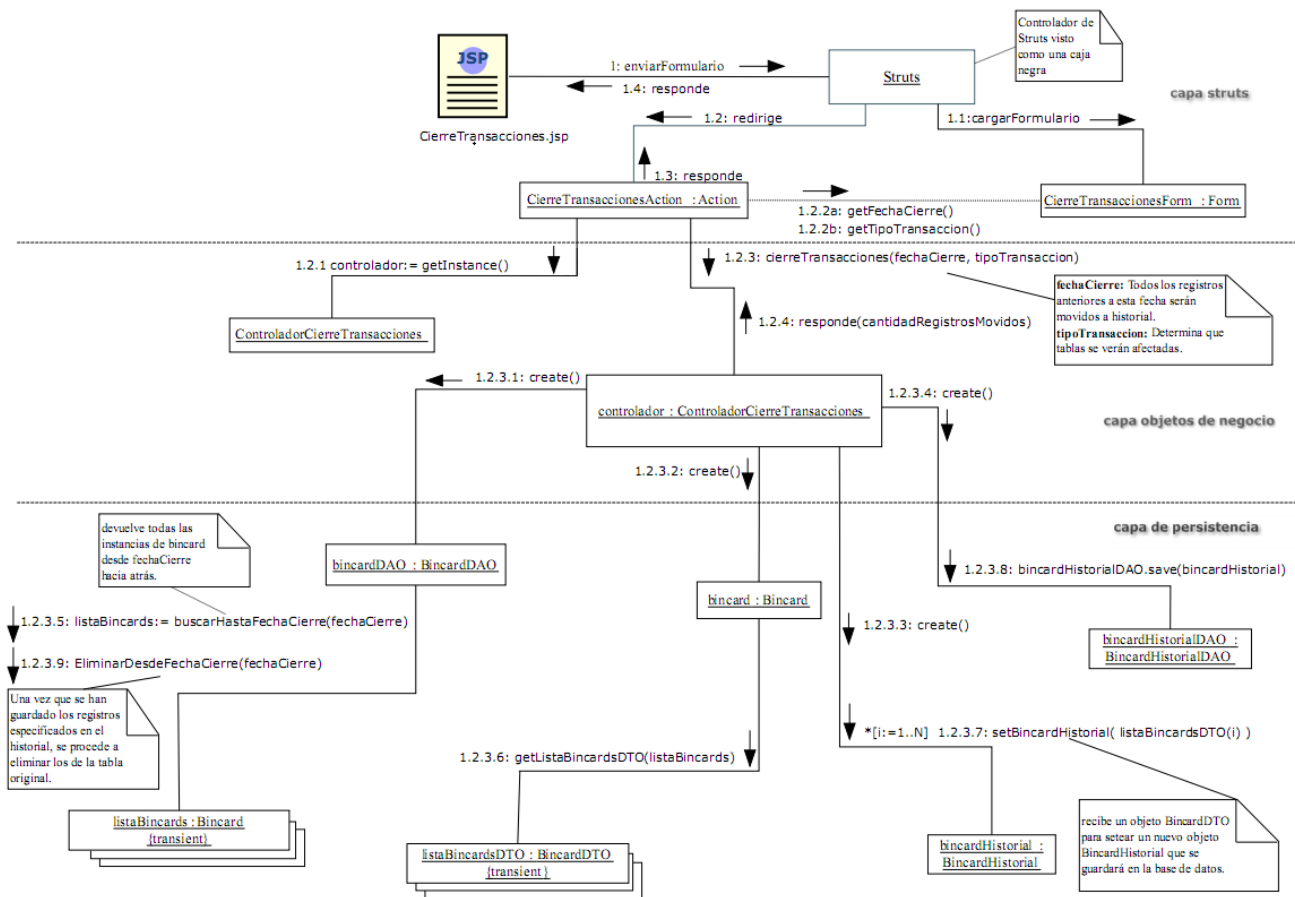


Fig. 32: Diagrama de colaboración Cierre de Transacciones, para el caso Movimientos de Inventario

En el diagrama de colaboración se presenta por simplicidad el caso del cierre de las transacciones de inventario, asociado a la tabla “bincard” y “bincard_historial” de la base de datos, teniendo presente que la operación es análoga para cada tipo de cierre de los mencionados en el punto 9.2.1.

9.3 Pruebas Módulo Cierre Anual de Transacciones.

Las pruebas realizadas al módulo son para verificar su funcionamiento correcto.

Caso de Prueba:	Módulo Cierre Anual de Transacciones: Dispensación de medicamentos a pacientes crónicos.
<p>1. Propósito de la Prueba:</p> <p>Verificar que el módulo de cierre anual de transacciones, realiza correctamente el cierre de las operaciones de <i>dispensación de medicamentos a pacientes</i> crónicos año 2008, es decir, mueve todos los registros con fecha de entrega del medicamento igual o anterior al 31-12-2008 (fecha cierre), desde la tabla “detalle_tarjeta” a la tabla “detalle_tarjeta_historial” de la base de datos del sistema.</p>	
<p>2. Estado Inicial:</p> <p>La tabla “detalle_tarjeta” tiene un total de 302.345 registros de dispensaciones a pacientes crónicos con fecha de entrega dentro de los años 2008 y 2009. La tabla “detalle_tarjeta_historial” no contiene registros.</p>	
<p>3. Estado Final Esperado:</p> <p>La tabla “detalle_tarjeta_historial” contiene 232.357 registros de dispensaciones a pacientes crónicos realizadas durante el año 2008.</p>	
<p>4. Mensajes en pantalla a visualizar:</p> <p>232.357 registros fueron movidos al historial de transacciones.</p>	

5. Posibles Excepciones durante la ejecución:

- a) No se puede conectar a la base de datos.
- b) No se puede leer la tabla “detalle_tarjeta”.
- c) No se puede escribir la tabla “detalle_tarjeta_historial”.
- d) La fecha de cierre es mayor que la fecha actual.

6. Condiciones Externas para Funcionamiento:

Servidor de bases de datos “MySQL 5.0” funcionando correctamente, servidor de aplicaciones “Apache Tomcat” funcionando correctamente, navegador web cliente funcionando correctamente.

7. Resultados:

Se movieron con éxito a la tabla “detalle_tarjeta_historial” 232.357 registros con fecha de cierre 31-12-2008.

8. Secuencia de Prueba:

1. Se cargaron 302.345 registros de prueba en la tabla “detalle_tarjeta” de la base de datos.
2. Se ejecuta el módulo del menú Cierre Anual de Transacciones.
3. Se elije cerrar transacciones de dispensación de medicamentos a pacientes crónicos.
4. Se ingresa como fecha de cierre el 31-12-2008.
5. El sistema lee todos los registros de la tabla “detalle_tarjeta” con fecha igual o anterior al “31-12-2008” y guarda una copia de ellos en la tabla “detalle_tarjeta_historial”.

6. Cuando ha finalizado la inserción de las copias de registro en “detalle_tarjeta_historial”, el sistema elimina los registros de la tabla “detalle_tarjeta”.

7. El sistema muestra en pantalla el número de registros movidos al historial.

Capítulo 10: Puesta en Marcha del Sistema de Farmacia 2.0.

10.1 Capacitación a los usuarios.

La capacitación a los usuarios está dirigida a las personas que serán administradores del sistema y usuarios cotidianos del sistema, es decir, la Jefa de Farmacia (Químico Farmacéutico), las Funcionarias Supervisoras y las Funcionarias Técnicas de Farmacia. Todas ellas atienden al público directamente, recibiendo las recetas y documentos en que se solicitan los insumos y medicamentos de los pacientes, utilizando el Sistema de Farmacia para chequear la existencia de los productos solicitados, preparando las entregas de los mismos y registrando la operación de dispensación y datos asociados en el sistema; además de las labores administrativas y de gestión que se llevan a cabo, por parte de la Jefa de Farmacia y Supervisoras, apoyadas por el sistema de información.

La capacitación consta de una sesión expositiva y sesiones de entrenamiento, que se realizan a las siete integrantes del equipo de las farmacias.

En la sesión expositiva, que es una reunión general de una hora y media, para todas las funcionarias, se muestra el “Sistema de Farmacia 2.0”, a través de una presentación proyectada, para conocer las interfaces de usuario y su relación con las actividades propias del trabajo de farmacia, explicando las funcionalidades y su similitud con las del “Sistema Heredado de Farmacia 1.0”, dando oportunidad de despejar dudas por parte de las funcionarias.

Respecto de las sesiones de entrenamiento, se realiza una sesión a cada funcionaria, con una duración de una hora aproximadamente cada sesión, en grupos

de dos personas. Habrá sesiones tipo para las funcionarias que trabajarán con perfil de usuario normal del sistema y usuarias con perfil de administrador y/o supervisor del sistema.

La capacitación se realiza en un computador asignado para el efecto, el cual se instala en la sala de educación existente en el CESFAM. Dicho computador tiene instalada una copia del “Sistema de Farmacia 2.0”, es decir, el software servidor de aplicación, el código ejecutable del sistema, el software servidor de base de datos, una copia de la base de datos con datos de prueba que fueron migrados y el software navegador cliente para iniciar sesión en el sistema.

Después de la sesión expositiva y sesiones de entrenamiento, se sigue apoyando a las funcionarias, durante el proceso de puesta en marcha, ayudando al uso del sistema en forma presencial en las farmacias, durante 5 jornadas, a fin de despejar la mayor cantidad posible de dudas de su uso y lograr el aprendizaje efectivo por parte de las usuarias.

10.2 Instalación del Sistema Informático.

Dado que el “Sistema de Farmacia 2.0” es una aplicación Web, la instalación de todos los componentes del sistema, se realiza en el computador servidor del CESFAM, donde además se encuentra instalado el sistema Web “Plataforma Virtual del CESFAM Violeta Parra”, el cual para funcionar utiliza el mismo software servidor de base de datos y software servidor de aplicaciones web que el sistema en implantación.

La instalación del “Sistema de Farmacia 2.0” se resume en los siguientes pasos:

1. Instalar servidor de bases de datos “MySQL 5.0.24a Community NT” para Windows. (Si ya está instalado se continúa con el siguiente punto).
2. Instalar SGBD “MySQL Administrator Versión 1.2.1.2” para Windows. (Opcional. Si ya está instalado se continúa con el siguiente punto).

3. Instalar “Java Platform Standard Edition 6” (J2SE 6) que contiene la máquina virtual de Java, “Java JRE versión 1.6” para Windows.
4. Instalar el servidor de aplicaciones Web Java: “Apache Tomcat versión 6.0.16” para Windows.
5. Copiar el archivo “farmacia.war” en el computador servidor, en la ubicación:
“C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\webapps”
6. Iniciar el servidor de bases de datos “MySQL 5.0” instalado anteriormente y crear la Base de Datos “dbfarmacia20” en el entorno del servidor de bases de datos MySQL. (Archivo “dbfarmacia20.sql” es importado al servidor de bases de datos con el SGBD MySQL-Front o MySQL Administrator).
7. Iniciar el servidor de aplicaciones web “Apache Tomcat 6.0” instalado.
8. Iniciar sesión en el “Sistema de Farmacia 2.0”, con datos de acceso especiales para la instalación y ejecutar “Migración-FS” desde el menú del sistema para integrar la base de datos “dbfarmacia” con “dbsome”.
9. Una vez finalizada la ejecución de “Migración-FS”, se debe cerrar sesión.

Luego de haber ejecutado correctamente los pasos anteriores, se está en condiciones de iniciar sesión como usuario en el “Sistema de Farmacia 2.0”, introduciendo los datos de acceso.

10.3 Evaluación de la Puesta en Marcha del Sistema Informático.

Se evalúan varios aspectos considerados importantes en el proceso de puesta en marcha del “Sistema de Farmacia 2.0”:

1. A nivel de los usuarios(as):

En cuanto a la Capacitación: Es imprescindible capacitar a los usuarios previamente en el uso del sistema, entrenarlos y simular las posibles situaciones de

uso del sistema. La capacitación influye directamente en el grado de aceptación del nuevo sistema por parte de los usuarios.

Aceptación: En general, la puesta en marcha del “Sistema de Farmacia 2.0” es acogida con buena disposición, porque se espera que el nuevo sistema solucione las dificultades que el sistema de información heredado estaba teniendo. Sólo existe un poco de resistencia de algunos usuarios, la que es normal por lo que significa enfrentar un sistema nuevo. La capacitación aclara las dudas del uso del sistema y contribuye a reafirmar la seguridad del usuario al utilizarlo, sumado a esto el hecho de que una vez conocidas las interfaces del usuario se reconocen muchos elementos en pantalla que se han mantenido similares al sistema de información heredado.

Aprendizaje: Con el primer acercamiento que los usuarios han tenido en la sesión expositiva, se puede continuar con la fase de entrenamiento personal, la que es determinante, ya que el sistema es utilizado en cada dispensación de medicamento a pacientes, por lo que cada usuario(a) quiere aprender a utilizarlo correctamente y así no cometer errores en el trabajo diario. Se destaca positivamente la responsabilidad con que los usuarios esperan aprender el uso del sistema.

2. A Nivel del Software:

Se evalúa positivamente la puesta en marcha como proceso global, ya que viene a solucionar un tema pendiente desde hace al menos dos años en que se necesitaba actualizar o cambiar el sistema de farmacia.

La migración de los datos es un punto crítico, ya que se esperaba contar con todos los datos disponibles en el sistema de información heredado, pero en el desarrollo de la migración se fueron tomando decisiones tendientes a elegir entre los datos de mejor calidad, comparando con los almacenados en la base de datos de la “Plataforma Virtual del CESFAM Violeta Parra”. Las múltiples decisiones y criterios quedaron plasmados en la aplicación “Migración-FS”, la que arrojo

resultados satisfactorios, sin descartar que a medida que pase el tiempo, se vaya reafirmando esta primera evaluación, pues será la utilización del sistema por parte de los usuarios la que vaya consolidando la información.

En cuanto a los módulos de administración implementados, se evalúa positivamente el valor agregado que dan al sistema de información, ya que cubren necesidades que se fueron visualizando durante el desarrollo del “Sistema de Farmacia 2.0”.

3. Nivel Hardware:

La evaluación a nivel de equipos es positiva, dado que el CESFAM cuenta con un equipo servidor adquirido hace pocos meses y de capacidad acorde a lo requerido, para el “Sistema de Farmacia 2.0”.

Al evaluar los computadores clientes se determina revisar su funcionamiento general, revisar el software antivirus, actualizarlo e instalarles la versión actualizada del navegador web “Mozilla Firefox” para trabajar adecuadamente.

Capítulo 11: Conclusiones.

Fue de especial utilidad e importancia en el desarrollo de la migración de los datos, el estudio y aplicación de algunos elementos propuestos por Caro *et. al*, en su investigación de metodologías de migración, en lo relacionado al preprocesamiento de los datos, antes de mapearlos a la base de datos del sistema que se pone en marcha.

El desarrollo de los módulos de administración requirió, un sólido análisis y diseño, que se ve en gran medida apoyado por las herramientas de ingeniería existentes para tales efectos como, diagramas UML de clases y de colaboración.

Sin duda alguna, una de las etapas más importantes para la puesta en marcha junto con la migración de los datos e instalación del sistema, son las pruebas que se aplican al sistema informático, lo que ayuda a mejorar las deficiencias y así poder dar garantía de que la aplicación en funcionamiento es confiable desde el punto de vista de resolver la necesidad para la que fue construida, es decir, la conservación de los datos y la funcionalidad.

Otro punto importante a destacar en el proceso de puesta en marcha, es la importancia de la capacitación a los usuarios del sistema. Si bien las horas que se dedican a enseñar el uso del sistema y entrenar a los usuarios no son suficientes para lograr que estos últimos dominen todos los aspectos del nuevo sistema, si es necesario y beneficioso para lograr que las personas se sientan seguras de sí mismas y eso sea un aliciente para cada usuario en continuar aprendiendo el uso de la aplicación hasta tener experticia.

Uno de los procesos más críticos y largos de realizar fue la migración de los datos por el esfuerzo adicional que trae consigo compatibilizar e integrar tres modelos de datos (modelo heredado, modelo actualizado y modelo al cual se integra). Para esto fue fundamental el análisis de los datos, el conocimiento de las reglas de negocio de todos los

modelos, la aplicación del modelado de datos con MER, la preparación y preprocesamiento de los datos, la construcción de las aplicaciones de migración y mapeo.

Bibliografía.

- [ADRIA96] Adriaans P., Zantinge D., Data Mining, Addison-Wesley. 1996.
- [AGRAW93] Agrawal R., Imielinski T, Swami A., Data Mining: A Performance Perspective, IEEE Transactions on Knowledge and Data Engineering. December 1993. pp : 914 - 925.
- [Bredemeyer, 2001] Malan, R., Bredemeyer, D. (2001). Defining Non-Functional Requirements. (PDF) Bredemeyer Consulting.
[On-line] http://www.bredemeyer.com/pdf_files/NonFunctReq.PDF
- [BRODI95] BRODIE M., STONEBRAKER M. Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach. Morgan Kaufman Publishers. 1995.
- [BISBAL99] BISBAL J., LAWLESS D., WU B., GRIMSON J.. “Legacy Information Systems: Issues and Directions”. IEEE Software, pp: 103-111. Septiembre/Octubre 1999.
- [Caro *et al.* 2002] Caro M.A., Bocca J., Campos D. “Migración de Sistemas Heredados. Una Metodología de Apoyo Basada en el Uso de Herramientas de KDD (Knowledge Discovery In Databases)”. Revista Ingeniería de Sistemas, volumen XVI, N° 1, Junio 2002.
- [Calero *et al.* 2006] Calero Coral, Genero Marcela, Ruiz Francisco. "Bases de Datos. Diseño Conceptual, Lógico y Físico". Apuntes Asignatura Bases de Datos. Esc. S. de Informática. Universidad de Castilla La Mancha. 2006. [On-line] www.uclm.es

- [CVP 2009] DIRECCIÓN CONSULTORIO VIOLETA PARRA DE CHILLÁN, 2009. Cuenta Pública 2008.
- [FAYYAD96] Fayyad U., Piatetsky-Shapiro G., Smyth P., The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communication of the ACM. Vol. 39, N° 11, November 1996.
- [ECLIPSE2008] Sitio web de la organización Eclipse, Desarrolladores del IDE Eclipse [En línea] <http://www.eclipse.org> [junio 2009]
- [Elmasri & Navathe 1997] Elmasri, Navathe; "Sistemas de Bases de Datos, conceptos fundamentales", Adison Wesley, 2a ed. 1997.
- [Gilfillan2004] Gilfillan, I. 2004. "La Biblia MySQL", Anaya Multimedia, 1ª ed. 2004.
- [González, R. 2006] González, R., "Aplicación Web para consulta de pacientes internados Hospital Herminda Martín de Chillán". Tesis para optar al título de Ingeniero de Ejecución en Computación e Informática. Chillán, Universidad del Bío-Bío, Facultad de Ciencias Empresariales, 2006.
- [Goodwill, J. 2002] "Mastering Jakarta Struts". Wiley Publishing Inc., 1ª Edición. 2002.
- [HALL2006] Hall M. Tutorial Servlets y JSP. Sitio Web "Programación en Castellano" 2007. [En línea]
http://www.programacion.com/java/tutorial/servlets_jsp/1/[consulta: junio 2009].
- [Hermosilla 2000] HERMOSILLA, A. 2000. Sistema De Control De Farmacia Para El Consultorio Violeta Parra – Chillán. Universidad del Bío-Bío, 2000. 329p.
- [Larman, 1999] Larman, C. 1999. UML y patrones: Introducción al análisis y diseño orientado a objetos. México:Prentice-Hall,1999.

- [Larman 2003] LARMAN, Craig. 2003. UML y Patrones. Una Introducción al análisis y diseño orientado a objetos y al proceso unificado. 2ª ed. Pearson Educación S.A. 624p.
- [MINSAL 1995]: MINISTERIO DE SALUD DE CHILE. Guía Para la Organización y Funcionamiento de la Atención Farmacéutica en Atención Primaria de Salud. Santiago, 1995. 46p.
- [en línea] Sitio Web del Ministerio de Salud de Chile.
<http://www.minsal.cl/juridico/RESOLUCION_1089_95.doc>
[consulta: 10 mayo 2009].
- [MVARAS2008] Varas M. Página Personal de Docente. Universidad de Concepción.
[En línea] <http://www.inf.udec.cl/~mvaras/> [consulta: Junio 2008].
- [OMS/UNICEF 1978] ORGANIZACIÓN MUNDIAL DE LA SALUD (OMS) y UNICEF. Conferencia Internacional sobre Atención Primaria de Salud. 6 al 12 de septiembre de 1978. Kazajistán. Ex URSS. Declaración de Alma-Ata. [en línea] Sitio Web de la Organización Panamericana de la Salud. N° V y VI [en línea]
http://www.paho.org/Spanish/dd/pin/alma-ata_declaracion.htm
[consulta: 13 mayo 2009]
- [Pressman 2002] Pressman, R., “Ingeniería del Software; Un Enfoque Práctico”. McGraw-Hill. 5a Edición, 2002.
- [Rumbaugh *et al.* 2007] RUMBAUGH, James., JACOBSON, Ivar., BOOCH, Grady. El Lenguaje Unificado de Modelado Manual de Referencia. 2ª ed. Madrid, Pearson Educación S.A., 2007. 688p.
- [Sommerville 2005] Sommerville, I., “Ingeniería de Software”. 7a Edición. 2005.

[Soto & Rodríguez 2000] Soto M.A., Rodríguez A. "Modelamiento de Datos, Modelo Entidad Relación (MER)". Apuntes Sistemas de Información I. Universidad del Bio-Bío. Chillán 2000.

[Vásquez & Valdés 2004] Vásquez, H. y Valdés, C., 2004. "Implementación de un sistema de apoyo al proceso de mejoramiento genético del trigo." Memoria para optar al título de Ingeniero de Ejecución en Computación e Informática. Chillán, Universidad del Bío-Bío, Facultad de Ciencias Empresariales. 2004.

[Wu 1997] Wu B. *et al.* "Legacy System Migration: A Legacy Data Migration Engine". XVII Conferencia Internacional de Bases de Datos, DATASEM '97. Berna, República Checa, 12 al 14 de Octubre de 1997.[en línea]
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7940>
[consulta: Abril 2009].

Anexos.

Interfaces de Usuario:

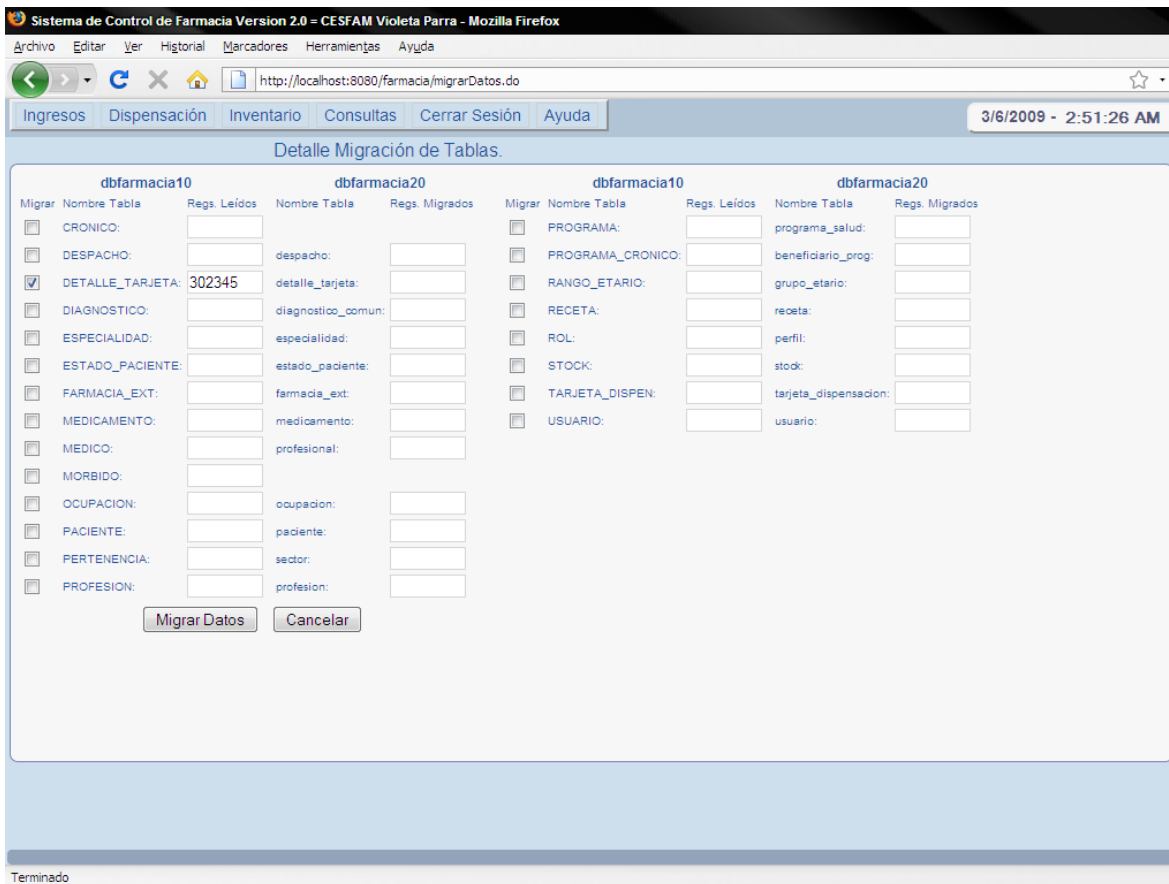


Fig. 33: Pantalla del módulo de Migración de "dbfarmacia10" a "dbfarmacia20".

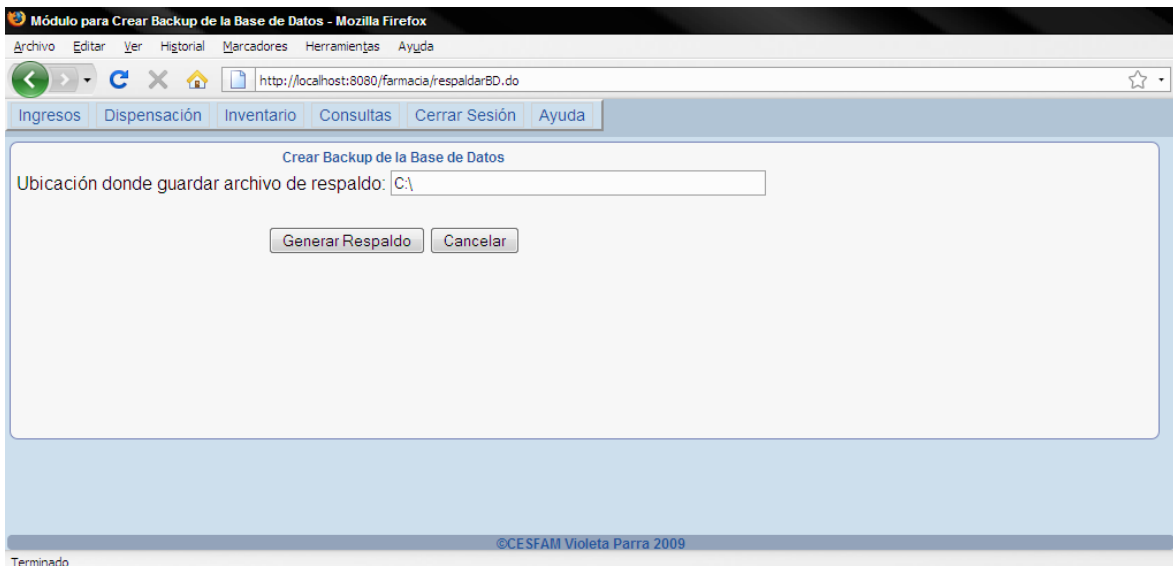


Fig. 35: Pantalla módulo Respaldar Base de Datos.



Fig. 34: Pantalla Módulo Restaurar Base de Datos.

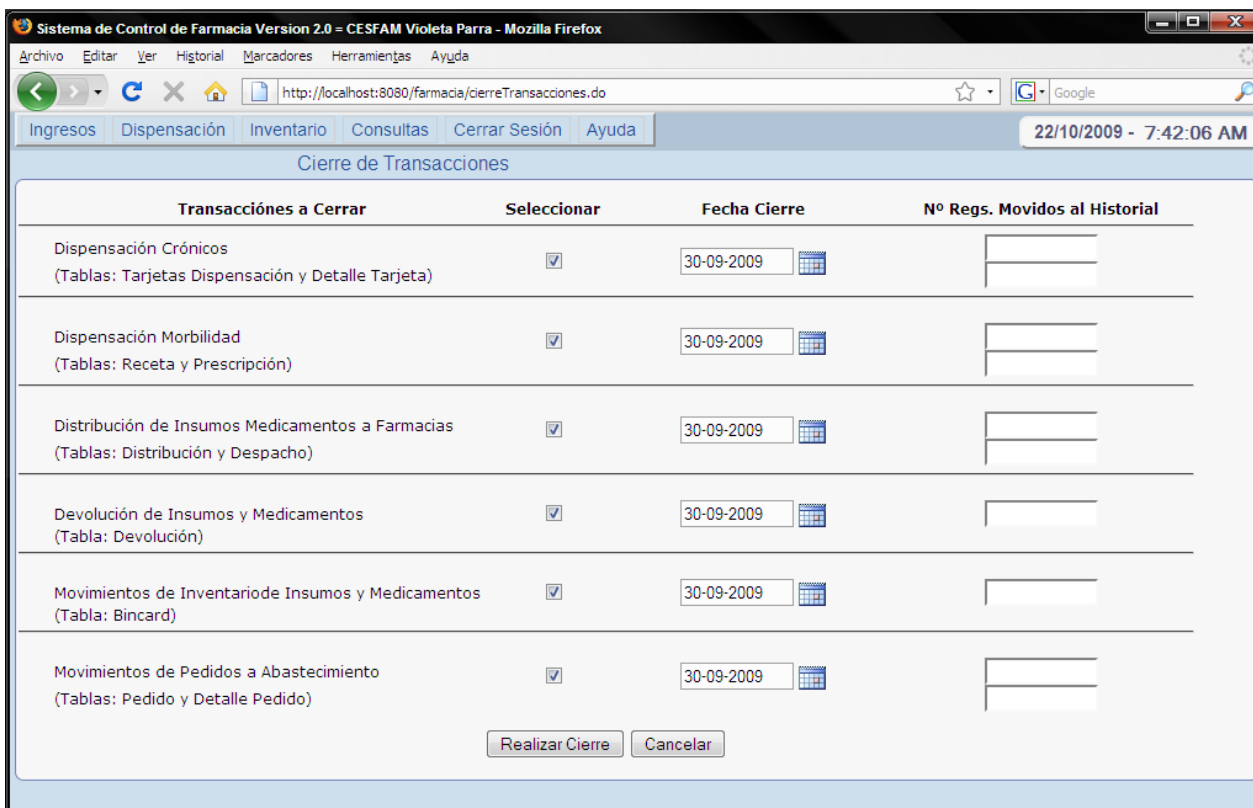


Fig. 36: Pantalla Módulo Cierre de Transacciones.

Descripción Detallada de Casos de Uso Sistema de Farmacia 2.0.

Caso de uso N° 1.	Ingresar Datos Autenticación
Actores	Usuario del sistema
Propósito	Identificar, validar y permitir el acceso y uso del Sistema de Control Farmacia.
Resumen	El usuario requiere acceder al sistema para realizar tareas en este, para lo cual debe ingresar su identificador de usuario y su contraseña, mediante los que el sistema determina su perfil y las tareas asociadas que le es permitido realizar. En caso de ser exitoso, el usuario efectivamente accede al sistema y en caso contrario se le informa el error.
Precondición	El usuario está ejecutando el navegador y a través de él se conecta al sistema de farmacia.
Postcondición	Se muestra pantalla con menú de opciones del sistema de farmacia.

Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario decide ingresar al sistema.	
2. El usuario dirige el browser a la dirección del sistema.	
	3. El sistema despliega en pantalla el formulario para requerir los datos de identificación y contraseña de usuario.
4. El usuario ingresa su identificación y contraseña.	
	5. El sistema despliega el menú de opciones de trabajo según el perfil del usuario.
Cursos alternos:	
4a.- Los datos del usuario son inválidos. Se indica el error al usuario, no se acepta el acceso al sistema y se vuelven a pedir los datos.	

Caso de uso N° 2.	Ingresar Datos Consultorio
Actores	Usuario Administrador del sistema.
Propósito	Capturar y almacenar los datos básicos del consultorio.
Resumen	El usuario administrador requiere introducir al sistema, los datos que identifican al consultorio para poder comenzar la operación del programa por primera vez o para modificar datos anteriores.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso de los datos del consultorio.
	Primario.

Tipo	
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Consultorio”.	
	2. El sistema despliega en pantalla el formulario para ingreso de los datos del consultorio.
3. El usuario completa el formulario con los datos del consultorio.	
	4. El sistema verifica si los datos proporcionados ya existen previamente.
	5. Los datos son grabados.
	6. Despliegue en pantalla de mensaje indicando que los datos fueron ingresados y

	finaliza caso de uso.
Cursos alternos:	
	4a.- Los datos proporcionados ya existen en el sistema y son los mismos, por lo que se envía mensaje al usuario informándole de la situación. No se graban los datos y finaliza el caso de uso.
	4b.- Los datos proporcionados difieren de los que ya existen en el sistema. Mediante mensaje en pantalla, se le solicita al usuario confirmar la modificación.
4c.- El usuario confirma la modificación de los datos.	
	4d.- Se graban los datos modificados.

Caso de uso N° 3.	Ingresar Datos Nuevo Usuario del Sistema
Actores	Usuario Administrador del sistema.
Propósito	Crear un nuevo usuario y definir perfil de este en el sistema de farmacia.

Resumen	El usuario administrador requiere crear una cuenta de usuario para el sistema de farmacia y asignarle un perfil para las tareas que realizará dicho usuario.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso de los datos del nuevo usuario.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción	

<p>“Ingresar Nuevo Usuario”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.</p>
	<p>3. El sistema despliega en pantalla el formulario para ingreso de los datos del nuevo usuario.</p>
<p>4. El usuario administrador completa el formulario con los datos requeridos, asigna un perfil, un nombre de usuario y contraseña al nuevo usuario y graba los datos.</p>	
	<p>5. El sistema verifica si los datos proporcionados ya existen previamente.</p>
	<p>6. Los datos son grabados.</p>
	<p>7. Despliegue en pantalla de mensaje indicando que los datos fueron ingresados, se ha creado la cuenta de nuevo usuario y finaliza caso de uso.</p>
<p>Cursos alternos:</p>	
	<p>2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación.</p>
	<p>5a.- Los datos proporcionados ya existen en el sistema y son los mismos, por lo que se</p>

	envía mensaje al usuario informándole que el usuario ya existe. No se graban los datos y finaliza el caso de uso.
	5b.- Los datos proporcionados difieren en nombre completo y/o contraseña de los que ya existen en el sistema. Mediante mensaje en pantalla se le solicita al usuario confirmar la modificación.
5c.- El usuario confirma la modificación de los datos.	
	5d.- Se graban los datos modificados.

Caso de uso N° 4.	Ingresar Datos Farmacia General
Actores	Usuario Administrador del sistema.
Propósito	Inicializar los datos de farmacia general para habilitar el uso del sistema de farmacia para los usuarios autorizados.
	El usuario administrador ingresa los datos de la farmacia general, ésta queda habilitada

Resumen	y el sistema puede operar las funcionalidades del sistema.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso de los datos de farmacia general.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Datos Farmacia General”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de

	administrador.
	3. El sistema despliega en pantalla el formulario para ingreso de los datos de la farmacia general.
4. El usuario administrador completa el formulario con los datos requeridos y solicita grabarlos.	
	5. El sistema verifica si los datos proporcionados ya existen previamente.
	6. Los datos son grabados.
	7. Despliegue en pantalla de mensaje indicando que los datos fueron ingresados y que se ha registrado la farmacia general exitosamente.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación.
	5a.- Los datos proporcionados ya existen en el sistema, por lo que se envía mensaje al usuario informándole del error y que la farmacia general ya existe. No se graban los datos y finaliza el caso de uso.

Caso de uso N° 5.	Ingresar Datos Farmacia Crónicos
Actores	Usuario Administrador del sistema.
Propósito	Inicializar los datos de farmacia crónicos para habilitar el uso del sistema de farmacia para los usuarios autorizados.
Resumen	El usuario administrador ingresa los datos de la farmacia crónicos, está queda habilitada y el sistema puede operar las funcionalidades asociadas al tipo de farmacia.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso de los datos de farmacia crónicos.
	Primario.

Tipo	
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Datos Farmacia Crónicos”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario para ingreso de los datos de la farmacia crónicos.
4. El usuario administrador completa el formulario con los datos requeridos y solicita grabarlos.	
	5. El sistema verifica si los datos proporcionados ya existen previamente.
	6. Los datos son grabados.
	7. Despliegue en pantalla de mensaje

	indicando que los datos fueron ingresados y que se ha registrado la farmacia crónicos exitosamente.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación.
	5a.- Los datos proporcionados ya existen en el sistema, por lo que se envía mensaje al usuario informándole del error y que la farmacia crónicos ya existe. No se graban los datos y finaliza el caso de uso.

Caso de uso N° 6.	Ingresar Datos Farmacia Morbilidad
Actores	Usuario Administrador del sistema.
Propósito	Inicializar los datos de farmacia morbilidad para habilitar el uso del sistema de farmacia para los usuarios autorizados.

Resumen	El usuario administrador ingresa los datos de la farmacia morbilidad, está queda habilitada y el sistema puede operar las funcionalidades asociadas al tipo de farmacia.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso de los datos de farmacia morbilidad.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.

<p>1. El usuario elige y ejecuta la opción “Ingresar Datos Farmacia Morbilidad”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.</p>
	<p>3. El sistema despliega en pantalla el formulario para ingreso de los datos de la farmacia morbilidad.</p>
<p>4. El usuario administrador completa el formulario con los datos requeridos y solicita grabarlos.</p>	
	<p>5. El sistema verifica si los datos proporcionados ya existen previamente.</p>
	<p>6. Los datos son grabados.</p>
	<p>7. Despliegue en pantalla de mensaje indicando que los datos fueron ingresados y que se ha registrado la farmacia morbilidad exitosamente.</p>
<p>Cursos alternos:</p>	
	<p>2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación.</p>
	<p>5a.- Los datos proporcionados ya existen en el sistema, por lo que se envía mensaje al usuario informándole del error y que la</p>

	farmacia morbilidad ya existe. No se graban los datos y finaliza el caso de uso.
--	--

Caso de uso N° 7.	Ingresar Datos Farmacia SAPU
Actores	Usuario Administrador del sistema.
Propósito	Inicializar los datos de farmacia SAPU, para habilitar las operaciones de traspaso de medicamentos hacia ella.
Resumen	El usuario administrador ingresa los datos de la farmacia SAPU, está queda habilitada y el sistema puede operar las funcionalidades de traspaso o devolución de medicamentos hacia ella.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de

	ingreso de los datos de farmacia SAPU.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Datos Farmacia SAPU”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario para ingreso de los datos de la farmacia SAPU.
4. El usuario administrador completa el formulario con los datos requeridos y solicita grabarlos.	

	5. El sistema verifica si los datos proporcionados ya existen previamente.
	6. Los datos son grabados.
	7. Despliegue en pantalla de mensaje indicando que los datos fueron ingresados y que se ha registrado la farmacia SAPU exitosamente.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación.
	5a.- Los datos proporcionados ya existen en el sistema, por lo que se envía mensaje al usuario informándole del error, que la farmacia SAPU ya existe y que la operación no se ha ejecutado. No se graban los datos

Caso de uso N° 8.	Ingresar Datos Farmacia Externa
Actores	Usuario Administrador del sistema.
Propósito	Inicializar los datos de farmacia externa (farmacia de posta rural o de otra institución) para habilitar las operaciones de despacho de medicamentos hacia ella.
Resumen	El usuario administrador ingresa los datos de la farmacia externa, está queda habilitada y el sistema puede operar las funcionalidades de traspaso o devolución de medicamentos hacia ella.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso de los datos de farmacia externa.
	Primario.

Tipo	
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Datos Farmacia Externa”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario para ingreso de los datos de la farmacia externa.
4. El usuario administrador completa el formulario con los datos requeridos y solicita grabarlos.	
	5. El sistema verifica si los datos proporcionados ya existen previamente.
	6. Los datos son grabados.
	7. Despliegue en pantalla de mensaje

	indicando que los datos fueron ingresados y que se ha registrado la farmacia externa exitosamente.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación.
	5a.- Los datos proporcionados ya existen en el sistema, por lo que se envía mensaje al usuario informándole del error, que la farmacia externa ya existe y que la operación no se ha ejecutado. No se graban los datos.

Caso de uso N° 9.	Ingresar Paciente Crónico.
Actores	Técnico de farmacia como Usuario de Farmacia Crónicos, Jefe de farmacia como Administrador del sistema.
Propósito	Incorporar un nuevo paciente crónico al sistema y generarle tarjeta de dispensación.

Resumen	El Jefe de farmacia como Administrador del sistema o un técnico de farmacia como usuario del sistema, registra un nuevo paciente crónico al que se le crea una tarjeta de dispensación de farmacoterapia prolongada.
Precondición	Se ha iniciado sesión con perfil administrador o usuario normal en el sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso del nuevo paciente crónico.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.

1. El usuario elige y ejecuta la opción “Ingresar Paciente Crónico”.	
	2. El sistema pide ingresar número de ficha del paciente o RUN.
3. El usuario ingresa N° de ficha o RUN del paciente crónico.	
	4. Verifica la existencia del paciente a través del N° de ficha o RUN.
	5. El sistema despliega en pantalla el formulario para ingreso de los datos del paciente crónico.
6. El usuario completa el formulario con los datos requeridos y solicita grabarlos.	
	7. Los datos del paciente son grabados.
	8. Creación de tarjeta de dispensación del paciente con n° de ficha y programa de atención.
	9. Despliegue en pantalla de mensaje indicando que los datos del paciente crónico fueron registrados.
Cursos alternos:	
	4a.- Si el usuario ingresó el RUN del paciente y es incorrecto se envía mensaje de error y se vuelve al N° 2.

	4b.- Si el paciente ya existe en el sistema se envía mensaje al usuario informando que el paciente crónico ya está registrado previamente y se vuelve a la pantalla principal del sistema.
	7a.- Si de los datos del paciente el RUN ingresado es erróneo se avisa al usuario para que lo corrija o se marca como dato pendiente en el registro y se graba el resto de los datos.

Caso de uso N° 10.	Ingresar Paciente Morbilidad.
Actores	Técnico de farmacia como Usuario de Farmacia Morbilidad o Supervisor, Jefe de farmacia como Administrador del sistema.
Propósito	Incorporar un nuevo paciente de morbilidad al sistema.
Resumen	El Jefe de farmacia como Administrador del sistema o un técnico de farmacia como usuario normal del sistema, registra un nuevo paciente de morbilidad.

Precondición	Se ha iniciado sesión con perfil administrador o usuario normal en el sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso del nuevo paciente de morbilidad.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Paciente Morbilidad”.	
	2. El sistema pide ingresar número de ficha del paciente o RUN.
3. El usuario ingresa N° de ficha o RUN del paciente crónico.	

	4. Verifica la existencia del paciente a través del N° de ficha o RUN.
	5. El sistema despliega en pantalla el formulario para ingreso de los datos del paciente de morbilidad.
6. El usuario completa el formulario con los datos requeridos y solicita grabarlos.	
	7. Se graban los datos del paciente.
	8. Despliegue en pantalla de mensaje indicando que los datos del paciente de morbilidad fueron registrados.
Cursos alternos:	
	4a.- Si el RUN del paciente o N° de ficha es incorrecto, se envía mensaje de error y se vuelve al N° 2.
	4b.- Si el RUN y/o N° de ficha del paciente ya existe en el sistema, se envía mensaje al usuario informando el error y se vuelve a la pantalla principal del sistema.

Caso de uso N° 11.	Ingresar Profesional.
Actores	Técnico de farmacia como Usuario de alguna farmacia, Jefe de farmacia como Administrador del sistema.
Propósito	Incorporar los datos de un nuevo Profesional de la salud al sistema de control de farmacia.
Resumen	El Jefe de farmacia como Administrador del sistema o un técnico de farmacia como usuario normal del sistema, registra los datos de un nuevo Profesional de la salud, que atiende a los pacientes y les prescribe medicamentos.
Precondición	Se ha iniciado sesión con perfil administrador o usuario normal en el sistema de control de farmacia.
Postcondición	Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso de los datos del nuevo Profesional.
	Primario.

Tipo	
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Profesional”.	
	2. El sistema pide ingresar el RUN del Profesional.
3. El usuario ingresa el RUN del Profesional.	
	4. Verifica la existencia del Profesional a través del RUN.
	5. El sistema despliega en pantalla el formulario para ingreso de los datos del Profesional.
6. El usuario completa el formulario con los datos requeridos y solicita grabarlos.	
	7. Se graban los datos del Profesional.

	8. Despliegue en pantalla de mensaje indicando que los datos del Profesional fueron registrados.
Cursos alternos:	
2a.- El usuario cancela la operación.	
	4a.- Si el usuario ingresó RUN incorrecto se envía mensaje de error y se vuelve al N° 2.
	4b.- Si el Profesional ya existe en el sistema, se envía mensaje al usuario informando que ya está registrado previamente y se vuelve a la pantalla principal del sistema.
	7a.- Si el RUN ingresado es erróneo se avisa al usuario para que lo corrija y se vuelve al N° 2.

Caso de uso N° 12.	Ingresar o Modificar Grupo Etario.
Actores	Usuario Administrador del sistema.

Propósito	Incorporar un nuevo grupo etario que no estaba registrado anteriormente en el sistema de farmacia o modificar un grupo existente.
Resumen	El usuario administrador registra un nuevo grupo etario para clasificar los pacientes o requiere modificar los rangos de edad de alguno de los grupos ya existentes.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Grupo etario creado o modificado queda grabado y se muestra en pantalla mensaje comunicando el resultado de la operación.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.

1. El usuario elige y ejecuta la opción “Ingresar o Modificar Grupo Etario”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el listado de grupos etarios existentes.
4. El usuario elige un grupo etario para modificarlo o escoge crear uno nuevo.	5. El sistema muestra el formulario con los detalles modificables del grupo etario elegido o muestra un formulario vacío para ingresar nuevo grupo etario.
6. El usuario administrador modifica la edad mínima y/o máxima del grupo etario existente en el formulario o ingresa los datos pedidos en caso de ser nuevo grupo etario. En ambos casos solicita grabar estos datos.	
	7. El sistema verifica si los datos proporcionados son válidos.
	8. Los datos son grabados.
	9. Despliegue en pantalla de mensaje indicando que se ha creado o modificado el grupo etario exitosamente.
Cursos alternos:	

	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación y luego vuelve a la pantalla principal del sistema.
3a.- El usuario puede cancelar la operación en cualquier momento mientras tiene el control y aún no se haya grabado algo.	
	7a.- Existe un error en la edad mínima y/o máxima del grupo, por lo que se envía mensaje al usuario informándole del error para que lo corrija. No se graban los datos y regresa al N° 5.

Caso de uso N° 13.	Ingresar o Modificar Programa de Salud
Actores	Usuario Administrador del sistema.
Propósito	Incorporar un programa de salud que no estaba registrado anteriormente en el sistema de control de farmacia o modificar uno existente

<p>Resumen</p>	<p>El usuario administrador registra un nuevo programa de salud dirigido a un grupo etario y tipo específico de pacientes o modifica los detalles de un programa existente.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso del nuevo programa de salud o modificación de alguno.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>
<p>1. El usuario elige y ejecuta la opción “Ingresar o Modificar Programa de Salud”.</p>	

	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario para ingreso de los datos de programa de salud y el listado de los programas existentes.
4. El usuario administrador completa el formulario con la información requerida de nombre del programa, grupo etario beneficiario, género de los beneficiarios y solicita grabar estos datos o elije un programa de salud para modificarlo, de entre los existentes que se muestran.	5. El sistema verifica si los datos proporcionados ya existen previamente y si son consistentes.
	6. Los datos son grabados.
	7. Despliegue en pantalla de mensaje indicando que los datos fueron ingresados y que se ha registrado el nuevo programa de salud exitosamente o se ha modificado exitosamente el que fue indicado.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación y vuelve a la pantalla principal del sistema de control de farmacia.

4a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
	5a.- El usuario ha elegido ingresar un nuevo programa y los datos proporcionados ya existen en el sistema, por lo que se envía mensaje al usuario informándole del error y que el programa de salud ya existe. No se graban los datos.
	5b.- El usuario ha elegido modificar un programa de salud existente y los datos que ha modificado son inconsistentes por lo que se muestra un mensaje de error al usuario para que corrija los errores y vuelva a intentar grabar las modificaciones.

Caso de uso N° 14.	Enviar Pedido de Insumos y Medicamentos a Abastecimiento.
Actores	Usuario Administrador del sistema.
Propósito	Enviar un pedido de insumos y/o medicamentos al departamento de abastecimiento para su posterior recepción y entrada al stock de farmacia general.

<p>Resumen</p>	<p>El usuario administrador confecciona un listado de medicamentos e insumos que serán solicitados a abastecimiento y luego envía este listado a través del sistema para que el personal de abastecimiento lo reciba y proceda a chequear y despachar los artículos hacia farmacia general.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Se muestra en pantalla mensaje comunicando el resultado de la operación de envío del nuevo pedido de insumos y medicamentos a abastecimiento.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>

<p>1. El usuario elige y ejecuta la opción “Ingresar Pedido”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.</p>
	<p>3. El sistema despliega en pantalla el formulario para ingreso de nuevo pedido que será enviado a abastecimiento.</p>
<p>4. El usuario administrador completa el formulario con la información requerida de artículo y cantidad, y solicita grabar el pedido en el sistema dejándolo pendiente de enviar o solicita enviarlo al departamento de abastecimiento.</p>	<p>5. El sistema verifica si los datos proporcionados son consistentes (cantidades).</p>
	<p>6. Los datos son grabados y enviados al departamento de abastecimiento.</p>
	<p>7. Despliegue en pantalla de mensaje indicando que el pedido fue grabado y/o que se enviado exitosamente al departamento de abastecimiento.</p>
<p>Cursos alternos:</p>	
	<p>2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación y vuelve a la pantalla principal del sistema de control de farmacia.</p>

4a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
	5a.- Las cantidades ingresadas para uno o más artículos excede la cantidad definida como stock máximo o se ha pedido cero unidades, por lo que se envía mensaje de error al usuario indicando el o los artículos en cuestión para que corrija el formulario.

Caso de uso N° 15.	Recibir Pedido de Insumos y Medicamentos.
Actores	Usuario Administrador del sistema o usuario Supervisor.
Propósito	Recibir un pedido de insumos y/o medicamentos desde el departamento de abastecimiento para entrada al stock de farmacia general.
Resumen	El usuario correspondiente recibe un formulario electrónico con el listado de medicamentos e insumos que fueron despachados desde abastecimiento y que están asociados a un pedido enviado previamente desde farmacia general. El

	<p>formulario de recepción debe ser aceptado por la Jefa de Farmacia o Funcionaria autorizada para el efecto, confirmando así la recepción física de los artículos, para lo cual debe chequear el pedido real que es recibido en farmacia.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia o como usuario supervisor.</p>
<p>Postcondición</p>	<p>Se muestra en pantalla mensaje comunicando el resultado de la operación de recepción del nuevo pedido de insumos y medicamentos y se envía confirmación de la recepción (conforme o con reparos) al departamento de abastecimiento a través de formulario electrónico.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>

1. El usuario elige y ejecuta la opción “Recibir Pedido”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador o supervisor.
	3. El sistema despliega en pantalla el formulario de recepción de pedidos que muestra los pedidos despachados desde abastecimiento hacia farmacia que están en estado pendiente de confirmación por parte de farmacia.
4. El usuario administrador elige un pedido del listado para proceder a revisarlo realizando el chequeo físico de los artículos recibidos.	5. El sistema despliega en pantalla el detalle y cantidad de los artículos que están siendo recibidos en farmacia.
6. Previo chequeo físico de los artículos recibidos, el usuario confirma indicando “recibido conforme” o “recibido con reparos”, o bien puede dejar el pedido en estado “pendiente” de confirmar.	7. El detalle y resultado de la operación es grabado en el sistema y a la vez enviado al departamento de abastecimiento a través del formulario en línea.
	8. Despliegue en pantalla de mensaje indicando que la operación de recepción fue finalizada.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no

	tiene privilegios de administrador o supervisor, por lo que el sistema muestra mensaje de error indicando la situación y vuelve a la pantalla principal del sistema de control de farmacia.
4a y 6a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
6b.- Si el usuario deja el pedido en estado de “recibido con reparos”, deberá indicar en cada caso los detalles o reparos.	

Caso de uso N° 16.	Ingresar Pedido de Insumo o Medicamento Especial.
Actores	Usuario Administrador del sistema.
Propósito	Enviar un pedido a abastecimiento para adquirir un insumo o medicamento especial, no disponible normalmente en el stock de farmacia, y que es solicitado por algún profesional para tratar alguna patología específica en pacientes.
	El Jefe de Farmacia como usuario administrador, ingresa un pedido especial de

Resumen	insumo o medicamento para enviarlo a abastecimiento, para lo que requiere previamente una solicitud escrita y autorizada por parte de la Dirección del consultorio.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Pedido especial enviado a abastecimiento. Se muestra en pantalla mensaje comunicando el resultado de la operación de envío del pedido especial.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Pedido Especial”.	

	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario para ingreso de pedido especial, que será enviado a abastecimiento.
4. El usuario administrador completa el formulario con la información requerida de RUN del Profesional que solicita, número autorización de Dirección, artículo y cantidad, y envía el pedido al departamento de abastecimiento.	5. El sistema verifica si los datos proporcionados son válidos.
	6. Los datos son grabados y enviados al departamento de abastecimiento.
	7. Despliegue en pantalla de mensaje indicando que el pedido especial fue grabado y enviado exitosamente al departamento de abastecimiento.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación y vuelve a la pantalla principal del sistema de control de farmacia.
4a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.

	5a.- Las cantidad pedida es cero unidades o falta algún dato en el formulario, por lo que se envía mensaje de error al usuario para que lo corrija y se vuelve a mostrar el formulario de ingreso de pedido especial.
--	---

Caso de uso N° 17.	Ingresar Identificación de Nuevo Medicamento o Insumo.
Actores	Usuario Administrador del sistema o Supervisor.
Propósito	Incorporar un nuevo medicamento o insumo que no estaba registrado anteriormente en el inventario de la farmacia y para el cual no había código conocido.
Resumen	El usuario administrador o supervisor registra un nuevo medicamento o insumo en el sistema habilitando el artículo para su inclusión en pedidos y en el stock de farmacia. Además se crea una bincard asociada al artículo.
Precondición	El usuario ha iniciado sesión con privilegios de administrador o supervisor del sistema de control de farmacia.

Postcondición	Insumo o medicamento ingresado al sistema; bincard creada para el artículo y se muestra en pantalla mensaje comunicando el resultado de la operación de ingreso del nuevo artículo.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Ingresar Identificación de Nuevo Medicamento o Insumo”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador o supervisor.
	3. El sistema despliega en pantalla el formulario para ingreso de los datos de

	identificación del nuevo artículo.
4. El usuario administrador completa el formulario con los datos requeridos de código genérico, nombre de artículo, stock crítico, stock mínimo, stock máximo, cantidad inicial y solicita grabarlos.	
	5. El sistema verifica si el artículo ya existe previamente y si los datos son válidos.
	6. Los datos son grabados.
	7. Despliegue en pantalla de mensaje indicando que el nuevo artículo fue guardado exitosamente.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador ni supervisor por lo que el sistema muestra mensaje de error indicando la situación.
	5a.- El artículo ya existe en el sistema, por lo que se envía mensaje al usuario informándole del error. No se graban los datos y se regresa al formulario de ingreso del nuevo artículo.
	5b.- Los datos de stock no son coherentes entre sí o por si solos, por lo que se envía mensaje de error al usuario indicando lo que debe corregir.

<p>Caso de uso N° 18.</p>	<p>Distribuir Insumos y Medicamentos a Farmacias del CVP.</p>
<p>Actores</p>	<p>Usuario Administrador del Sistema.</p>
<p>Propósito</p>	<p>Los artículos de un pedido procedente de abastecimiento que son recibidos en farmacia general, son repartidos entre las farmacias internas del consultorio: Crónicos, Morbilidad y SAPU.</p>
<p>Resumen</p>	<p>El usuario administrador distribuye los artículos recibidos en farmacia general, hacia las distintas farmacias internas del consultorio, aumentando el stock de los artículos según corresponda en el inventario de cada farmacia, lo que actualiza además las bincards asociadas a los artículos.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Artículos distribuidos quedan sumados al stock de las farmacias internas del CVP. Las bincards de cada artículo recibido quedan actualizadas. Se muestra en pantalla</p>

	mensaje comunicando el resultado de la operación de distribución.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Distribuir Insumos y Medicamentos”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario para distribución de los artículos recibidos en farmacia general.
4. El usuario administrador distribuye las cantidades de cada artículo de farmacia general, entre las farmacias internas y confirma la operación al repartir la totalidad de los artículos recibidos desde	

<p>abastecimiento o solicita grabar la distribución para modificarla o confirmarla más tarde.</p>	
	<p>5. El sistema verifica si todos los artículos recibidos, fueron asignados al stock de alguna farmacia interna.</p>
	<p>6. Los stock de cada farmacia son actualizados.</p>
	<p>7. Despliegue en pantalla de mensaje indicando que la distribución de artículos esta terminada.</p>
<p>Cursos alternos:</p>	
	<p>2b. El usuario que solicita la ejecución no tiene privilegios de administrador ni supervisor por lo que el sistema muestra mensaje de error indicando la situación.</p>
<p>4a.- El usuario cancela la operación.</p>	<p>4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.</p>
	<p>5a.- El usuario confirma la operación de distribución pero existen artículos en farmacia general que no han sido asignados a ninguna farmacia interna, por lo que se muestra mensaje de error al usuario pidiéndole que asigne la totalidad de los artículos de farmacia general a las farmacias internas o que grabe la distribución parcial. Se regresa al formulario de distribución de</p>

	artículos.
	5b.- El usuario solicitó grabar la distribución parcial para modificarla o confirmarla más tarde. El sistema graba la distribución y vuelve a la pantalla principal del sistema de control de farmacia.
	5c.- La cantidad distribuida de uno o más artículos es mayor que la cantidad existente a repartir, por lo que se envía mensaje de error al usuario indicando lo que debe corregir.

Caso de uso N° 19.	Registrar Despacho de Insumos y Medicamentos a Farmacia Externa.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
Propósito	Registrar un despacho de artículos hecho desde farmacia del consultorio a farmacias externas como postas, centros comunitarios de salud familiar o farmacias de otras instituciones.

<p>Resumen</p>	<p>El usuario administrador confecciona una lista de insumos y/o medicamentos que serán registrados como despacho con destino a una farmacia externa al consultorio. Esto lo realiza rebajando artículos del stock de una o más farmacias para formar el despacho.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Artículos despachados quedan rebajados del stock de las farmacias internas del CVP. Las bincards de cada artículo despachado quedan actualizadas. Se muestra en pantalla mensaje comunicando el resultado de la operación de despacho.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>

1. El usuario elige y ejecuta la opción “Registrar Despacho a Farmacia Externa”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario de despacho de insumos y medicamentos a farmacias externas.
4. El usuario administrador indica los artículos y cantidades que se despachará, el nombre de la farmacia externa de destino y solicita grabar el despacho para confirmarlo o modificarlo más tarde, o solicita confirmarlo de inmediato para su envío físico.	
	5. El sistema verifica si hay stock suficiente a despachar, en las farmacias internas a las que se les descontarán los productos.
	6. El stock de cada farmacia interna es actualizado; de igual forma se actualizan las bincard asociadas.
	7. Se genera el despacho con un folio de operación, fecha, hora, destino, persona responsable, detalle de productos y cantidades.

	8. Despliegue en pantalla de mensaje indicando que el despacho se generó exitosamente y mostrando detalles de la operación.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación. Se regresa a pantalla principal del sistema de control de farmacia.
4a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
	5a.- No hay stock suficiente de todos o algunos artículos, por lo que se muestra mensaje de error al usuario indicando cuales artículos no pueden ser incluidos en el despacho para corregir. Se regresa al formulario de despacho.

Caso de uso N° 20.	Dar de Baja Insumos o Medicamentos.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).

Propósito	Reducir unidades de uno o más artículos de alguna farmacia indicando los motivos.
Resumen	El usuario administrador reduce la cantidad de uno o más insumos o medicamentos en existencia en alguna farmacia por mermas o pérdidas.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Los artículos dados de baja quedan restados del stock de las farmacias correspondientes y las bincards quedan actualizadas. Se muestra en pantalla mensaje comunicando el resultado de la operación dar de baja.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	

Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Dar de Baja Insumos o Medicamentos”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema despliega en pantalla el formulario para dar de baja insumos o medicamentos.
4. El usuario administrador selecciona la farmacia donde se producen las bajas, indica el código o nombre del artículo y cantidad que se dará de baja, el motivo (merma o pérdida), y agrega estos datos a la lista.	
<i>Los pasos 4 y 5 se repiten hasta que el usuario confirma la operación en el N° 6.</i>	5. El sistema verifica que la cantidad que se dará de baja no supera la cantidad existente del artículo; también verifica que se haya especificado la farmacia donde se aplican las bajas y actualiza la lista en pantalla, de los artículos.
6. El usuario confirma la operación de dar	

de baja los artículos indicados en N° 4.	
	7. El stock de la farmacia que corresponda es actualizado; de igual forma se actualizan las bincard asociadas.
	8. Se asigna un folio a la operación de baja, fecha, hora, motivo, persona responsable, detalle de productos y cantidades. Todo esto queda grabado en el Sistema.
	9. Despliegue en pantalla de mensaje indicando que la operación de baja se realizó correctamente, mostrando detalles de la operación.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación. Se regresa a pantalla principal del sistema de control de farmacia.
4a – 6a.- El usuario cancela la operación.	4b-6b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
	5a.- La cantidad de uno o más artículos que se darán de baja, supera la cantidad existente en la farmacia correspondiente, por lo que se muestra mensaje al usuario indicándole el error para que lo corrija. Se regresa al formulario de baja de artículos.

	5b.- No se ha especificado la farmacia donde se aplican las bajas, por lo que se muestra mensaje de error al usuario indicándole que debe establecer a que farmacia corresponde la operación.
--	---

Caso de uso N° 21.	Devolución de Insumos o Medicamentos.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema) y Sistema de Abastecimiento.
Propósito	Registrar los insumos o medicamentos que son devueltos desde farmacia al departamento de abastecimiento, por encontrarse vencidos o por no estar en condiciones de ser consumidos por los pacientes.
Resumen	El usuario administrador devuelve una cantidad de artículos desde la farmacia al departamento de abastecimiento, reduciendo el stock de dichos artículos en farmacia, registrando la operación y los motivos (vencimiento de artículos, mal estado de conservación).

<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Los artículos dados devueltos quedan restados del stock de las farmacias correspondientes y las respectivas bincards quedan actualizadas. Se muestra en pantalla mensaje comunicando el resultado de la operación devolución.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>
<p>1. El usuario elige y ejecuta la opción “Devolución de Insumos o Medicamentos”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de Administrador.</p>

	<p>3. El sistema despliega en pantalla el formulario para devolución de insumos o medicamentos.</p>
<p>4. El usuario administrador indica los artículos y cantidades que se devolverán, el motivo (vencimiento, deterioro, orden ministerial, etc.), el nombre de la farmacia donde se produce la devolución, el n° de pedido y fecha con que se ingresaron los artículos, Solicitando al sistema confirmar la operación y actualizar el stock.</p>	
	<p>5. El sistema verifica que la cantidad que se devolverá no supera la cantidad existente del artículo; también verifica que se haya especificado la farmacia donde se aplica la devolución y verifica que el N° de pedido y fecha con que ingresaron los artículos, esté registrado.</p>
	<p>6. El stock de la farmacia que corresponda es actualizado; de igual forma se actualizan las bincards asociadas.</p>
	<p>7. Se asigna un folio a la operación de devolución y se registra la fecha, hora, motivo, persona responsable, detalle de productos y cantidades. Todo esto queda grabado en el Sistema de Control de Farmacia y se envía vía intranet al Sistema de Abastecimiento para su posterior</p>

	confirmación.
	8. Despliegue en pantalla de mensaje indicando que la operación de devolución se realizó correctamente y que queda pendiente confirmación por parte del departamento de Abastecimiento.
	9. El sistema recibe confirmación de la devolución por parte del Sistema de Abastecimiento, a través del cambio de estado de la operación en la tabla de la base de datos que comparten ambos sistemas.
Cursos alternos:	
	2b. El usuario que solicita la ejecución no tiene privilegios de administrador por lo que el sistema muestra mensaje de error indicando la situación. Se regresa a pantalla principal del sistema de control de farmacia.
4a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
	5a.- La cantidad de uno o más artículos que se devolverán, supera la cantidad existente en la farmacia correspondiente, por lo que se muestra mensaje al usuario indicándole el error para que lo corrija. Se regresa al formulario de devolución de insumos o medicamentos.

	5b.- No se ha especificado la farmacia donde se aplica la devolución, por lo que se muestra mensaje de error al usuario indicándole que debe establecer a que farmacia corresponde la operación.
	5c.- La fecha y N° de pedido asociado a los artículos que se devolverán, no existe en el sistema, por lo que se avisa al usuario del error para que corrija.

Caso de uso N° 22.	Dispensar Receta desde Farmacia Morbilidad.
Actores	Usuario de Farmacia Morbilidad o Usuarios Supervisor y Administrador del Sistema.
Propósito	Registrar los datos de la receta presentada por un paciente de morbilidad, al momento del retiro de los medicamentos o insumos desde la farmacia de morbilidad.
Resumen	El usuario del sistema ingresa una receta detallando los artículos solicitados para dispensación. El sistema verifica la inscripción del paciente, el stock de artículos, luego graba la operación,

	actualizando el stock, el registro de dispensaciones y si es necesario los datos del paciente.
Precondición	El usuario ha iniciado sesión con privilegios de usuario de la farmacia de morbilidad, o supervisor, o administrador del sistema de control de farmacia.
Postcondición	Los artículos dispensados quedan restados del stock de la farmacia de morbilidad y las respectivas bincards quedan actualizadas. Además queda registrada la dispensación con detalles. Se muestra en pantalla mensaje comunicando el resultado de la operación de dispensación.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción	

<p>“Dispensar Receta Morbilidad”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de usuario de la farmacia morbilidad, o supervisor, o administrador del sistema.</p>
	<p>3. El sistema despliega en pantalla el formulario para dispensar insumos o medicamentos desde farmacia morbilidad.</p>
<p>4. El usuario ingresa N° de receta y N° de ficha del paciente, RUN del Profesional que prescribe y diagnóstico.</p>	<p>5. El sistema verifica que la receta no existe.</p> <p>6. El sistema verifica que el paciente esté inscrito y activo.</p>
	<p>7. El sistema activa la sección del formulario para ingreso de las prescripciones contenidas en la receta.</p>
<p>8. El usuario ingresa el, código del medicamento o insumo, días de tratamiento, dosis y frecuencia de administración en horas.</p> <p><i>Se repiten los pasos 8 al 13, hasta completar todas las prescripciones contenidas en la receta.</i></p>	<p>9. El sistema calcula la cantidad a entregar en base a los días, dosis y frecuencia indicada.</p> <p>10. El sistema verifica que la cantidad a entregar no supere la cantidad máxima permitida por paciente.</p> <p>11. El sistema verifica que la cantidad a entregar exista en stock.</p> <p>12. El stock de la farmacia morbilidad es actualizado; de igual forma se actualizan las bincards asociadas.</p>

	13. El sistema presenta en pantalla la lista actualizada de prescripciones de la receta que se está ingresando.
	14. Se graba la operación de dispensación con la fecha, hora, N° Receta, RUN paciente, RUN Profesional, código artículo dispensado, días tratamiento, dosis, frecuencia, cantidad entregada, usuario responsable.
	15. Despliegue en pantalla de mensaje indicando que la operación de dispensación se realizó correctamente.
Cursos alternos:	
	2a. El usuario que solicita la ejecución no tiene privilegios de usuario de farmacia morbilidad, o supervisor, o administrador, por lo que el sistema muestra mensaje de error indicando la situación. Se regresa a pantalla principal del sistema de control de farmacia.
4a-. El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
	5a.- El sistema detecta que la receta con el N° que se ha ingresado ya existe, por lo que rechaza el ingreso y envía mensaje al usuario avisando el error correspondiente.

	6a.- El sistema detecta que el paciente no está inscrito en el sistema, por lo que da la opción para inscribirlo como paciente “de paso” y un conjunto de datos básicos, avisando la situación con mensaje al usuario.
6b.- El usuario cancela la operación.	6c.- El sistema regresa al N° 3.
6d.- El usuario ingresa el RUN, nombres, apellidos y dirección del paciente.	6e.- El sistema asigna un N° de ficha temporal al paciente, el cual es mostrado al usuario. El paciente queda inscrito provisoriamente para realizar la dispensación.
	10a.- La cantidad a entregar supera la cantidad permitida por paciente, se da mensaje al usuario para que reduzca los días.
	11a.- La cantidad a entregar supera el stock en la farmacia, se da mensaje al usuario indicándole el error y cuantas unidades disponibles hay del medicamento.

Caso de uso N° 23.	Dispensar Receta desde Farmacia Crónicos.
Actores	Usuario de Farmacia Crónicos o Usuarios Supervisor y Administrador del Sistema.

<p>Propósito</p>	<p>Registrar los medicamentos o insumos entregados a un Paciente crónico, en la tarjeta de dispensación personal, al momento del retiro de los artículos desde la farmacia crónicos.</p>
<p>Resumen</p>	<p>El usuario del sistema ingresa la identificación del paciente crónico y el detalle de los artículos solicitados para entrega, de acuerdo a la tarjeta de dispensación correspondiente. El sistema verifica la inscripción del paciente, el stock, luego graba la operación, actualizando el stock y la tarjeta de dispensación del paciente.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de usuario de la farmacia crónicos, o supervisor, o administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Los artículos dispensados quedan restados del stock de la farmacia crónicos y las respectivas bincards quedan actualizadas. Además queda registrada la entrega con detalles en la tarjeta de dispensación del paciente crónico. Se muestra en pantalla mensaje comunicando el resultado de la operación de dispensación.</p>

<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>
<p>1. El usuario elige y ejecuta la opción “Dispensar Receta Crónicos”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de usuario de la farmacia crónicos, o supervisor, o administrador del sistema.</p>
	<p>3. El sistema despliega en pantalla el formulario para dispensar insumos o medicamentos desde farmacia crónicos y solicita ingreso de RUN o N° de ficha del paciente.</p>
<p>4. El usuario ingresa N° de ficha o RUN del paciente.</p>	<p>5. El sistema verifica que el RUN y/o N° de ficha sea correcto, y que el paciente esté inscrito y activo.</p>

	6. El sistema verifica que el paciente sea crónico.
	7. El sistema muestra la tarjeta electrónica de dispensación del paciente con las entregas anteriores.
	8. El sistema solicita ingreso de número de días para próxima entrega.
9. El usuario ingresa el número de días para la próxima entrega.	10. El sistema calcula y muestra en el formulario la fecha para la próxima entrega.
	11. El sistema solicita ingreso del programa de salud al que se acoge el paciente crónico.
12. El usuario selecciona el programa de salud.	13. El sistema verifica que el programa ingresado corresponda al género y edad del paciente.
	14. El sistema genera internamente lista con diagnósticos asociados al programa de salud seleccionado.
	15. El sistema solicita seleccionar el diagnóstico, que se asociará con el grupo de medicamentos que se ingresarán.
16. El usuario selecciona el diagnóstico, que será asociado al grupo de medicamentos a ingresar.	17. El sistema muestra en pantalla si la patología diagnosticada está incluida o no en el plan de Garantías Explícitas de Salud (GES).
19. El usuario ingresa el código del	18. El sistema solicita ingresar los datos del

<p>medicamento o insumo, tipo (controlado o no), días de tratamiento, dosis, frecuencia de administración en horas.</p> <p><i>Se repiten los pasos 18 al 24, hasta que el usuario cambie el programa de salud o diagnóstico actuales (N° 25) o ejecute la acción del N° 27.</i></p>	<p>insumo o medicamento a dispensar.</p> <p>20. El sistema calcula la cantidad a entregar en base a los días, dosis y frecuencia indicada.</p> <p>21. El sistema verifica que la cantidad a entregar no supere la cantidad máxima permitida por paciente.</p> <p>22. El sistema verifica que la cantidad a entregar exista en stock.</p> <p>23. El stock de la farmacia crónicos es actualizado; de igual forma se actualizan las bincards asociadas.</p> <p>24. El sistema presenta en pantalla, la lista detallada actualizada, del grupo de medicamentos que se está ingresando.</p>
<p>25. El usuario cambia el programa de salud o el diagnóstico, que estaba establecido hasta ahora para el grupo de medicamentos listados, que se esta ingresando a la tarjeta de dispensación.</p>	<p>26. El sistema regresa al N° 11 si se cambió el programa de salud, o regresa al N° 15 si se cambió el diagnóstico. Los medicamentos que se siguen ingresando de aquí en adelante, quedan asociados al programa y/o diagnóstico cambiado.</p>
<p>27. El usuario solicita grabar la tarjeta electrónica de dispensación para finalizar la operación.</p>	<p>28. Se graba la tarjeta electrónica de dispensación con la fecha y hora de entrega, fecha de la próxima entrega, RUN paciente, RUN Profesional, códigos de medicamentos dispensados,</p>

	Programas y diagnósticos asociados, días tratamiento, dosis, frecuencia, cantidad entregada y usuario responsable de la dispensación.
	29. Despliegue en pantalla de mensaje indicando que la operación de dispensación se realizó exitosamente.
Cursos alternos:	
	2a. El usuario que solicita la ejecución no tiene privilegios de usuario de farmacia crónicos, o supervisor, o administrador, por lo que el sistema muestra mensaje de error indicando la situación. Se regresa a pantalla principal del sistema de control de farmacia.
3a-4a-. El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del sistema de control de farmacia.
	5a.- El sistema detecta que el paciente no está inscrito en el sistema o no está activo, o el RUN y/o N° de ficha son incorrectos, por lo que da mensaje al usuario informando el error.
	6a.- El sistema detecta que el paciente está inscrito y activo, pero no es un paciente crónico, por lo que da mensaje al usuario avisando que el paciente es primera vez que retira medicamentos por patología crónica y que será clasificado crónico, una vez que se

	<p>haya finalizado exitosamente, la primera dispensación de medicamentos.</p>
	<p>10a.- El sistema detecta error en el número de días para la próxima entrega, por exceso de días o por error en el formato del número ingresado, por lo que envía mensaje al usuario para que proceda a corregir el valor.</p>
	<p>13a.- El sistema detecta que el programa de salud seleccionado no está dirigido al paciente por no cumplir los requisitos de edad o género, por lo que da mensaje de error al usuario para que elija otro programa de salud.</p>
<p>19a.- Si el medicamento es controlado, el usuario ingresa el N° de receta de la prescripción de dicho medicamento.</p>	
	<p>21a.- La cantidad a entregar supera la cantidad permitida por paciente, se da mensaje al usuario para que reduzca los días.</p>
	<p>22a.- La cantidad a entregar supera el stock en la farmacia, se da mensaje al usuario indicándole el error y cuantas unidades disponibles hay del medicamento, además se establece a “pendiente” de entregar el estado del artículo en la tarjeta de dispensación.</p>

	28a.- Si es primera vez que el paciente retira medicamentos en farmacia crónicos, el sistema modifica los datos personales y clasifica al paciente como crónico.
--	--

Caso de uso N° 24.	Dispensar Receta desde Farmacia SAPU.
Actores	Usuario de Farmacia SAPU o Usuarios Supervisor y Administrador del Sistema.
<i>Este caso de uso es análogo al N° 22, con la diferencia que es realizado en la Farmacia SAPU.</i>	

Caso de uso N° 25.	Generar Informe de Stock, Entrada y Salida de Artículos por Farmacia.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema) o Encargado de Farmacia (Supervisor).

<p>Propósito</p>	<p>Mostrar al usuario un listado del estado del stock, cantidad de entrada y cantidad que ha salido, de cada medicamento e insumo, durante un periodo específico, para una farmacia específica, para revisión del inventario físico de esta última.</p>
<p>Resumen</p>	<p>El usuario solicita generar el informe de stock, entradas y salidas para una de las farmacias del consultorio. El sistema entrega un listado en pantalla con todos los medicamentos y su respectivo stock, cantidad que ha entrado y cantidad que ha sido entregada, durante el periodo que indique el usuario que puede ser un día, un mes, o entre 2 fechas dadas.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de supervisor o administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Informe generado en pantalla puede ser enviado a impresora.</p>
<p>Tipo</p>	<p>Primario.</p>

<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Informe de Stock, Entrada y Salida de Artículos”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador o es el supervisor.
	3. El sistema solicita que el usuario seleccione la farmacia para la cual se generará el informe y el periodo a informar.
4. El usuario selecciona la farmacia y periodo a informar que puede ser un día, un mes, o alguno entre 2 fechas dadas por el usuario.	
	5. El sistema verifica, en caso que se haya dado 2 fechas, que estas sean válidas y coherentes entre sí.
	6. El sistema genera listado de insumos y medicamentos con su respectivo stock y cantidad que ha entrado y salido durante el

	día, o mes o periodo de tiempo que haya indicado el usuario y lo muestra en pantalla.
Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para generar el informe, por lo que informa el error al usuario.
6a.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6b.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 26.	Generar Informe de Stock, Entrada y Salida de Artículos en General.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
Propósito	Mostrar al usuario un listado del estado del stock, cantidad de entrada y cantidad que ha salido, de cada medicamento e insumo, durante un periodo específico, para la farmacia general; esto para revisión del inventario físico general.

<p>Resumen</p>	<p>El usuario solicita generar un informe de stock, entradas y salidas de la farmacia general del consultorio. El sistema entrega un listado en pantalla con todos los medicamentos y su respectivo stock, cantidad que ha entrado y cantidad que ha sido entregada, durante el periodo que indique el usuario que puede ser un día, un mes, o entre 2 fechas dadas. El informe es generado con los datos de todas las farmacias del consultorio (crónicos, morbilidad, SAPU y los despachos a farmacias externas).</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Informe generado en pantalla puede ser enviado a impresora.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	

Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Informe de Stock, Entrada y Salida de Artículos General”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema solicita que el usuario seleccione el periodo a informar.
4. El usuario selecciona el periodo a informar que puede ser un día, un mes, o alguno entre 2 fechas dadas por el usuario.	
	5. El sistema verifica, en caso que se haya dado 2 fechas, que estas sean válidas y coherentes entre sí.
	6. El sistema genera listado de insumos y medicamentos con su respectivo stock y cantidad que ha entrado y salido durante el día, o mes o periodo de tiempo que haya indicado el usuario, incluyendo las transacciones de todas las farmacias del sistema y muestra en pantalla el informe.

Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para generar el informe, por lo que informa el error al usuario.
6a.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6b.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 27.	Visualizar Bincard de Medicamento o Insumo.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema) o Encargado de Farmacia (Supervisor).
Propósito	Mostrar al usuario la bincard asociada a un medicamento o insumo, donde se reflejan los movimientos en las cantidades de entrada, salida y saldos para el artículo en un periodo de tiempo especificado y en una farmacia determinada. La bincard es usada para controlar y revisar la existencia de un producto en el inventario físico.
Resumen	El usuario solicita visualizar la bincard de un producto, del inventario de una

	<p>farmacia seleccionada. El sistema genera la vista de la bincard de acuerdo al periodo de tiempo que el usuario especificó. En la bincard se muestran las entradas, salidas y saldo del artículo, con fechas asociadas a cada movimiento y totales para el periodo de tiempo, que puede ser de un mes, o entre dos fechas requeridas.</p>
Precondición	<p>El usuario ha iniciado sesión con privilegios de supervisor o administrador del sistema de control de farmacia.</p>
Postcondición	<p>La bincard se muestra en pantalla y puede ser enviada a impresora.</p>
Tipo	<p>Primario.</p>
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.

1. El usuario elige y ejecuta la opción “Visualizar Bincard de Medicamento o Insumo”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador o de supervisor.
	3. El sistema solicita que el usuario seleccione el artículo para el cual se visualizará la bincard, la farmacia que lo posee y el periodo considerado.
4. El usuario selecciona el artículo por nombre o código; selecciona la farmacia y el periodo de tiempo indicando la fecha de inicio y fecha de término.	5. El sistema verifica que las fechas dadas sean válidas y coherentes entre sí.
	6. El sistema muestra en pantalla la bincard del artículo, con los siguientes datos: código y nombre del artículo, fecha de consulta, stock mínimo, stock crítico, stock máximo y detalle con fecha de cada movimiento de entrada y salida, junto con el cálculo parcial de saldo. Todos los movimientos mostrados están dentro del periodo de tiempo que haya indicado el usuario.
Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para visualizar

	una bincard, por lo que informa el error al usuario.
4a.-El usuario cancela la solicitud de visualizar la bincard.	4b.- El sistema regresa a la pantalla principal del Sistema de Control de Farmacia.
	5a.- El sistema detecta que las fechas de inicio y fin dadas para el intervalo requerido no están en el orden cronológico esperado, por lo que envía mensaje de error al usuario para que corrija el problema. El sistema regresa al N° 3.
6a.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6b.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 28.	Solicitar Informe de Entregas de Medicamentos Controlados.
Actores	Jefe de Farmacia. (Usuario Administrador del Sistema).
Propósito	Generar y mostrar en pantalla un reporte con el detalle de los medicamentos sujetos a control legal, como psicotrópicos y estupefacientes de uso médico, que han sido

	dispensados durante un periodo de tiempo desde cualquiera de las farmacias del consultorio.
Resumen	El usuario requiere generar el informe de entregas de medicamentos controlados, por lo que el sistema le solicita que indique el intervalo de fechas a considerar, para buscar en las dispensaciones de todas las farmacias. Luego el sistema muestra en pantalla los registros de cada entrega o en caso de no existir entregas, muestra mensaje indicando la situación.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla informe de entregas de medicamentos controlados, el que puede ser enviado a impresora por el usuario.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	

Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Informe de Medicamentos Controlados”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema solicita que el usuario seleccione, la fecha de inicio y la fecha de término del periodo a considerar, en la búsqueda de entregas de medicamentos controlados.
4. El usuario selecciona la fecha de inicio y fecha de término.	5. El sistema verifica que las fechas dadas sean válidas y coherentes entre sí.
	6. El sistema muestra en pantalla el informe de medicamentos controlados que ha sido entregado dentro del periodo de tiempo indicado por el usuario, detallando lo siguiente: código y nombre del medicamento, fecha de entrega, cantidad entregada, RUN Paciente, N° de ficha del paciente, nombres y apellidos del paciente, RUN del Profesional que recetó.

Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que comunica el error al usuario.
4a.- El usuario cancela la solicitud del informe.	4b.- El sistema va a la pantalla principal del Sistema de Control de Farmacia.
	5a.- El sistema detecta que las fechas de inicio y fin dadas para el intervalo requerido no están en el orden cronológico esperado, por lo que envía mensaje de error al usuario para que corrija el problema. El sistema regresa al N° 3.
6a.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6b.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 29.	Solicitar Informe de Medicamentos e Insumos Devueltos.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
	Mostrar al usuario un reporte con el

Propósito	detalle de los insumos y medicamentos devueltos a farmacias del consultorio, dentro de un intervalo de tiempo indicado por el usuario.
Resumen	El usuario solicita ver informe de los medicamentos e insumos que han sido devueltos desde las farmacias al Departamento de Abastecimiento. El sistema genera el informe de acuerdo al intervalo de fechas que el usuario especificó. El usuario puede imprimir el informe si lo desea.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	La bincard se muestra en pantalla y puede ser enviada a impresora.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	

Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Informe de Medicamentos e Insumos Devueltos”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema solicita que el usuario seleccione las fechas de inicio y fin del periodo a consultar para generar el informe de devoluciones.
4. El usuario selecciona la fecha de inicio y fecha de término del periodo a consultar.	5. El sistema verifica que las fechas dadas sean válidas y coherentes entre sí.
	6. El sistema muestra en pantalla los registros de devoluciones encontrados, que están dentro del intervalo de fechas indicado por el usuario. Se detalla las devoluciones con el N° y fecha de la devolución, N° y fecha de pedido, código y nombre del artículo, cantidad devuelta, motivo de la devolución y estado de confirmación de la operación por parte de Abastecimiento.

Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que da aviso del error al usuario.
	5a.- El sistema detecta que las fechas de inicio y fin dadas para el intervalo requerido no están en el orden cronológico esperado, por lo que envía mensaje de error al usuario para que corrija el problema. El sistema regresa al N° 3.
6a.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6b.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 30.	Solicitar Informe Pacientes Inasistentes por Periodo.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
Propósito	Mostrar al usuario un informe con el detalle de los pacientes crónicos del consultorio, que no asistieron a citación para retirar sus medicamentos desde

	farmacia crónicos.
Resumen	De acuerdo al intervalo de tiempo definido por el usuario, el sistema busca y muestra los registros de pacientes que no asistieron a retirar medicamentos, estando citados dentro de dicho intervalo de tiempo.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla el informe de pacientes inasistentes y puede ser enviado a impresora.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.

<p>1. El usuario elige y ejecuta la opción “Informe de Pacientes Inasistentes por Periodo”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.</p>
	<p>3. El sistema solicita que el usuario seleccione las fechas de inicio y fin del periodo a consultar para generar el informe de pacientes inasistentes.</p>
<p>4. El usuario selecciona la fecha de inicio y fecha de término del periodo a consultar.</p>	<p>5. El sistema verifica que las fechas dadas sean válidas y coherentes entre sí.</p>
	<p>6. El sistema busca dentro del intervalo de fechas indicado, los registros de los pacientes que no asistieron en la fecha de citación, a retirar medicamentos, y los muestra en pantalla.</p>
<p>Cursos alternos:</p>	
	<p>2a.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que da aviso del error al usuario.</p>
	<p>5a.- El sistema detecta que las fechas de inicio y fin dadas para el intervalo requerido no están en el orden cronológico esperado, por lo que envía mensaje de error al usuario para que corrija el problema. El sistema</p>

	regresa al N° 3.
	6a.- El sistema no encontró registros de pacientes inasistentes en el periodo consultado, por lo que muestra en pantalla en mensaje “No se encontraron Pacientes inasistentes en el periodo.”
6b.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6c.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 31.	Solicitar Informe de Consumo por Familia.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
Propósito	Mostrar al usuario un informe con el detalle del consumo de medicamentos e insumos por parte de los pacientes integrantes de una misma familia, dentro de un periodo de tiempo específico.
Resumen	El usuario solicita informe de consumo de medicamentos e insumos de una familia específica, define el intervalo de tiempo para la consulta y el sistema muestra en

	pantalla el informe solicitado que puede ser enviado a impresora.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla el informe de consumo por familia dentro del periodo solicitado y puede ser enviado a impresora.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Informe de consumo por Familia”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de

	administrador.
	3. El sistema solicita que el usuario seleccione las fechas de inicio y fin del periodo a consultar para generar el informe de consumo por familia.
4. El usuario selecciona la fecha de inicio y fecha de término del periodo a consultar y el N° de familia.	5. El sistema verifica que las fechas dadas sean válidas y coherentes entre sí y que la Familia exista.
	6. El sistema busca dentro del intervalo de fechas indicado, los registros de dispensaciones hechas a integrantes de la familia consultada, calcula cantidades totales por medicamento y los muestra en pantalla.
Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que da aviso del error al usuario.
	5a.- El sistema detecta que las fechas de inicio y fin dadas para el intervalo requerido no están en el orden cronológico esperado o el N° de familia consultada no existe en el sistema, por lo que envía mensaje de error al usuario para que corrija el problema. El sistema regresa al N° 3.

	6a.- El sistema no encontró registros de dispensaciones a integrantes de la familia en el periodo consultado, por lo que muestra en pantalla en mensaje “La Familia consultada, no registra consumo de medicamentos en el periodo.”
6b.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6c.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 32.	Solicitar Informe de Consumo por Paciente.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
Propósito	Mostrar al usuario un informe con el detalle del consumo de medicamentos e insumos por parte de un paciente, dentro de un periodo de tiempo específico.
Resumen	El usuario solicita informe de consumo de medicamentos e insumos de un Paciente específico, define el intervalo de tiempo a consultar y el sistema muestra en pantalla el informe solicitado que puede ser enviado a

	impresora.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla el informe de consumo por Paciente dentro del periodo solicitado y puede ser enviado a impresora.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Informe de consumo por Paciente”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.

	3. El sistema solicita que el usuario seleccione las fechas de inicio y fin del periodo a consultar, e ingrese el N° de ficha del Paciente.
4. El usuario selecciona la fecha de inicio y fecha de término del periodo a consultar y el N° de ficha del Paciente.	5. El sistema verifica que las fechas dadas sean válidas y coherentes entre sí y que el Paciente esté registrado en el consultorio.
	6. El sistema busca dentro del intervalo de fechas indicado, los registros de dispensaciones hechas al Paciente consultado, calcula cantidades totales por medicamento y los muestra en pantalla.
Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que da aviso del error al usuario.
	5a.- El sistema detecta que las fechas de inicio y fin dadas para el intervalo requerido no están en el orden cronológico esperado o el N° de ficha del Paciente consultado no existe en el sistema, por lo que envía mensaje de error al usuario para que corrija el problema. El sistema regresa al N° 3.
	6a.- El sistema no encontró registros de dispensaciones al Paciente dentro del periodo consultado, por lo que muestra en

	pantalla en mensaje “El Paciente no registra consumo de medicamentos en el periodo.”
6b.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6c.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 33.	Solicitar Informe de Medicamentos Prescritos por Profesional.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
Propósito	Mostrar al usuario un informe con el detalle de los medicamentos e insumos prescritos por parte de un Profesional, dentro de un periodo de tiempo específico.
Resumen	El usuario solicita informe de medicamentos prescritos por un Profesional Específico en un periodo de tiempo ingresado y el sistema muestra en pantalla el informe solicitado que puede ser enviado a impresora.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de

	control de farmacia.
Postcondición	Se muestra en pantalla el informe con listado de medicamentos prescrito s por el Profesional, dentro del periodo solicitado y puede ser enviado a impresora.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Informe de Medicamentos Prescritos por Profesional”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema solicita que el usuario seleccione las fechas de inicio y fin del periodo a consultar, e ingrese el N° de RUN

	del Profesional.
4. El usuario selecciona la fecha de inicio y fecha de término del periodo a consultar y el N° de RUN del Profesional.	5. El sistema verifica que las fechas dadas sean válidas y coherentes entre sí y que el RUN del Profesional esté registrado en el consultorio.
	6. El sistema busca dentro del intervalo de fechas indicado, los registros de dispensaciones donde aparece el Profesional consultado, calcula cantidades totales por medicamento y los muestra en pantalla.
Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que da aviso del error al usuario.
	5a.- El sistema detecta que las fechas de inicio y fin dadas para el intervalo requerido no están en el orden cronológico esperado o el N° de RUN del Profesional consultado no existe en el sistema, por lo que envía mensaje de error al usuario para que corrija el problema. El sistema regresa al N° 3.
	6a.- El sistema no encontró registros de dispensaciones donde aparezca el Profesional consultado, dentro del periodo especificado, por lo que muestra en pantalla

	el mensaje “El Profesional no registra prescripciones realizadas en el periodo.”
6b.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6c.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 34.	Solicitar Informe Censo de Pacientes por Grupo Etario y Tipo de Atención.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
Propósito	Mostrar al usuario un informe donde se muestra la cantidad de pacientes por grupo etario y tipo de atención recibida.
Resumen	El usuario solicita informe de censo por grupo etario tipo de atención recibida que puede ser atención de morbilidad o atención de pacientes crónicos.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.

<p>Postcondición</p>	<p>Se muestra en pantalla el informe con las cantidades de pacientes clasificados por grupo etario y por tipo de atención. El informe puede ser enviado a impresora.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>
<p>1. El usuario elige y ejecuta la opción “Censo de Pacientes por Grupo Etario y Tipo Atención”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.</p>
	<p>3. El sistema solicita que el usuario confirme la generación del informe.</p>
<p>4. El usuario confirma la generación del informe.</p>	<p>5. El sistema contabiliza la cantidad de pacientes que hay por cada grupo etario y de acuerdo al tipo de atención (morbilidad o</p>

	crónicos).
	6. El sistema muestra en pantalla el informe generado.
Cursos alternos:	
	2ª.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que da aviso del error al usuario.
4a.- El usuario cancela la generación del informe.	4b.- El sistema regresa a la pantalla principal del Sistema de Control de Farmacia.
6a.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6b.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 35.	Solicitar Informe de Pacientes con Diagnóstico Climaterio.
Actores	Jefe de Farmacia (Usuario Administrador del Sistema).
	Mostrar al usuario un informe donde se

Propósito	muestran los pacientes con diagnóstico climaterio.
Resumen	El usuario solicita informe de pacientes que se les haya dispensado medicamentos y cuyo diagnóstico es “climaterio”. El sistema presenta los datos de pacientes, ordenados por género.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Se muestra en pantalla el informe con los pacientes con diagnóstico climaterio, clasificados por género. El informe puede ser enviado a impresora.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.

1. El usuario elige y ejecuta la opción “Pacientes con Diagnóstico Climaterio”.	
	2. El sistema verifica que el usuario que solicita la ejecución tiene privilegios de administrador.
	3. El sistema solicita que el usuario confirme la generación del informe.
4. El usuario confirma la generación del informe.	5. El sistema busca los datos de pacientes con diagnóstico climaterio y los ordena por género.
	6. El sistema muestra en pantalla el informe generado.
Cursos alternos:	
	2ª.- El sistema detecta que el usuario no tiene privilegios suficientes para solicitar el informe, por lo que da aviso del error al usuario.
4a.- El usuario cancela la generación del informe.	4b.- El sistema regresa a la pantalla principal del Sistema de Control de Farmacia.
6a.- El usuario solicita enviar a impresora el informe que se ve en pantalla.	6b.- El sistema envía el informe a impresora e informa al usuario del resultado del envío.

Caso de uso N° 36.	Modificar Datos de Paciente.
Actores	Administrador, Supervisor, Usuario
Propósito	Permitir la modificación de los datos personales, registrados de un Paciente.
Resumen	El usuario solicita al sistema la modificación de datos personales de un Paciente. El sistema verifica el perfil del usuario que solicita el servicio y muestra los datos del Paciente para su modificación, espera confirmación y graba las modificaciones.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia, supervisor o usuario normal.
Postcondición	Datos del Paciente quedan modificados y se muestra en pantalla mensaje indicando que la modificación se realizó con éxito.

Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Modificar Paciente”.	
	2. El sistema verifica que el usuario que solicita la ejecución haya iniciado una sesión en el sistema de forma correcta.
	3. El sistema solicita que el usuario ingrese el N° de ficha o RUN del Paciente.
4. El usuario ingresa el N° de ficha o RUN del Paciente.	5. El sistema busca el paciente y si este existe, muestra los datos personales en pantalla.
5. El usuario modifica los datos personales que requiere del Paciente y solicita grabarlos.	6. El sistema graba las modificaciones y muestra mensaje al usuario “Los datos del Paciente han sido modificados”.

Cursos alternos:	
	2ª.- El sistema detecta que el usuario no se ha autenticado en el sistema correctamente, por lo que envía mensaje de error por pantalla.
4a-5a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del Sistema de Control de Farmacia.
	5a.- El sistema no encuentra al Paciente buscado, por lo que envía mensaje de error al usuario.

Caso de uso N° 37.	Modificar Datos de Usuario.
Actores	Jefe de Farmacia (Administrador)
Propósito	Permitir la modificación de los datos de la cuenta, de un Usuario del sistema.
Resumen	El Administrador del Sistema solicita modificar datos de un usuario del sistema,

	para lo que debe indicar el RUN del usuario y el Sistema le mostrará los datos de la cuenta de usuario para que modifique lo que requiera. Puede modificar Nombre, Apellidos, cargo, login, password o perfil del Usuario.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.
Postcondición	Datos del Usuario del Sistema quedan modificados y se muestra en pantalla mensaje indicando que la modificación se realizó con éxito.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción	

<p>“Modificar Usuario”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la modificación tenga perfil de Administrador del Sistema de Farmacia.</p>
	<p>3. El sistema solicita que el usuario ingrese el N° de RUN del Usuario que desea modificar.</p>
<p>4. El usuario ingresa el N° de RUN del Usuario que desea modificar.</p>	<p>5. El sistema busca el Usuario a modificar y si existe, muestra los detalles de la cuenta en pantalla.</p>
<p>5. El usuario modifica los datos que requiere del Usuario como Nombre, Apellidos, login, password, perfil y solicita grabarlos.</p>	<p>6. El sistema graba las modificaciones y muestra mensaje al usuario “Los datos del Usuario han sido modificados”.</p>
<p>Cursos alternos:</p>	
	<p>2a.- El sistema detecta que el usuario no tiene privilegios de Administrador del Sistema de Farmacia, por lo que envía mensaje de error por pantalla y rechaza la petición de modificación.</p>
<p>4a-5a.- El usuario cancela la operación.</p>	<p>4b.- El sistema regresa a la pantalla principal del Sistema de Control de Farmacia.</p>
	<p>5a.- El sistema no encuentra al Usuario buscado, por lo que envía mensaje de error por pantalla.</p>

Caso de uso N° 38.	Modificar Datos de Farmacia.
Actores	Jefe de Farmacia (Administrador)
Propósito	Permitir la modificación de los datos de una farmacia externa inscrita en el sistema.
Resumen	El Administrador del Sistema solicita modificar datos de una farmacia externa inscrita en el sistema, para lo que debe indicar el nombre de la farmacia y el Sistema le mostrará los datos asociados, para que modifique lo que requiera. Puede modificar Nombre de la farmacia, dirección o fono.
Precondición	El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.

Postcondición	Datos de la Farmacia inscrita en el Sistema quedan modificados y se muestra en pantalla mensaje indicando que la modificación se realizó con éxito.
Tipo	Primario.
<u>Curso normal de los eventos.</u>	
Acción del actor	Respuesta del sistema.
1. El usuario elige y ejecuta la opción “Modificar Farmacia”.	
	2. El sistema verifica que el usuario que solicita la modificación tenga perfil de Administrador del Sistema de Farmacia.
	3. El sistema solicita que el usuario ingrese el nombre de la farmacia que desea modificar.
4. El usuario ingresa el nombre de la farmacia que desea modificar.	5. El sistema busca la farmacia a modificar y si existe, muestra los detalles de esta en

	pantalla.
5. El usuario modifica los datos que requiere de la farmacia como Nombre, dirección o fono y solicita grabarlos.	6. El sistema graba las modificaciones y muestra mensaje al usuario “Los datos de la farmacia han sido modificados”.
Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios de Administrador del Sistema de Farmacia, por lo que envía mensaje de error por pantalla y rechaza la petición de modificación.
4a-5a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del Sistema de Control de Farmacia.
	5a.- El sistema no encuentra la farmacia buscada, por lo que envía mensaje de error por pantalla.

Caso de uso N° 39.	Modificar Datos de Insumos y Medicamentos.
Actores	Jefe de Farmacia (Administrador)
Propósito	Permitir la modificación de los datos de los insumos y medicamentos existentes en el sistema.

<p>Resumen</p>	<p>El Administrador del Sistema solicita modificar datos de un insumo o medicamento (artículo) existente en las farmacias del sistema, para lo que debe indicar el código del artículo y el Sistema le mostrará los datos asociados, para que modifique lo que requiera.</p>
<p>Precondición</p>	<p>El usuario ha iniciado sesión con privilegios de administrador del sistema de control de farmacia.</p>
<p>Postcondición</p>	<p>Datos del insumo o medicamento quedan modificados y se muestra en pantalla mensaje indicando que la modificación se realizó con éxito.</p>
<p>Tipo</p>	<p>Primario.</p>
<p><u>Curso normal de los eventos.</u></p>	
<p>Acción del actor</p>	<p>Respuesta del sistema.</p>
<p>1. El usuario elige y ejecuta la opción “Modificar Datos de Insumos o Medicamentos”.</p>	
	<p>2. El sistema verifica que el usuario que solicita la modificación tenga perfil de</p>

	Administrador del Sistema de Farmacia.
	3. El sistema solicita que el usuario ingrese el código del artículo que desea modificar.
4. El usuario ingresa el código del artículo que desea modificar.	5. El sistema busca el artículo a modificar y si existe, muestra los detalles de este en pantalla.
5. El usuario modifica los datos que requiere del insumo o medicamento y solicita grabarlos.	6. El sistema graba las modificaciones y muestra mensaje al usuario “Los datos del artículo han sido modificados”.
Cursos alternos:	
	2a.- El sistema detecta que el usuario no tiene privilegios de Administrador del Sistema de Farmacia, por lo que envía mensaje de error por pantalla y rechaza la petición de modificación.
4a-5a.- El usuario cancela la operación.	4b.- El sistema regresa a la pantalla principal del Sistema de Control de Farmacia.
	5a.- El sistema no encuentra el artículo buscado, por lo que envía mensaje de error por pantalla.