

Índice

1. Introducción	5
1.1. Introducción del proyecto.....	5
1.2. Estructura de la memoria.....	7
1.2.1 Solución y objetivos del proyecto	9
2. Estudio y Aplicación de la metodología Métrica 3.....	9
2.1. Introducción.....	9
2.1.1 Introducción a Métrica 3	9
2.1.2. Características	9
2.1.3. Origen	10
2.1.4. Aportaciones	10
2.1.5. Estructura	11
2.2. Estudio de la viabilidad del sistema (EVS).....	14
2.2.1. Actividad EVS 1: Establecimiento del alcance del sistema.....	15
2.2.2. Actividad EVS 2: Estudio de la situación actual.....	21
2.2.3. Actividad EVS 3: Definición de requisitos del sistema.....	23
2.2.4. Actividad EVS 4: Estudio de alternativas de solución	28
2.2.5. Actividad EVS 5: Valoración de las alternativas	30
2.2.6. Actividad EVS 6: Selección de la solución	31
2.3.2. Análisis del sistema de información (ASI).....	33
2.3.1. Actividad ASI 1: Definición del sistema	34
2.3.2. Actividad ASI 2: Establecimiento de requisitos.....	38
2.3.3. Actividad ASI 3: Identificación de los subsistemas de análisis	52
2.3.4. Actividad ASI 4: Análisis de casos de uso	56
2.3.5. Actividad ASI 5: Análisis de clases	57
2.3.6. Actividad ASI 6: Elaboración del modelo de datos	58
2.3.7. Actividad ASI 7: Elaboración del modelo de datos	58
2.3.8. Actividad ASI 8: Definición de interfaces de usuario	58

2.3.9. Actividad ASI 9: Análisis de consistencia y especificación de requisitos	58
2.3.10. Actividad ASI 10: Especificación del plan de pruebas.....	65
2.4. Diseño del sistema de información (DSI)	67
2.4.1. Actividad DSI 1: Definición de la arquitectura del sistema.....	68
2.4.2. Actividad DSI 2: Diseño de la arquitectura de soporte	71
2.4.3. Actividad DSI 3: Diseño de casos de uso reales	71
2.4.4. Actividad DSI 4: Diseño de clases	72
2.4.5. Actividad DSI 6: Diseño físico de datos.....	73
2.4.6. Actividad DSI 7: Verificación y aceptación de la arquitectura del sistema	74
2.4.7. Actividad DSI 8: Generación de especificaciones de construcción	75
2.4.8. Actividad DSI 9: Diseño de la migración y carga inicial de datos.....	77
2.4.9. Actividad DSI 10: Especificación técnica del plan de pruebas.....	77
2.4.10. Actividad DSI 11: Establecimiento de requisitos de implantación.....	79
2.4.11. Actividad DSI 12: Aprobación del diseño del sistema de información	80
2.5. Construcción del sistema de información (CSI)	81
2.5.1. Actividad CSI 1: Preparación del entorno de generación y construcción	82
2.5.2. Actividad CSI 2: Generación del código de los componentes y procesos	83
2.5.3. Actividad CSI 3: Ejecución de las pruebas unitarias.....	83
2.5.4. Actividad CSI 4: Ejecución de las pruebas de integración.....	84
2.5.5. Actividad CSI 5: Ejecución de las pruebas del sistema.....	85
2.5.6. Actividad CSI 6: Elaboración de los manuales de Usuario	87
2.5.7. Actividad CSI 7: Definición de la formación de usuarios finales.	87
2.5.8. Actividad CSI 8: Construcción de los componentes y procedimientos de migración y carga inicial de datos	88

2.5.9 Actividad CSI 9: Aprobación del sistema de información	88
2.6. Planificación.....	89
3. Diagramas.....	91
3.1. Diagramas de casos de uso del proyecto	91
3.2. Diagramas de clases del proyecto	97
3.3. Diagrama de clases de la gestión de datos	98
3.4. Diagramas de interacción del proyecto	102
3.5. Diagramas transición de estados del proyecto.....	137
4 Herramientas utilizadas.....	149
4.1. MySql.....	149
4.1.1. Introducción	149
4.1.2. Características	150
4.2. Visual Studio	156
4.2.1. Introducción	156
4.2.2. Características	156
4.2.3. Resumen.....	161
4.3. Aqua Data Studio	163
4.3.1. Introducción	163
4.3.2. Características	163
5. Manual de Usuario.....	165
5.1. Instalación	165
5.2. Introducción.....	167
5.3. Manual de Usuario	167
5.3.1. Elementos Comunes	168
5.3.2. Acceso al Sistema	175
5.3.3. Pacientes	178
5.3.4. Proveedores.....	187
5.3.5. Tratamiento.....	190
5.3.6. Contabilidad.....	197
5.3.7. Caja.....	202
5.3.8. Base de Datos.....	204

5.3.9. Almacén.....	205
5.3.10. Ayuda	207
6. Conclusiones	211
6.1. Problemas encontrados durante el desarrollo del proyecto	211
6.2. Conclusiones finales	212
6.3. Posibles mejoras del sistema de información.....	212
6. Bibliografía.....	215

Cápitul01: Introducción

1.1: Introducción del proyecto

El objetivo principal de este proyecto es el desarrollo completo de un sistema de información para la gestión de una consulta de masajes terapéuticos. La consulta está relacionada con el nuevo mundo de la medicina oriental como método alternativo de cura. El mecanismo de la consulta consiste en realizar un estudio previo al paciente de las enfermedades que padece y de todos los masajes existentes identificando cuáles son aceptables y cuáles deben rechazarse. Para cada masaje pueden aplicarse diferentes técnicas y seleccionar las zonas de masaje. Este conjunto de posibilidades sobre un paciente se define como tratamiento. Por último, después de aplicar un masaje se anotan los resultados obtenidos tras emplear ciertas pruebas al paciente. Este punto es el más importante porque junto con el número de tratamientos que lleva un paciente y sus resultados ante las pruebas se pueden obtener una evolución de la mejora.

Para mayor control del paciente, el sistema debe de almacenar, además de los datos personales, datos físicos y psíquicos que puedan intervenir en el tratamiento, hábitos, ritmo de vida y otros posibles factores. El sistema debe almacenar también el seguimiento del tratamiento que está recibiendo, así como un listado con los tratamientos recibidos, y la evolución del paciente por medio de un historial médico.

El sistema debe cubrir otros aspectos que no pueden faltar en un sistema de información como son la gestión de una contabilidad de pacientes y proveedores, gestión de una agenda para el control de citas, un stock de almacén, un listado de informes para poder imprimirlos y opciones de importar y exportar la base de datos.

1.1.1 Solución y objetivos del proyecto:

Para llevar una gestión eficiente de todo lo argumentado se sabe que, en la actualidad existen distintos tipos de software estándares que desarrollen parte del producto que se exige, pero daría lugar a la utilización de varias aplicaciones lo cual es un gran inconveniente. Tener que manejar, actualizar, interaccionar y depender de varias aplicaciones es bastante ineficiente en la que se puede realizar en una misma. La

falta de información que se tiene de las demás aplicaciones para la reutilización requiere el desarrollo de la aplicación desde el principio.

El sistema de información se desarrollará con la estructura orientada a objetos de la metodología de Métrica 3. Utilizamos Métrica porque se caracteriza por dar más importancia al análisis que al desarrollo. Esto permite una mayor capacidad de adaptación a los cambios que pueden surgir en la comunicación con el cliente y disminuye la gran tarea del mantenimiento del software.

Para el análisis y diseño del sistema de información se utilizará el lenguaje de especificación UML y la herramienta para generar los distintos diagramas, modelos y tablas será MagicDraw.

Se necesitará la creación de una base de datos que almacene toda la información que se manipula, además se desarrollará una aplicación que se utilice con la base de datos y juntos con otros datos que interactúe con el usuario pueda definir toda la funcionalidad que se pide. Una característica deseable es que disponga de un entorno amigable para que el modelo sirva como solución y no provoque una problemática a la consulta, por lo que se realizará sobre sistema operativo gráfico y utilizando un lenguaje visual. Contará con una interfaz intuitiva y sencilla para no perder tiempos ociosos en la aplicación y con una instalación e implantación fácil y rápida. Por ello la implementación de la aplicación se realizará con la plataforma de Visual C# y la creación de la base de datos se hará con el gestor MySQL por ser de licencia gratuita.

Desde un punto de vista informático, podemos observar que el proyecto requiere un estudio exhaustivo de la organización de la consulta para la creación, manejo e implementación de una base de datos eficiente, además de la adquisición de conocimientos en lenguajes visuales y aprendizaje de comunicación del lenguaje visual con la base de datos.

Desde el punto de vista empresarial, este proyecto minimiza los tiempos en búsquedas de datos, organización y planificación de la organización y con su facilidad de uso podrá invertirse dicho tiempo en invertir en actualizar nuevos campos de la materia, y actualizar la aplicación en ofertas de tratamientos.

1.2. Estructura de la memoria

La memoria de este proyecto está dividida en seis capítulos y un séptimo con la bibliografía utilizada. A continuación se resumen el contenido de cada capítulo

- Capítulo 1: Introducción. Es el capítulo en el que nos encontramos. Aquí se describe una introducción de los objetivos del proyecto.
- Capítulo 2: Estudio y Aplicación de la metodología Métrica. En este capítulo, que es el más extenso y uno de los más importantes de la memoria, se describe las definiciones de las actividades y tareas de Métrica, así como la aplicación en el proyecto.
- Capítulo 3: Diagramas. En este capítulo están todos los diagramas que se han elaborado por necesidad de métrica 3. Desde el capítulo anterior se harán numerosas referencias hacia este capítulo.
- Capítulo 4: Herramientas Utilizadas. Se describe el software utilizado para la realización del proyecto así como las características más especiales de cada uno.
- Capítulo 5: Manual de Usuario: Este capítulo está dirigido al usuario final que vaya a utilizar la aplicación. Aquí se podrá informar de cómo utilizar la aplicación desarrollada para sacarle el mayor partido posible.
- Capítulo 6: Conclusiones. Se describe los problemas que se han encontrado a lo largo del desarrollo de la aplicación, unas conclusiones finales y para terminar unas líneas sobre una ampliación para el futuro de la aplicación.
- Capítulo 7: Bibliografía: Se describe el material bibliográfico necesario para realizar el proyecto.

Capítulo 2: Estudio y Aplicación de la Metodología de Métrica 3

2.1 Introducción

2.1.1 Introducción a Métrica 3

Con la metodología Métrica versión 3 tenemos a nuestra disposición una herramienta muy útil para el desarrollo de todo el ciclo de un sistema de información. Métrica 3 es en sí una metodología para la planificación, desarrollo y mantenimiento del sistema de información. Si utilizamos Métrica 3 podemos lograr los siguientes objetivos:

- Definir un sistema de información que nos facilite a cumplir los objetivos de nuestra empresa a través de definiciones para el desarrollo.
- Invertir mucho más tiempo en el análisis de requisitos que en la implementación del sistema.
- Aumentar la comunicación entre departamentos a través de una mayor capacidad para la adaptación de nuevos cambios e intentando reutilizar en todo lo posible.
- Mejorar y facilitar la comunicación entre los diferentes integrantes en la realización del sistema de información.

2.1.2 Características

- Está pensado para todas las diferentes tecnologías que en la actualidad existen para la creación de software, teniendo en cuenta criterios como calidad, coste y plazos.
- Se da cobertura dos tipos de desarrollo, por un lado el estructurado, y por otro lado el orientado a objetos.
- Todas las actividades de métrica pueden ser automatizadas debido a que las técnicas que se utilizan pueden ser cubiertas por una amplia variedad de herramientas.

- Es una metodología dinámica, debido a que se adapta con facilidad según sean las condiciones de la organización.
- Es simple, utiliza un vocabulario común y un conjunto de tareas y productos para construir con éxito y calidad los sistemas de información

2.1.3 Origen

- Esta es una versión mejorada de la anterior, en la que se mantienen características tales como adaptabilidad, flexibilidad y sencillez. También mantiene el formato de actividades y tareas. Para cada tarea se informa de los participantes que actúan, los productos que son de entrada y salida y todas las técnicas necesarias para obtener los productos.
- Para elaborar esta última versión, se ha tenido en cuenta los métodos más extendidos y los últimos estándares en la ingeniería del software. Por otro lado, también se han intervenido referencias de seguridad y gestión de proyectos, así como la experiencia de los usuarios en antiguas versiones.

2.1.4 Aportaciones

- Nos da una metodología mixta ya que permite desarrollos estructurados y orientados a objetos.
- Es totalmente compatible con otros estándares establecidos, por lo tanto conseguimos calidad de software y una ingeniería del software.
- Incorpora estructuras genéricas con detallando los participantes que interactúan.
- Incluye la claridad producto obtenido por cada tarea y actividad.
- Añade la especificación de participantes a nivel de tarea, un ciclo de vida de pruebas reforzado y nuevas tecnologías orientadas a objetos como Cliente/Servidor.

2.1.5 Estructura

Según hemos visto, Métrica tiene un enfoque orientado al proceso, debido a su tendencia en los estándares que buscan este sentido. Al cumplir estos estándares, métrica cubre el proceso de desarrollo y mantenimiento de los sistemas de información.

Abarca un desarrollo completo para cualquier sistema de información, por lo que su estructura se tiene que adaptar según sea cada proyecto.

La estructura de métrica se basa en actividades y estas en tareas. Para cada tarea se define su contenido, detallando sus acciones, técnicas, prácticas, productos y participantes.

Las actividades no son secuenciales, y podemos realizarlas en orden distinto a la enumeración que tienen o cuando sea posible de forma paralela. De todas formas no se terminará un proceso hasta no haber finalizado todas las actividades del mismo.

A continuación se describe los procesos de los que se compone métrica:

- Planificación de sistemas de información

El entorno de amplia competitividad y cambio de las que dispone una organización, hace que sea más difícil de tener sistemas y tecnologías de información con flexibilidad para las nuevas exigencias y con la velocidad que ofrece el entorno.

Objetivos:

- Describir la situación actual.
 - Proporcionar una arquitectura de información.
 - Planificación de proyectos, teniendo en cuenta prioridad, recursos y costes.
 - Plan de seguimiento y cumplimiento de lo planificado.
 - Desarrollo del sistema de información
- Este proceso contiene todas las actividades necesarias para desarrollar un sistema de información, dando cobertura desde el análisis de los requisitos necesarios hasta la instalación del software.

Debido al gran número de actividades se ha dividido en los siguientes procesos para su mejor comprensión

- Estudio de Viabilidad del Sistema.
 - En este objetivo se quiere estudiar un conjunto de necesidades y proponer una solución a corto plazo. Los criterios para realizar la propuesta son tácticos y están relacionados con aspectos económicos, técnicos, operativos y legales.
 - Los resultados aquí obtenidos nos servirán de base para tomar decisiones en los siguientes procesos.
 - Se valorarán varias alternativas y para determinar la solución a realizar se estudiará su impacto en la organización.
- Análisis del Sistema de Información.

Los objetivos que se quieren abarcar en este proceso son:

 - Especificación detallada del sistema.
 - Obtener modelos que cubran las necesidades de información.
 - Cubrir desarrollos estructurados y orientados a objetos.
 - Identificar subsistemas de análisis y especificar interfaces entre sistema y usuario.
- Diseño del Sistema de Información.

Los objetivos de este proceso son:

 - Definir la arquitectura de sistema y su entorno tecnológico.
 - Definir detalladamente los componentes del sistema.
 - Generar especificaciones de construcción.
- Construcción del Sistema de Información

Los objetivos que se pretende obtener en este proceso son:

 - Construir y probar los componentes del sistema.
 - Desarrollar procedimientos de operación y seguridad.
 - Elaborar manuales de usuario.
 - Aplicar pruebas unitarias y de integración.

- Hacer la migración de datos y si fuera necesario carga inicial de datos.
- Implantación del Sistema de Información.
Los objetivos de este proceso son:
 - Entrega y aceptación del sistema final.
 - Realizar lo necesario para conseguir la producción del sistema.
 - Pruebas con aceptación de usuarios.
- Mantenimiento de sistemas de información

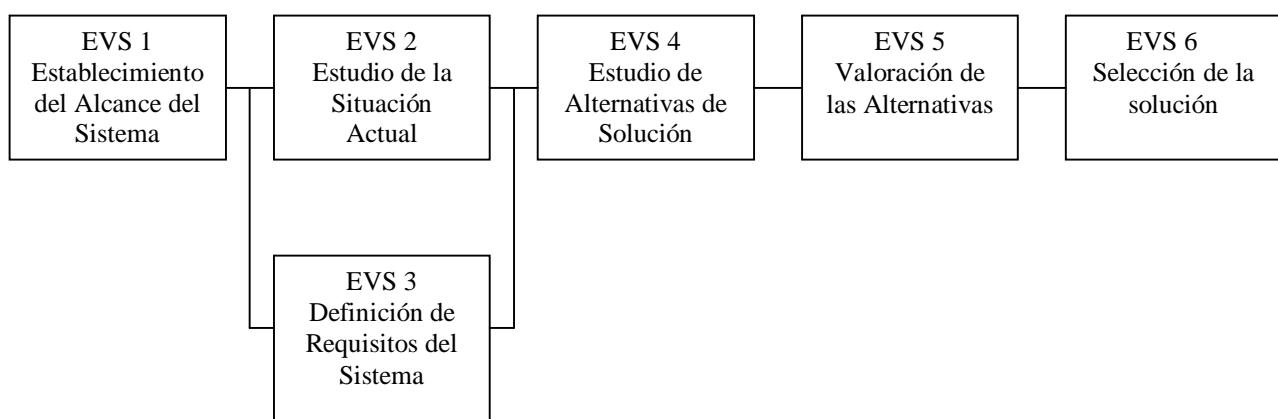
En este proceso existen actividades y tareas relativas a la modificación o retirada de todos los componentes de un Sistema de información. Métrica 3 está orientada al proceso de desarrollo del software, por lo que este proceso no entra dentro de su rango de actuación.

2.2 ESTUDIO DE VIABILIDAD DEL SISTEMA

El objetivo del estudio de viabilidad del sistema es analizar un número determinado de necesidades y proponer una solución en un tiempo a corto plazo, teniendo en cuenta restricciones (técnicas, operativas, legales y económicas). Esta solución podrá disponer de varios proyectos que enfoquen a varios sistemas de información.

A partir del estado inicial, la situación actual y los requisitos exigidos por el cliente se estudian las alternativas a la solución. Una vez detalladas las alternativas se comprueba el impacto que provocan en la organización, para poder elegir la mejor solución que satisfaga en mayor conjunto las necesidades determinadas, y definiendo una planificación según fechas de entrega.

Actividades a realizar:



Actividad 1: Establecimiento del alcance del sistema.

En esta actividad se estudia el alcance de la necesidad planteada por nuestro cliente, realizando una breve descripción del sistema. Se determinan objetivos y se inicia un estudio de requisitos identificando las unidades de la estructura organizativa afectada. También se realiza un análisis de las restricciones que puedan alterar al estudio y una planificación de la solución. Para terminar se determina el equipo de trabajo y su planificación asignando a cada uno sus tareas y responsabilidades.

Tarea 1.1: Estudio de la solicitud.

➤ Descripción General del Sistema:

El sistema que se pretende realizar trata de la gestión y organización de una empresa de masajes que en un principio la vamos a llamar MSJ.

El producto final va dedicado a un usuario único y con conocimientos muy básicos en el mundo de la informática. Siempre estaremos en contacto con él para que el producto satisfaga todas sus necesidades y buscando la sencillez como prioridad ante toda la funcionalidad.

La descripción de la organización que vamos a informatizar se basa exclusivamente en el trato con clientes para la realización de masajes. La organización debe contar con un mantenimiento de clientes con datos exhaustivos para conocer más a fondo a los clientes y al seguimiento que hay que realizar antes y después del masaje. El sistema va a cubrir un control para los tratamientos. Un tratamiento está compuesto por un conjunto de técnicas, zonas de aplicación y tipos de masaje. Comentados estos dos puntos importantes falta nombrar el que los une, que sería un diario para organizar las citas y los horarios entre pacientes, y todo esto requiere un mantenimiento de la organización.

Esto es lo más importante de la organización pero no se puede olvidar de que además se necesita de un control de proveedores, que suministran los productos dietéticos y cremas, un control de contabilidad de la organización para gestionar la entrada y salida de dinero, una facturación para pacientes que provenga de empresas y

necesiten justificar los servicios ofrecidos y otros aspectos que irán saliendo en los puntos siguientes al comentar de forma más detallada la organización.

➤ **Catálogo de objetivos del EVS**

El objetivo del estudio de viabilidad del sistema fija en detallar como se encuentra la organización de la empresa a estudiar y realiza un conjunto de soluciones que cubra en mayor parte las funciones requeridas por el cliente. En cada actividad los objetivos primordiales son:

Objetivo: Estudio de la situación actual.

Descripción: Se trata de un estudio del funcionamiento de los sistemas de información que posee para conseguir la posibilidad de mejora, ayudados de los requisitos exigidos por el cliente.

Objetivo: Definición de requisitos del sistema.

Descripción: Un requisito es en modo general un conjunto de funcionalidad que requiere nuestro cliente, junto con una descripción de comportamiento y una prioridad.

Objetivo: Estudio de alternativas de solución

Descripción: Tras haber estudiado el campo de desarrollo establecer un conjunto de todas las posibles soluciones que satisfagan las necesidades.

Objetivo: Valoración de alternativas

Descripción: Valorar cada alternativa descrita según provoque modificaciones en la empresa, costes, beneficios y riesgos.

Objetivo: Selección de la solución

Descripción: Se elige la solución que más se adapte a la empresa y en caso contrario se inicializa el estudio de viabilidad.

➤ Catalogo de requisitos

A continuación se exponen un resumen de los requisitos, comentados en un primer contacto con el cliente. En la sección 3 se identificarán todos los requisitos y se catalogarán añadiéndoles prioridades.

- Una gestión de clientes para tener un control lo más amplio posible sobre un conjunto de datos. De todos los datos que debemos de almacenar, tenemos además de los típicos datos personales (nombre, apellidos, dirección...) otros datos más individuales referentes a la persona que son lo más importantes: son las padencias de las personas. Esto es necesario almacenarlo porque dependiendo de las enfermedades (físicas, psicológicas, hereditarias, crónicas, degenerativas) el conjunto de masajes estará limitado según el rango y grado de sus enfermedades. Se necesitaría funcionalidad para el control de inserción, eliminación, modificación y consultas de clientes. A petición del usuario la posibilidad de que la eliminación no fuese física para poder rescatar los datos ante un posible regreso del cliente.
- Un mantenimiento con la mayor facilidad posible para poder relacionar un cliente, con un masaje y con el tratamiento que más se adecue a la zona de masaje. Esto es necesario debido a que dado un paciente determinado no se le puede aplicar cualquier masaje, éste debe ser lo más personal posible para combatir su problema. Para un problema determinado del paciente se debe de elegir un masaje determinado y dentro de cada masaje tendremos que encontrar un tratamiento o en cierto caso una técnica lo más específico/a a la zona que se le va aplicar el masaje, y determinar el número de sesiones más apropiado. Por lo comentado antes necesitamos un control personal del cliente en el que reflejara el número de sesiones ejecutadas por cada tratamiento, esto mantendría informado al masajista a partir de que sesión el paciente empieza a notar el efecto del tratamiento o por si en caso contrario lo rechaza.
- Un diario sobre el que se pueda controlar el tema de las citas, la hora de la cita y su duración, al cliente que va dirigido, tipo de masaje o tratamiento.

Además de lo comentado de la organización de pacientes y masajes, aquí interviene un apartado muy importante en este mundo de los masajes, y es que el masajista; para la recepción de una cita, debe poseer toda la información de las dolencias del paciente, verificar el masaje que se le debe aplicar, y lo más importante de la cita es la realización de pruebas y chequeos de cómo ha respondido al masaje. Todo esto sirve para que a la nueva llegada del paciente, el masajista compruebe que masaje le aplicó, y cual ha sido su resultado. De esta forma se realiza un seguimiento del paciente. Por organización actual de la empresa, el único ordenador que va a mantener la aplicación estará ubicado en la sala de masajes, y como solo existe un masajista, se desea que el diario esté constantemente visible para que el acceso al control de citas sea de inmediato del propio masajista.

- Al igual que los clientes, la aplicación poseerá un apartado de proveedores para poder obtener un resumen de cada página Web del propio proveedor, es decir, tendremos que almacenar de cada proveedor información de comunicación con la entidad: como nombre, teléfono, dirección o fax y un listado de cuales son los productos que puede ofrecernos y el precio.
- Facturación. La facturación se va a dividir en dos partes, una en la que el sistema registra el dinero que debería existir al final de cada mes según el conjunto de masajes dados al cabo del día y restando los gastos directos de los proveedores; y por otro lado se debe de registrar el dinero efectivo que entra en la empresa al ingresar de cada cliente ya sea por masaje o por compra de producto. Se recuerda que no todas las citas son iguales debido a que no son todos los masajes iguales, además de cada elección de masaje existe un tratamiento específico y según el paciente obtendremos una zona de masaje distinta.
- Generación de informes como método de recuperación de información impresa, ya sea de utilidad para el paciente o para nuestro cliente. También se realizará presupuestos para los pacientes que lo requieran.

- Un último apartado y de gran importancia en un sistema de información que es la posibilidad de importar y exportar la base de datos, es decir, darle la posibilidad al cliente de hacer copias de seguridad y restaurarlas.

Tarea 1.2: Identificación del alcance del sistema.

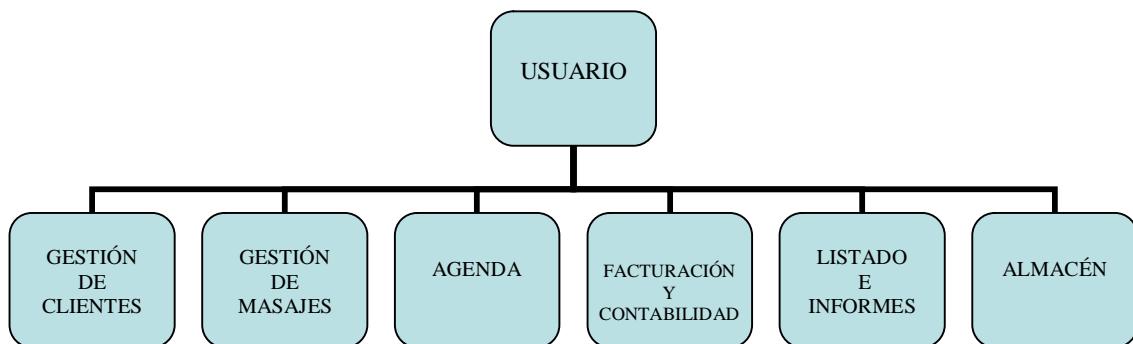
➤ Descripción General del Sistema:

- Contexto del sistema

Por todo lo argumentado en la descripción del sistema se observa que pertenece dentro del sistema por lo que salvo que el cliente lo requiera se cancela el estudio y desarrollo de todo lo que se encuentre fuera del sistema.

- Estructura organizativa

Ya que solo existe un único usuario que interactúe con el sistema, reflejamos con una gráfica como se comporta el sistema.



➤ Catalogo de requisitos

Como se ha comentado anteriormente, al ser un proyecto secuencial, el catalogo de requisitos queda intacto al no haber añadido ninguna restricción por sincronización con otros proyectos.

➤ **Catalogo de usuarios**

En concreto vamos a tener un único usuario que interactúe con la aplicación, y para mantener la privacidad se le concederá un ‘nombre y una ‘contraseña’ como administrador del sistema. Aún así le daremos la posibilidad de crear distintos tipos de usuarios con distintos tipos de funcionalidad para no dejar obsoleto la aplicación, pero todos ellos dependerán del administrador, el cual podrá realizar todo tipo de cambios en ellos.

Tarea 1.3: Especificación del alcance del sistema.

➤ **Catalogo de objetivos**

El objetivo de esta tarea es determinar hasta que punto de detalle se va a investigar en el estudio de la empresa. En este caso en concreto se ha decidido realizar la actividad número dos, que contempla la situación actual de la empresa con el objetivo de aprender más el funcionamiento de la organización de la empresa y obtener ideas de desarrollo, para satisfacer en mejor medida los requisitos.

➤ **Catalogo de usuarios**

En principio sólo existirá un único usuario que será el administrador pero se le dará la posibilidad de realizar nuevos roles con un subconjunto de funciones para no dejar obsoleta la aplicación en caso de aumento de plantilla.

➤ **Plan de trabajo**

La realización del plan de trabajo se ha hecho mediante la aplicación Microsoft Project. En la que se estima la duración de cada tarea y su planificación. Esta grafica podemos observarla en la planificación.

Actividad 2: Estudio de la situación actual.

Estamos ante los sistemas de información que dispone la empresa al inicio del estudio. En esta actividad se realiza un estudio de valoración de los sistemas que se ven afectados, y dependiendo de la valoración así se realizará el nivel de detalle del estudio. Si es necesario se realiza un equipo de trabajo. Como resultado obtenemos un informe de la eficiencia de los sistemas generando con problemas y mejoras.

Tarea 2.1: Valoración del estudio de la situación actual.

➤ Descripción de la situación actual:

Actualmente la organización consta de distintos sistemas de información que tiene indirectamente un desorden en la estructura informativa y además no posee tanta información, como la que se pretende conseguir, por incompatibilidad con los sistemas. Los sistemas de información son: un diario para el control de citas, una agenda para la información de los pacientes, Word para el desarrollo de presupuestos e informes y por último el Excel para gestionar la contabilidad.

Tarea 2.2: Identificación de los usuarios participantes en el estudio de la situación actual.

➤ Catálogo de usuarios

El realizar un proyecto fin de carrera conlleva que solo se dispone de un usuario para realizar el estudio, sin contar con otras opiniones y aspectos.

Tarea 2.3: Descripción de los sistemas de información existentes.

Actualmente posee un diario físico en el que se anotan las citas de los pacientes. El inconveniente de este sistema, es que el estar tan ligado a los pacientes a unas horas determinadas nos mueve a desplazamientos y anulaciones de citas. Comentado nuestro cliente el caos que provocaba meses con días festivos donde el cambio de citas era muy normal y era incompatible con un diario físico.

Con otro libro de agenda almacena la información de los pacientes que es mucho menor de lo que se va a conseguir. Además posee el inconveniente de las búsquedas a mano que son mucho más lentas que las procesadas por la aplicación.

En otro aspecto dispone de la herramienta de Office y mediante las dos aplicaciones de Word y Excel soluciona los siguientes problemas. Por Word realiza los presupuestos y facturas que solicitan los clientes pero está dificultado por la necesidad de buscar en la agenda la información y luego traspasar la información para cada presupuesto o factura tantas veces que haga falta. Con la hoja de cálculo Excel realiza la contabilidad, mezclando cobros y pagos con clientes y proveedores, para poder resaltar gastos e ingresos al final de mes. Como se puede comprobar, es muy complicado y difícil rescatar la información para realizar las estadísticas. Si añadimos la posibilidad de un gran número de pacientes la tarea se complica para la contabilidad.

Tarea 2.4: Realización del diagnóstico de la situación actual.

➤ Descripción de la situación actual

En el apartado anterior hemos comentado las faltas y deficiencias que se tiene en la situación actual. Es por esto, por lo que nuestro cliente requiere de una única aplicación y eliminar en todo lo posible todas esas carencias comentadas, y a través de las entrevistas realizadas hemos controlado estas sugerencias en los requisitos.

Actividad 3: Definición de requisito del sistema.

Aquí determinamos los requisitos generales (condición o capacidad que debe cumplir o poseer un sistema) y son capturados mediante técnicas como: entrevistas, reuniones, grupos de trabajo, equipos de discusión, cuestionarios, prototipos, etc. Se le asignan prioridades y se añaden al catálogo de requisitos.

Tarea 3.1: Identificación de las directrices técnicas y de gestión.

➤ Catalogo de normas

Identificador: Análisis y diseño.

Descripción: La elaboración del análisis y diseño está construida mediante las técnicas de catalogación de Métrica 3, que se basa en el lenguaje de especificación UML y una estructura orientada a objetos. El diseño de los diagramas está realizado mediante la herramienta MagicDraw.

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Identificador: Implementación.

Descripción: El entorno de programación es Microsoft Visual Studio 2005, por ser un entorno de desarrollo rápido de software y especial en la programación visual.

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Identificador: Arquitectura.

Descripción: Estará delimitado en una arquitectura de sistema local.

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Identificador: Base de datos

Descripción: Usaremos MySQL, por ser de libre distribución y ser un gran gestor local

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Identificador: Modelo de proceso.

Descripción: Entre la variedad de los distintos modelos que existen, se utilizará el iterativo incremental, debido a que realizaremos varias versiones en las que nuestro cliente nos podrá ir corrigiendo deficiencias o carencias de la aplicación, y además porque la funcionalidad con más prioridad, al ser desarrollada antes, recibe más pruebas. Con lo que la aplicación estará bien probada ante cualquier fallo.

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Identificador: Entrega

Descripción: En principio no existe ningún plazo de entrega.

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Tarea 3.2: Identificación de requisitos.

➤ Identificación de requisitos

Tras varias sesiones de entrevista con el cliente, respetando los horarios laborables y con acuerdo mutuo para las citas, hemos identificado todos los requisitos que requiere el cliente. Como el modelo de proceso es el iterativo incremental, estaremos siempre en contacto tras cualquier avance para asegurar que el producto final satisfaga lo máximo posible.

Tarea 3.3: Catalogación de requisitos.

➤ Catálogo de requisitos

El catálogo de requisitos se muestra a continuación y contiene los requisitos funcionales, seguridad o de datos según sea el requisito de agenda, cliente, masaje. De esta forma se ve a modo general el catálogo reuniendo requisitos comunes y además se describen el funcionamiento.

Nº	REQUISITO	TIPO	CONCEPTO	PRIORIDAD
1	Insertar Paciente	Funcional	Según la plantilla, se podrá crear un paciente con todos los datos que lo compone.	Alta
2	Eliminar Paciente	Funcional	La eliminación será de forma lógica y no física, para poder rescatar los datos en un futuro.	Alta
3	Modificar Paciente	Funcional	Modificar datos de un paciente que ya está dado de alta.	Alta
4	Consultar Paciente	Funcional	Dado un cliente de alta, mostrar sus datos personales, clínicos y otros.	Alta
5	Insertar Proveedor	Funcional	Insertar un proveedor junto con sus catálogos de artículos.	Alta
6	Eliminar Proveedor	Funcional	Eliminar de forma lógica un proveedor o algunos de sus artículos.	Alta
7	Modificar Proveedor	Funcional	Modificar los datos de un proveedor o bien sus artículos.	Alta
8	Consultar Proveedor	Funcional	Mostrar los datos de un proveedor junto con sus artículos.	Alta
9	Insertar Técnica	Funcional	Insertar una nueva técnica de masaje.	Alta
10	Eliminar	Funcional	Eliminar de forma lógica una	Alta

	Técnica		técnica.	
11	Modificar Técnica	Funcional	Modificar una técnica existente.	Alta
12	Consultar Técnica	Funcional	Mostrar los datos de una técnica.	Alta
13	Insertar Zona de Masaje	Funcional	Insertar una nueva zona de masaje.	Alta
14	Eliminar Zona de Masaje	Funcional	Eliminar de forma lógica una zona de masaje que ha está insertada.	Alta
15	Modificar Zona de Masaje	Funcional	Modificar una zona de masaje existente.	Alta
16	Consultar Zona de Masaje	Funcional	Mostrar los datos referentes a una zona de masaje	Alta
17	Insertar Masaje	Funcional	Insertar un nuevo masaje que se puede aplicar.	Alta
18	Eliminar Masaje	Funcional	Eliminar de modo lógico un masaje.	Alta
19	Consultar Masaje	Funcional	Mostrar los datos de un masaje que ha está insertado	Alta
20	Modificar Masaje	Funcional	Modificar los datos referentes a un masaje determinado.	Alta
21	Insertar Factura	Funcional	Crear una factura nueva	Alta
22	Modificar Factura	Funcional	Modificar una factura existente	Alta
23	Consultar Factura	Funcional	Ver el contenido de una factura que ya esta creada.	Alta
24	Insertar Presupuesto	Funcional	Crear un presupuesto nuevo	Alta
25	Modificar Presupuesto	Funcional	Modificar un presupuesto existente.	Alta
26	Consultar Presupuesto	Funcional	Ver el contenido de un presupuesto que ya está creado	Alta
27	Asignar cita	Funcional	Asignar una cita a un paciente.	Alta
28	Eliminar cita	Funcional	Anular la cita a un paciente.	Alta

29	Datos de la consulta	Funcional	Modificar y consultar los datos de la consulta.	Alta
30	Acceso	Seguridad	Modificar el acceso al sistema	Alta
31	Copia de seguridad	Seguridad	Importar o exportar la base de datos	Alta
32	Listado	Funcional	Seleccionar un listado determinado de la base de datos.	Alta
33	Almacén	Funcional	Modificar el almacén por la entrada y salida de productos	Alta
34	Compra	Funcional	Registrar la compra de artículos a los proveedores	Alta
35	Venta	Funcional	Registrar la venta de artículos por los pacientes.	Alta
36	Legibilidad y fácil acceso.	No Funcional	La interacción con el usuario y la interfaz entre ventanas sean de agrado y sencillas.	Alta

Sobre los requisitos geográficos no se dispone ninguno por ser la aplicación local y referente a los requisitos tecnológicos quedan descartados por ser una aplicación sin ejecuciones complejas.

Actividad 4: Estudio de alternativas de solución.

En esta actividad se basa en realizar diversas alternativas que den una solución y responda de forma satisfactoria a los requisitos planteados anteriormente.

Tarea 4.1: Preselección de alternativas de solución.

➤ Descomposición inicial del sistema en subsistemas.

Tras el estudio del sistema se intenta realizar una descomposición minimizando las ligaduras entre subsistemas para evitar problemas de redundancia o conflictos de datos. Nótese que esta descomposición sería de mayor utilidad en un proyecto formado por grupos para repartir el trabajo e ir avanzando en paralelo.

Subsistema 1: Estaría formado por la funcionalidad para gestionar las fichas de los clientes, con las cuatro operaciones básicas de inserción, consulta, modificación o borrado.

Subsistema 2: Se trata del control de la agenda. Una gestión de citas en la que sea posible dar citas, anularlas y cambiarlas de fecha.

Subsistema 3: Sería el encargado del mantenimiento de toda la variedad y posibilidades de masajes que se ofertan al paciente.

Subsistema 4: Compuesto por la contabilidad y facturas de la consulta, control de almacén y la elaboración de informes y listados.

➤ Alternativas de solución a estudiar.

Debido a nuestro cliente, que requiere una aplicación muy personal y concreta, nos vemos obligado a rechazar la cualquier posible solución por medio de la adquisición de software; por lo que en el tema sobre estudios de soluciones, estamos limitados al

desarrollo de la aplicación sobre un entorno cerrado. Quedan anuladas las alternativas de solución. (No se si poner que como es un proyecto fin de carrera queda excluido también la posibilidad de trabajo en grupo con otros sistemas parecidos para trabajar en paralelo o una combinación de las dos ideas propuestas).

Tarea 4.2: Descripción de las alternativas de solución.

Por lo argumentado anteriormente el catálogo de requisitos no sufre ninguna variación.

La aplicación se realizará mediante la programación orientada a objetos. Nos vemos obligados a realizar los diagramas de casos de uso y el diagrama de clases (modelo de negocio y de dominio).

Los diagramas de caso de uso documentan el comportamiento del sistema desde un punto de vista del usuario, por tanto los casos de usos determinan los requisitos funcionales del sistema. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con nuestro cliente. Los casos de uso deben especificar un comportamiento deseado, pero nunca deben imponer cómo se llevará a cabo dicho comportamiento. Para ello se utilizará los escenarios, que son interacciones entre el sistema y los actores, descritos mediante una secuencia de mensajes. Esto se realizará en el ASI.

El diagrama de clase se corresponde con los sustantivos más importantes que hace referencia al ámbito del sistema. Cada clase define unos atributos que son datos asociados a los objetos y sus métodos. Las clases entre sí están relacionadas por asociación, herencia o agregación normalmente.

Los diagramas de caso de uso y de clases están en el capítulo de diagramas.

Actividad 5: Valoración de las alternativas.

Realizadas las alternativas se comprueba una valoración de ellas considerando el impacto en la empresa, desde los puntos de vista tecnológico, organizativo, y posibles beneficios y gastos asociados. Se realiza un análisis de riesgos intentando minimizarlos.

Tarea 5.1: Estudio de la inversión.

➤ **Valoración de la alternativa (única solución).**

- Impacto en la organización de la alternativa.

El desarrollo de dicho software suministrará a nuestro un cliente un control total de su consulta, creando un entorno más amigable para el engoroso tema de las citas, permitirá además de tener más información sobre los clientes, un continuo seguimiento de su evolución, y conseguirá de una forma más clara tener una idea de cuales son los beneficios y gastos que aporta la organización.

De esta forma el cliente solo se tiene que preocupar de organizar el día a día de una forma muy sencilla y no tendrá que estar consultando movimiento de hojas para poder trabajar.

- Coste / Beneficio de la alternativa.

Costes del sistema:

- Adquisición de hardware: 1 Ordenador personal e impresora.
- Adquisición de software. MagicDraw, Visual C#, MySQL, Word
- Coste de desarrollo: sueldo mensual de 1200€
- Gastos de comunicación: Internet y llamadas al cliente.
- Gastos de instalación: ninguno.
- Gastos del mantenimiento del sistema: no se posee el servicio.
- Gastos de formación: ninguno.

- Gastos de material: material de oficina general (papel, tinta de impresora, complementos de escritorio).
- Gastos de inexperiencia: gastos que siguen una curva decremental al ser las primeras etapas del proyecto, más elevadas, por falta de conocimientos.
- Costes financieros:

Estimación del proyecto: 6 meses.

Coste mensual: 1200€

Coste total: 7200€

Tarea 5.2: Estudio de los riesgos.

El estudio de riesgos no es necesario por dos motivos:

- No existe la incertidumbre al estar en pleno contacto con nuestro cliente y se tienen claras las ideas de que es lo que se busca.
- La complejidad del sistema es inexistente por ser un sistema de información muy personal en las funciones requeridas, y no necesitan la realización de algoritmos complejos, ni relación con otras bases de datos, por lo que no habrá incompatibilidad con nada al ser todo creado desde cero.

Actividad 6: Selección de la solución.

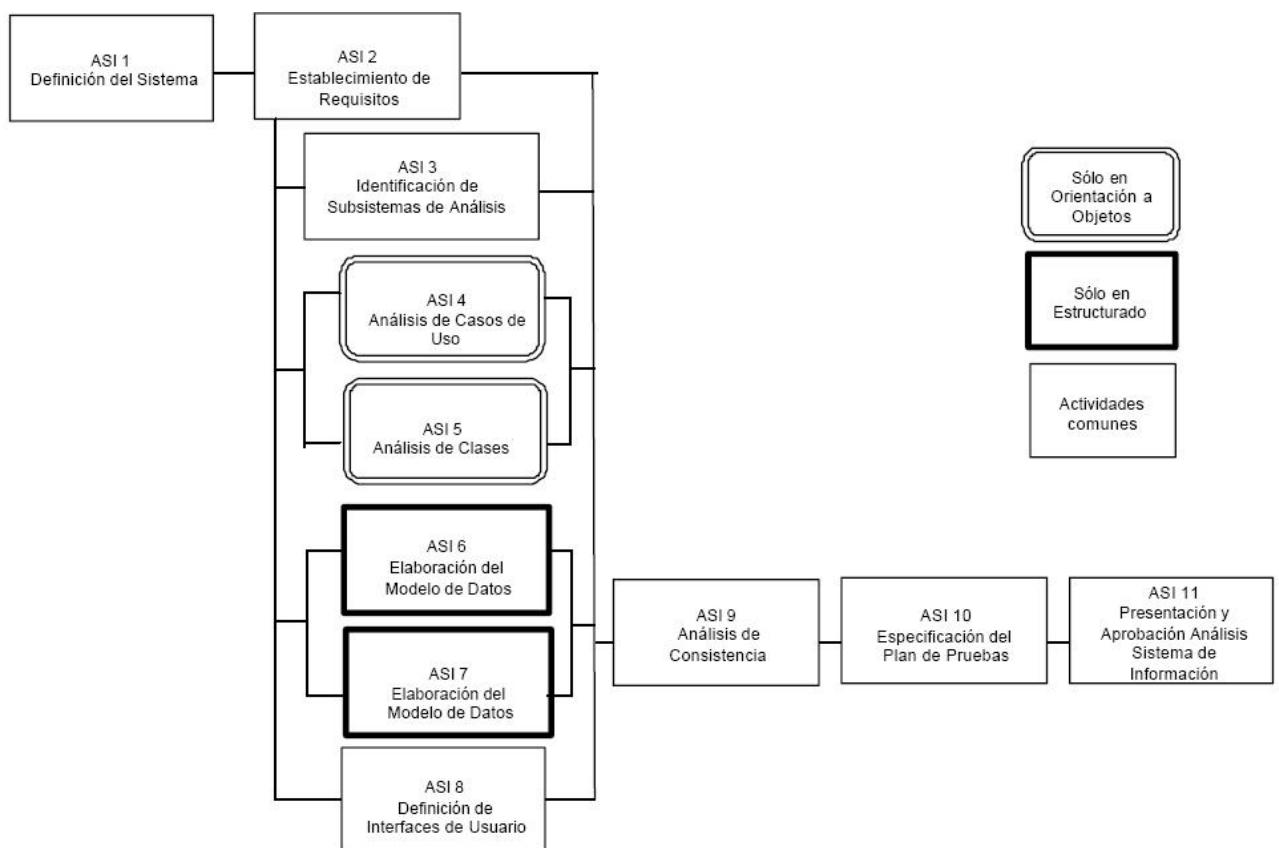
Como hemos visto anteriormente, solo hemos desarrollado una solución viable y es la que vamos a desarrollar.

Esta actividad no tiene sentido porque se trataría de todas las soluciones aportadas, estudiar cual es la de mayor viabilidad en conjunto entre el cliente y el equipo de desarrollo, y en caso de no encontrar una adecuada a las necesidades comentadas volver a elaborar una nueva partida de soluciones a estudiar.

2.3 ANALISIS DEL SISTEMA DE INFORMACIÓN

El objetivo del ASI es obtener de forma detallada las especificaciones del sistema, que satisfaga las necesidades de información y sirva de apoyo vital para el diseño del software.

Las tareas a realizar se muestran en el siguiente gráfico, hay que indicar que no todas las pruebas se realizan (ya que van a depender en gran parte de la envergadura del sistema de información) y como el diseño se realizará de forma estructurada u orientada a objetos, se realizarán las actividades 6 y 7 u 8 y 9 respectivamente.



Actividad ASI 1: Definición del sistema.

En esta actividad haremos una descripción del sistema, viendo cual es el alcance, estableciendo la interfaz con otros sistemas e indicando los usuarios más importantes.

Tarea ASI 1.1: Determinación del alcance del sistema.

Basándonos en el modelo de procesos, indicaremos cuales son los procesos que pertenecen al sistema y cuales son las entidades externas.

➤ Catalogo de requisitos

Al catálogo de requisitos le vamos a añadir los siguientes que se han determinado mediante las reuniones con el cliente.

Nº	REQUISITO	TIPO	CONCEPTO	PRIORIDAD
37	Insertar Paciente	Funcional	Según la plantilla, se podrá crear un paciente con todos los datos que lo compone.	Alta
38	Eliminar Paciente	Funcional	La eliminación será de forma lógica y no física, para poder rescatar los datos en un futuro.	Alta
39	Modificar Paciente	Funcional	Modificar datos de un paciente que ya está dado de alta.	Alta
40	Consultar Paciente	Funcional	Dado un cliente de alta, mostrar sus datos personales, clínicos y otros.	Alta
41	Insertar Proveedor	Funcional	Insertar un proveedor junto con sus catálogos de artículos.	Alta
42				
43	Insertar Paciente	Funcional	Según la plantilla, se podrá crear un paciente con todos los datos que lo	Alta

			compone.	
44	Eliminar Paciente	Funcional	La eliminación será de forma lógica y no física, para poder rescatar los datos en un futuro.	Alta
45	Modificar Paciente	Funcional	Modificar datos de un paciente que ya está dado de alta.	Alta
46	Consultar Paciente	Funcional	Dado un cliente de alta, mostrar sus datos personales, clínicos y otros.	Alta

➤ [Glosario](#)

El glosario está compuesto por términos relacionados con el ámbito de negocio. El glosario nos ayudará a definir las especificaciones del sistema de información con una precisión mas fina.

Usuario: Será el individuo que interaccione con la aplicación.

Paciente: Persona a la que va destinada los tratamientos, facturas, asistencias e historiales.

Datos clínicos: son los datos médicos que están asociados a un único paciente.

Masaje: Características biológicas y anatómicas de cómo incidir un masaje.

Zona de masaje: localizaciones del cuerpo donde va dirigido el masaje.

Técnica: Aparatos y cremas necesarias para atacar el problema.

Tratamiento: Es un conjunto de las distintas posibilidades de los 3 términos comentados anteriormente.

Proveedor: Empresa que nos proporciona los productos necesarios para impartir nuestros servicios.

Artículo: Producto proveniente de un proveedor con características estándar.

Cita: Fecha y hora en la que se espera un paciente para la realización de un tratamiento.

Caja: estado físico de la situación económica de la empresa.

Factura: Documento que defiende la venta de los servicios a un paciente.

Historial: Estado físico del paciente antes de iniciar un tratamiento.

Pruebas: Documento que refleja la respuesta de un paciente ante un tratamiento.

Registro: Forma de acreditar que tenemos permiso para acceder a la aplicación.

Manual de usuario: Documentación de ayuda sobre el manejo de la aplicación.

Almacén: Estado físico del tipo y cantidad de productos que se encuentran en el almacén.

➤ **Modelo de Negocio**

Ver diagrama de casos de uso en el capítulo de diagramas.

Tarea ASI 1.2: Identificación del entorno Tecnológico.

En esta tarea comentaremos, en alto nivel, cual es el entorno tecnológico que se necesita para responder a las necesidades de información, detallando cuales son sus condiciones y restricciones.

➤ **Catálogo de requisitos.**

Igual que en la tarea ASI 1.1

➤ **Descripción general del entorno tecnológico del sistema**

En la situación en la que nos encontramos, no vamos a estar limitado por la tecnología, porque al poseer un ordenador personal en plataforma Windows conectado a una impresora, podremos utilizar por completo la aplicación al ser de ámbito local y no necesitar ni red ni conexión de Internet.

Tarea ASI 1.3: Especificación de Estándares y Normas.

Esta tarea previene la definición de estándares y normas que hay que tener en cuenta en el periodo de desarrollo del proceso.

➤ **Catalogo de Normas**

Se ha de verificar los apartados expuestos en la tarea EVS 3.1.

Identificador: Entrega

Descripción: Se ha estimado que la entrega final del proyecto sea cercana al mes de junio del 2007.

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Identificador: Seguridad

Descripción: La obligada identificación ante el sistema permite medidas de seguridad para asegurar la privacidad de la información contenida en la aplicación. El usuario que disponga de la identificación podrá acceder al total de la aplicación.

Autor: Rafael Avilés Arroyo.

Estado: Aprobado.

Tarea ASI 1.4: Identificación de los usuarios participantes y finales.

En esta tarea, indicaremos cuales son los usuarios participantes y finales para la obtención de los requisitos finales del sistema.

➤ **Catalogo de usuarios**

Solo va a existir un único usuario final, que controle toda la operatividad del sistema.

➤ **Plan de trabajo**

Igual que en la tarea EVS 1.3

ACTIVIDAD ASI 2: ESTABLECIMIENTO DE REQUISITOS.

En esta actividad trabajaremos con definición, análisis y validación de los requisitos. El objetivo es obtener un catalogo detallado de los requisitos para verificar que los productos generados se ajustan a los requisitos de usuario.

Tarea ASI 2.1: Obtención de requisitos.

➤ **Catalogo de requisitos:**

No se altera y queda como en la tarea ASI 1.1

➤ **Modelo de Casos de Uso**

Ver diagrama en el capítulo de diagramas.

Tarea ASI 2.2: especificación de casos de uso.

➤ **Catalogo de requisitos**

Como en la tarea ASI 2.1

➤ **Modelo de Casos de Uso**

Como en la tarea ASI 2.1

➤ Especificación de Casos de Uso

El conjunto formado por el catálogo de requisitos y el modelo de casos de uso, es el que nos da la ayuda en el análisis de la interacción con el usuario.

Este análisis consta que para cada caso de uso se crea dos tipos de escenarios en los que participa el usuario. El primero de ellos es el normal. Ocurre cuando en la interacción del usuario con el programa no se produce ninguna incidencia. El segundo escenario es cuando proceden situaciones anormales. Otra opción que toma el escenario de excepción es cuando el usuario está en una ventana para eliminar alguna información y este no posee privilegios o permisos suficientes para la operación. Esta opción no ocurrirá en nuestro proyecto por poseer un único usuario.

Habrá situaciones en los que los casos de uso solo posean el escenario normal, debido a que no sea posible la existencia de situaciones que provoquen errores.

A continuación se exponen los escenarios, pero debemos tener en cuenta que se requiere realizar además los escenarios mediante la aplicación de MagicDraw que están comentadas posteriormente.

- Paciente
 - Insertar paciente
 - Escenario normal
 - El usuario quiere dar de alta
 - El usuario introduce los datos en la ficha del nuevo paciente.
 - El usuario solicita insertar al paciente
 - El paciente ha sido dado de alta
 - Escenario excepción
 - El usuario quiere dar de alta
 - El usuario introduce los datos en la ficha del nuevo paciente.
 - El usuario solicita insertar al paciente
 - El paciente no es dado de alta por error en los datos o falta de información.
 - Búsqueda de paciente
 - Escenario normal
 - El usuario quiere buscar al paciente.

- El usuario utiliza el filtro.
- El usuario obtiene un listado menor debido al filtro.
- El usuario selecciona el paciente deseado.

○ Modificar paciente

- Escenario normal
 - El usuario quiere modificar al paciente.
 - Los datos son mostrados al usuario.
 - El usuario modifica los datos.
 - El usuario quiere guardar los cambios
 - Los cambios son guardados con éxito.
- Escenario excepción
 - El usuario quiere modificar al paciente.
 - Los datos son mostrados al usuario.
 - El usuario modifica los datos.
 - El usuario quiere guardar los cambios.
 - Los cambios no son guardados por error en los datos.

○ Consultar Paciente

- Escenario normal
 - El usuario quiere consultar al paciente.
 - Los datos son mostrados al usuario.
 - El usuario cierra la ventana.

○ Eliminar paciente

- Escenario normal
 - El usuario quiere eliminar al paciente.
 - El usuario selecciona el paciente.
 - El usuario confirma la eliminación.
 - El paciente es eliminado.
- Escenario normal2
 - El usuario quiere eliminar al paciente.
 - El usuario selecciona el paciente.
 - El usuario confirma la eliminación.
 - El paciente no es eliminado.

- Tratamiento

- Buscar tratamiento

- Escenario normal

- El usuario quiere buscar un tratamiento
 - El usuario utiliza el filtro.
 - El usuario selecciona un tratamiento.
 - El usuario cierra la ventana

- Insertar tratamiento

- Escenario normal

- El usuario quiere insertar un tratamiento.
 - El usuario rellena los datos solicitados.
 - El usuario quiere guardar los datos.
 - El tratamiento ha sido guardado con éxito.

- Escenario excepción

- El usuario quiere insertar un tratamiento.
 - El usuario rellena los datos solicitados.
 - El usuario quiere guardar los datos.
 - El tratamiento no es guardado por error en los datos.

- Consultar tratamiento

- Escenario normal

- El usuario quiere consultar un tratamiento.
 - El usuario selecciona el tratamiento.
 - Se muestra los datos del tratamiento.
 - El usuario cierra la ventana.

- Modificar tratamiento

- Escenario normal

- El usuario quiere modificar un tratamiento.
 - Se muestra los datos del tratamiento.
 - El usuario rellena los datos solicitados.
 - El usuario quiere modificar los datos.
 - El tratamiento ha sido modificado.

- Escenario excepción:

- El usuario quiere modificar un tratamiento.

- Se muestra los datos del tratamiento.
 - El usuario rellena los datos solicitados.
 - El usuario quiere modificar los datos.
 - El tratamiento no es modificado por error en los datos.
- Eliminar tratamiento
 - Escenario normal
 - El usuario quiere eliminar el tratamiento.
 - El usuario selecciona el tratamiento.
 - El usuario confirma la eliminación.
 - El tratamiento es eliminado.
 - Escenario normal2
 - El usuario quiere eliminar el tratamiento.
 - El usuario selecciona el tratamiento.
 - El usuario confirma la eliminación.
- Zonas
 - Buscar Zonas
 - Escenario normal
 - El usuario solicita buscar zonas.
 - El usuario visualiza todas las posibles zonas.
 - El usuario selecciona la zona deseada.
 - El usuario cierra la ventana.
 - Insertar Zonas
 - Escenario normal
 - El usuario quiere insertar una zona nueva.
 - El usuario rellena los campos.
 - El usuario quiere almacenar la zona nueva.
 - La zona nueva es guardada.
 - Escenario excepción
 - El usuario quiere insertar una zona nueva.
 - El usuario rellena los campos.
 - El usuario quiere almacenar la zona nueva.
 - La operación es cancelada por error en los datos.

- Consultar zonas
 - Escenario normal
 - El usuario quiere ver la información de una zona.
 - El usuario selecciona una zona específica.
 - Se muestran los datos de la zona deseada.
 - El usuario cierra la ventana.
- Modificar zonas
 - Escenario normal
 - El usuario quiere modificar una zona.
 - Se muestran los datos de la zona.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - Se almacena los cambios en la zona.
 - Escenario excepción
 - El usuario quiere modificar una zona.
 - Se muestran los datos de la zona.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - La zona no se modifica por error en los datos.
- Eliminar zona
 - Escenario normal
 - El usuario quiere eliminar una zona.
 - El usuario selecciona la zona a eliminar.
 - El usuario confirma la eliminación.
 - La zona es eliminada.
- Buscar técnica
 - Escenario normal
 - El usuario solicita buscar técnica.
 - El usuario visualiza todas las posibles técnicas.
 - El usuario selecciona la técnica deseada.
 - El usuario cierra la ventana.

- Insertar técnicas
 - Escenario normal
 - El usuario quiere insertar una técnica nueva.
 - El usuario rellena los campos.
 - El usuario quiere almacenar la técnica nueva.
 - La técnica nueva es guardada.
 - Escenario excepción
 - El usuario quiere insertar una técnica nueva.
 - El usuario rellena los campos.
 - El usuario quiere almacenar la técnica nueva.
 - La técnica nueva no es guardada por error en los datos.
- Consultar técnicas
 - Escenario normal
 - El usuario quiere ver la información de una técnica.
 - El usuario visualiza la técnica.
 - El usuario cierra la ventana.
- Modificar técnica
 - Escenario normal
 - El usuario quiere modificar una técnica.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - Se almacena los cambios en la técnica.
 - Escenario excepción
 - El usuario quiere modificar una técnica.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - La zona no se modifica por error en los datos.
- Eliminar técnica.
 - Escenario normal
 - El usuario quiere eliminar una técnica.
 - El usuario selecciona la técnica a eliminar.
 - El usuario confirma la eliminación.
 - La técnica es eliminada.

- **Mensaje**

- **Buscar mensaje**

- **Escenario normal**

- El usuario solicita buscar mensaje.
 - El usuario visualiza todos los posibles mensajes.
 - El usuario utiliza el filtro.
 - Se muestran los mensajes que cumplen el filtro.
 - El usuario selecciona el mensaje deseado.
 - El usuario cierra la ventana.

- **Insertar mensaje**

- **Escenario normal**

- El usuario quiere insertar un mensaje nuevo.
 - El usuario rellena los campos.
 - El usuario quiere almacenar el nuevo mensaje.
 - El mensaje nuevo es guardado.

- **Escenario excepción**

- El usuario quiere insertar un mensaje nuevo.
 - El usuario rellena los campos.
 - El usuario quiere almacenar el mensaje nuevo.
 - El mensaje nuevo no es guardado por error en los datos.

- **Consultar mensajes**

- **Escenario normal**

- El usuario quiere ver la información de un mensaje.
 - El usuario visualiza el mensaje.
 - El usuario cierra la ventana.

- **Modificar mensaje**

- **Escenario normal**

- El usuario quiere modificar un mensaje.
 - Se muestra el contenido del mensaje.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - Se almacena los cambios en el mensaje.

- **Escenario excepción**

- El usuario quiere modificar un masaje.
- Se muestra el contenido del masaje.
- El usuario modifica los campos.
- El usuario quiere guardar los cambios.
- El masaje no se modifica por error en los datos.
- Eliminar masaje.
 - Escenario normal
 - El usuario quiere eliminar un masaje.
 - El usuario selecciona el masaje a eliminar.
 - El usuario confirma la eliminación.
 - El masaje es eliminado.
- Buscar masaje
 - Escenario normal
 - El usuario solicita buscar masaje.
 - El usuario visualiza todos los posibles masajes.
 - El usuario selecciona el masaje deseado.
 - El usuario cierra la ventana.
- Bono
 - Insertar bono
 - Escenario normal
 - El usuario quiere insertar un bono nuevo.
 - El usuario rellena los campos.
 - El usuario quiere almacenar el nuevo bono.
 - El bono nuevo es guardado.
 - Escenario excepción
 - El usuario quiere insertar un bono nuevo.
 - El usuario rellena los campos.
 - El usuario quiere almacenar el bono nuevo.
 - El bono no es guardado por error en los datos.
 - Consultar bono
 - Escenario normal
 - El usuario quiere ver la información de un bono.
 - El usuario utiliza el filtro de bonos.

- El usuario selecciona el bono.
 - El usuario visualiza el bono.
 - El usuario cierra la ventana.
- Modificar bono
 - Escenario normal
 - El usuario quiere modificar un bono.
 - Se muestra la información del bono.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - Se almacena los cambios en el bono.
 - Escenario excepción
 - El usuario quiere modificar un bono.
 - Se muestra la información del bono.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - El bono no se modifica por error en los datos.
 - Eliminar bono.
 - Escenario normal
 - El usuario quiere eliminar un bono.
 - El usuario selecciona el bono a eliminar.
 - El usuario confirma la eliminación.
 - El bono es eliminado.
- Historial
 - Insertar Historial
 - Escenario normal
 - El usuario quiere insertar un historial nuevo.
 - El usuario rellena los campos.
 - El usuario quiere almacenar el nuevo historial.
 - El historial es guardado.
 - Escenario excepción
 - El usuario quiere insertar un historial nuevo.
 - Se cancela la operación porque el usuario no ha seleccionado una asistencia.

- Consultar historial
 - Escenario normal
 - El usuario quiere ver la información de un historial.
 - El usuario selecciona el historial.
 - Se muestra la información del historial.
 - El usuario cierra la ventana.
- Modificar historial
 - Escenario normal
 - El usuario quiere modificar un historial.
 - Se muestra la información actual del historial.
 - El usuario modifica los campos.
 - El usuario cancela la operación
 - No se modifican los valores y se cierra la ventana.
 - Escenario excepción
 - El usuario quiere modificar un historial.
 - Se muestra la información del historial.
 - El usuario modifica los campos.
 - El usuario quiere guardar los cambios.
 - Los cambios no son almacenados por no ser correctos.
- Eliminar historial.
 - Escenario normal
 - El usuario quiere eliminar un historial.
 - El usuario selecciona el historial a eliminar.
 - El usuario confirma la eliminación.
 - El historial es eliminado.
- Prueba
 - Insertar Prueba
 - Escenario normal
 - El usuario quiere insertar una prueba.
 - El usuario rellena los campos.
 - El usuario quiere almacenar la prueba.
 - La prueba es almacenada.

- Escenario excepción
 - El usuario quiere insertar una prueba.
 - Se cancela la operación porque el usuario no ha seleccionado una asistencia o un historial.
- Consultar prueba
 - Escenario normal
 - El usuario quiere ver la información de una prueba.
 - El usuario selecciona una prueba determinada.
 - Se muestra la información de la prueba.
 - El usuario cierra la ventana.
 - Eliminar prueba.
 - Escenario normal
 - El usuario quiere eliminar una prueba.
 - El usuario selecciona la prueba.
 - El usuario cancela la eliminación.
 - La prueba no se elimina.
 - Factura
 - Insertar Factura
 - Escenario normal
 - El usuario quiere crear una factura nueva.
 - El usuario rellena todos los campos.
 - El usuario quiere almacenar la factura.
 - La factura es almacenada.
 - Escenario excepción
 - El usuario va a crear una factura.
 - Se cancela la operación al faltar datos necesarios.
 - Consultar factura
 - Escenario normal
 - El usuario quiere ver el contenido de una factura.
 - El usuario selecciona la factura.
 - Se muestra la información de la factura.
 - El usuario cierra la ventana.

- Eliminar factura.
 - Escenario normal
 - El usuario quiere eliminar una factura.
 - El usuario localiza la factura a eliminar.
 - El usuario confirma la eliminación.
 - La factura es eliminado.
- Imprimir
 - Imprimir documento
 - Escenario normal
 - El usuario desea imprimir un documento.
 - Se muestra el contenido del documento
 - El usuario decide imprimir.
 - El documento se imprime.
 - Escenario excepción
 - El usuario desea imprimir un documento.
 - Se muestra el contenido del documento
 - El usuario decide imprimir.
 - No se puede imprimir el documento al no detectar ninguna impresora.
- Caja
 - Insertar Caja
 - Escenario normal
 - El usuario quiere añadir una línea en caja.
 - El usuario rellena todos los campos.
 - El usuario cancela la operación.
 - La caja queda sin alterar.
- Almacén
 - Consultar Almacén
 - Escenario normal
 - El usuario quiere ver el almacén.
 - Se muestra el género que existe en el almacén.
 - El usuario decide imprimir el estado del almacén.
 - Se imprime el estado.

- Añadir Almacén
 - Escenario normal
 - El usuario quiere añadir género al almacén.
 - Se muestra el género que existe en el almacén.
 - El usuario añade nuevos géneros.
 - El usuario confirma el estado.
 - Escenario normal2
 - El usuario quiere añadir género al almacén.
 - Se muestra el género que existe en el almacén.
 - El usuario cancela la operación.
 - El almacén no se modifica.
- Eliminar Almacén
 - Escenario normal
 - El usuario quiere eliminar género del almacén.
 - Se muestra el género que existe en el almacén.
 - El usuario selecciona el género al eliminar.
 - El usuario confirma y se actualiza el estado.

Tarea ASI 2.3: Análisis de Requisitos.

➤ Catalogo de requisitos

Como en la Tarea ASI 1.1

➤ Modelo de Casos de Uso

Ver diagrama de casos de uso en el capítulo de diagramas.

➤ Especificación de Casos de Uso

Como en la tarea ASI 2.2

Tarea ASI 2.4: Validación de Requisitos.

Podemos asegurar que el conjunto formado por los requisitos especificados, que se encuentran en el catálogo de requisitos, son válidos, consistentes y completos.

➤ **Catálogo de requisitos**

Como en la Tarea ASI 1.1

➤ **Modelo de Casos de Uso**

Ver diagrama en el capítulo de diagramas.

➤ **Especificación de Casos de Uso**

Como en la tarea ASI 2.2

ACTIVIDAD ASI 3: IDENTIFICACIÓN DE LOS SUBSISTEMAS DE ANÁLISIS.

El objetivo de esta actividad es descomponer el sistema en subsistemas, para ayudar al análisis del sistema de información.

Tarea ASI 3.1: Determinación de Subsistemas de Análisis.

Para la descomposición en subsistemas nos basaremos especialmente en los procesos de negocios.

➤ **Descripción de subsistemas de análisis**

Como en la Tarea EVS 4.1.

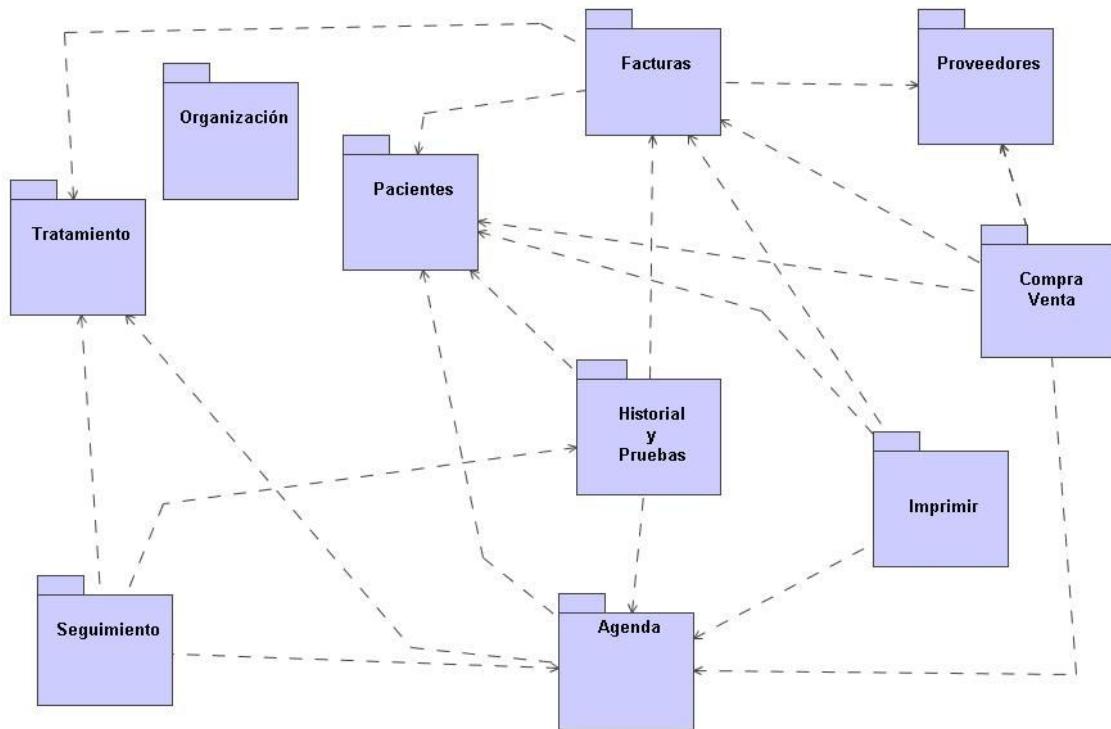
➤ **Descripción de interfaces entre subsistemas**

Subsistema	Caso de uso
Agenda	Gestión (Insertar cita, Consultar cita, Eliminar cita, Modificar cita) Recordatorio Tratamientos Pacientes
Compra Venta	Compra productos Almacén Venta productos Proveedores Pacientes Gastos internos
Facturas	Gestión (Insertar factura, Consultar factura, Eliminar factura, Modificar factura) Calcular factura Productos Tratamientos Proveedores
Historial Pruebas	Gestión (Insertar historial, Consultar historial, Eliminar historial, Modificar historial) Gestión (Insertar prueba, Consultar prueba, Eliminar prueba, Modificar prueba) Agenda
Imprimir	Facturas Agenda Pacientes Listados Proveedores Información Empresa
Organización	Información Empresa Acceso al sistema

	Validación Seguridad Copia de seguridad Restaurar copia
Pacientes	Gestión(Insertar Paciente, Consultar Paciente, Eliminar Paciente, Modificar Paciente) Validar paciente Búsqueda
Proveedores	Gestión(Insertar proveedor, Consultar proveedor, Eliminar proveedor, Modificar proveedor) Gestión(Insertar producto, Consultar producto, Eliminar producto, Modificar producto) Buscar proveedor Buscar producto Validar proveedor Validar producto
Seguimiento	Pruebas Agenda Tratamiento Pacientes Historial
Tratamiento	Gestión(Insertar tratamiento, Consultar tratamiento, Eliminar tratamiento, Modificar tratamiento) Gestión(Insertar bono, Consultar bono, Eliminar bono, Modificar bono) Gestión(Insertar técnica, Consultar técnica, Eliminar técnica, Modificar técnica) Gestión(Insertar masaje, Consultar masaje, Eliminar masaje, Modificar masaje) Gestión(Insertar zona, Consultar zona, Eliminar zona, Modificar zona) Búsqueda (Tratamiento, bono, zonas, masaje, técnica)

Esta organización nos proporciona una delimitación y nos evita redundancias de clases.

➤ **Diagrama de paquetes**



Se puede apreciar que hay un gran número de dependencias entre los distintos subsistemas, esto es porque muchas de las funciones de un paquete en particular, se requieren en otros subsistemas. Como resultado tenemos garantizado la reutilización que se busca en la programación orientada a objetos.

Tarea ASI 3.2: Integración de Subsistemas de Análisis.

Esta tarea nos ayudará para que en esta actividad evitemos la ausencia de duplicidad en los modelos realizados dentro de la actividad.

➤ **Descripción de subsistemas de análisis**

Como en la Tarea EVS 4.1

➤ **Descripción de interfaces entre subsistemas**

Como en la Tarea ASI 3.1

ACTIVIDAD ASI 4: ÁNALISIS DE CASOS DE USO.

En esta actividad, que es exclusiva para el Análisis orientado a objetos, consiste en averiguar cuales son las clases que tienen objetos necesarios para crear un caso de uso y describir su comportamiento mediante la interacción con los objetos comentados.

Tarea ASI 4.1: Identificación de clases asociadas a cada caso de uso

➤ *Modelo de Clases de Análisis*

Ahora determinaremos las clases que se necesitan respecto la funcionalidad de los casos de uso.

Ver en el capítulo de diagramas.

Tarea ASI 4.2 Descripción de la interacción de objetos.

Para cada caso de uso se elaborara dos diagramas de secuencia que hace referencia al escenario normal y al escenario de excepción. Al recordar el gran número de casos de uso y sabiendo que se deben de realizar dos diagramas por caso de uso se omitirán los diagramas comunes como consultar, insertar o eliminar.

Ver diagrama de secuencias y maquina de estados en el capítulo de diagramas.

ACTIVIDAD ASI 5: ANÁLISIS DE CLASES.

En esta actividad, que también solo es exclusiva para el caso orientado a objetos, debemos definir cada una de las clases que ha surgido, identificando las relaciones entre ellas y sus atributos.

Tarea ASI 5.1: Identificación de responsabilidades y atributos.

En esta tarea hay que ver cuales son las responsabilidades y atributos más relevantes de una clase.

Las responsabilidades de una clase indican cual es la funcionalidad y están determinadas por sus objetos dentro de los distintos casos de uso. A través de las responsabilidades podremos identificar operaciones relevantes a la clase.

En cambio los atributos determinan propiedades, y se identifican por estar muy relacionados con las responsabilidades.

Ver diagrama de clases en el capítulo de diagramas.

Tarea ASI 5.2- Identificación de asociaciones y agregaciones.

En esta tarea identificamos los mensajes que existen entre los objetos del diagrama de interacción con el fin de definir qué asociaciones hay entre las clases correspondientes.

Normalmente las asociaciones están muy ligadas con las expresiones verbales incluidas en las especificaciones.

Este apartado es de vital importancia debido a que gracias a la identificación de las asociaciones y las agregaciones nos van a suponer un ahorro de tiempo y de código.

ACTIVIDAD ASI 6: ELABORACIÓN DEL MODELO DE DATOS.

Como se comentó anteriormente, esta actividad no se desarrolla al estar pensada para la programación estructurada.

ACTIVIDAD ASI 7: ELABORACIÓN DEL MODELO DE DATOS.

Como se comentó anteriormente, esta actividad no se desarrolla al estar pensada para la programación estructurada.

ACTIVIDAD ASI 8: DEFINICIÓN DE INTERFACES DE USUARIO.

En esta actividad definimos el comportamiento de la interfaz entre el sistema y el usuario como por ejemplo los formatos de pantalla, diálogo e informes. El objetivo es establecer cuales son los procesos del sistema de información, junto con la interacción del usuario, para cubrir los requisitos establecidos.

Tarea ASI 8.1: Especificación de principios generales de la interfaz.

En esta tarea se va a definir las directrices que deben seguir todo el conjunto que forma la interfaz.

Este apartado es más útil cuando el sistema de información lo va a desarrollar un conjunto de programadores. De esta forma anulamos adjudicar la creación de diseños a los programadores. Además de evitar de como diseñar un mensaje de error también

evitamos obtener un producto heterogéneo con diversidad de ventanas, mensajes, errores etc.

➤ **Principios generales de la interfaz**

La utilización de la aplicación es lo mas agradable e intuitiva posible, debido a que el usuario final, no es experto en manejo de software.

La ventana principal estará formada como si fuera un árbol jerárquico. Al principio tendremos las acciones más importantes y cuanto más nos adentremos en cada una de las acciones podremos ir detallando la información requerida.

Desde una misma ventana podremos hacer búsquedas de las acciones que estemos haciendo, y en la misma ventana tendremos las 4 acciones de manipulación de datos (crear, editar, ver y eliminar).

Todas las ventanas tendrán una estructura lo más parecida posible diferenciando las ventanas en colores según sean sus acciones. Diferenciaremos por colores las ventanas de pacientes, cajas, proveedores, tratamientos y ventanas de manipulación de datos.

Todas las acciones que se realicen con algún riesgo, se tendrá que confirmar, para asegurarnos de que no ha sido una equivocación que hemos llegado hasta aquí.

Los formatos de impresión serán las propias ventanas de forma organizada para ubicarlas en un formato A4 con fondo blanco. Las ventanas que posean tablas la impresión se dividirá en dos partes, la primera será una impresión con todos los datos y la segunda será una impresión exclusiva de la tabla.

Todas las ventanas dispondrán de ayuda para saber cual es el funcionamiento de la ventana. Esta ayuda se podrá localizar por búsqueda en el glosario.

Tarea ASI 8.2: Identificación de perfiles y diálogos.

El objetivo de esta tarea es identificar los perfiles de usuario, de acuerdo a su nivel de responsabilidad y al alcance de las funciones que realizan.

➤ **Especificación de Interfaz de usuario**

Catalogo de perfiles de usuario: Solo existe un único usuario que acceda a toda la funcionalidad de la aplicación

Tarea ASI 8.3: Especificación de formatos individuales de la interfaz de pantalla.

En esta tarea se explicará el funcionamiento que se espera de cada ventana en un estado estático.

Disponible en la interfaz de usuario.

Tarea ASI 8.4: Especificación del comportamiento dinámico de la interfaz.

En esta tarea describiremos cual es el comportamiento entre las ventanas y cuales son los flujos de información entre ellas. Al igual que para los diagramas de secuencia, en este apartado omitiremos los diagramas que sean comunes por disponer de un comportamiento claramente parecido.

Ver diagrama de transición de estados en el capítulo de diagramas.

Tarea ASI 8.5: Especificación de formatos de impresión.

Dicha tarea describe como van a ser las características de los formatos de impresión de los informes.

Disponible en la interfaz de usuario.

ACTIVIDAD 9. ANÁLISIS DE CONSISTENCIA Y ESPECIFICACIÓN DE REQUISITOS.

La misión de esta actividad es lograr que la variedad de los modelos que se generen estén garantizados.

La calidad vendrá dada tras aplicar en conjunto la verificación de la calidad técnica de cada modelo, aseguramiento de la coherencia entre los distintos modelos y la validación del cumplimiento de los requisitos.

Tarea ASI 9.1: Verificación de los modelos.

En esta tarea debemos verificar la calidad de los modelos según su técnica para la elaboración y el catálogo de normas.

➤ **Interfaz de usuario.**

La definición de la interfaz entre el sistema y el usuario se centra exclusivamente formada por ventanas, diálogos e informes.

Esta interacción consta de ciertos aspectos resumidos en el siguiente párrafo.

- Opciones de cambios de tamaño y ubicación.
- Existencia de dispositivos de entrada y salida necesarios para la ejecución total.
- Representación de datos mediante tablas.

➤ **Modelo y especificación de casos de uso.**

Para la creación de los casos de uso se ha tenido en cuenta todos los apartados requeridos por Métrica y la información ofrecida por el cliente. Uniendo los dos aspectos se ha concluido con un sistema completo por parte del cliente y coherente por las directrices de métrica.

Además para cada caso de uso se han desarrollado los escenarios para completar el comportamiento que posee.

➤ **Modelo y comportamiento de clases de análisis**

Para la creación de este modelo hemos tenido que tener en cuenta que se ve afectado según el estudio de los casos de uso y de la interfaz de usuario. En los cuales hemos detallado anteriormente la garantía de su calidad.

Resumidamente podemos garantizar que en el sistema de información todos los modelos cumplen la garantía que se exige debido a que se ha trabajado de forma paralela la realización de los modelos con las especificaciones del cliente. Uniendo a este conjunto varios prototipos de la interfaz de usuario, ha certificado la completitud, coherencia, consistencia y correcto de los distintos diagramas.

Tarea ASI 9.2: Análisis de consistencia entre modelos.

El objetivo de esta tarea es asegurarnos que en la relación existente entre modelos no existe duplicación de información, son coherentes entre sí y no existen ambigüedades.

➤ **Modelo de Clases/Diagramas dinámicos:**

La consistencia de clases se realiza mediante comprobaciones de mensajes. Ante todo se realiza un informe de todos los objetos que existen de todas las clases y se detallan los mensajes que se realizan unos a otros.

Cuando se requiere que se soliciten ciertos datos, nos aseguramos que los datos están disponibles para el envío y la recepción.

Para terminar se disponen de los diagramas de interacción, donde se detalla evento a evento la comunicación entre clases

➤ **Modelo de clases/Interfaz de usuario:**

La relación de clases y la interacción con el usuario para la toma de datos se realiza mediante la interfaz de usuario. Debido a que las clases son correctas y completas, la interfaz de usuario debe cumplir estas características para continuar con la garantía que poseemos respecto al diagrama de clases.

➤ **Análisis de la realización de los casos de uso/interfaz de usuario**

Todos los elementos que obligan a la navegación por pantallas, tendrán que poseer una asociación con un mensaje en el diagrama de interacción de objetos indicando el motivo.

➤ **Análisis de los diagramas de estados/diagramas de interacción**

La elaboración de los diagramas de estado (que resulta de ser un diagrama por ventana de la interfaz de usuario) se empieza a través de los diagramas de interacción, pero en estos diagramas se tienen que incluir todos los casos, acciones y opciones que puedan dar a lugar. Gracias a dichos diagramas nos han sido muy útiles para las especificaciones de la actividad 8.

Al realizar el desarrollo y análisis en conjunto con estos tres elementos podemos garantizar que la consistencia es correcta para todos los modelos.

Tarea ASI 9.3: Validación del modelo.

En esta tarea tendremos que validar todos los modelos realizados ayudados por el catálogo de requisitos, salvo para la interfaz de usuario en el que es más útil diseñar un prototipo como método de validación.

Si por algún motivo existe un modelo que no es válido volveremos hacia atrás para realizar los cambios necesarios hasta que todo sea correcto. Por el contrario, si todos los elementos son validos podemos continuar. Para poder continuar debemos de verificar que el conjunto de todos los diagramas y la interfaz no albergan ningún tipo de fallo. Sería muy útil (y de gran funcionalidad en cualquier sistema de gestión) que todo el sistema estuviera preparado para ajustar cualquier tipo de cambio con la misión de minimizar costes de mantenimiento.

Tarea ASI 9.4: Elaboración de la especificación de requisitos software (ERS).

Esta tarea consta de que vamos a dar por finalizado el enfoque del análisis y vamos a dar el salto al desarrollo del software, para ello todo debe de estar bien definido y no debe de quedar ninguna duda de cual va a ser la funcionalidad y objetivo de cada clase, ventana o diálogo.

Ahora realizamos una revisión de todo lo especificado para verificar que el análisis ha concluido con un gran éxito.

1. Introducción: Igual que en la tarea EVS 1.1 (Descripción general del sistema)
2. Ámbito y alcance: Igual que en la tarea EVS 1.2 (Contexto del sistema; Estructura organizativa)
3. Participantes: Igual que en la tarea ASI 1.4 (Catálogo de usuarios)
4. Requisitos del sistema de información: Igual que en la tarea ASI 1.1 (Catálogo de requisitos)
5. Visión general del sistema de información: Igual que en la tarea ASI 8.2 (Catálogo de perfiles de usuario)

6. Referencia de los productos a entregar:

El producto serán los modelos que han sido comentados a lo largo del presente análisis:

- Diagramas de casos de uso.
- Diagramas de clases: dominio del problema.
- Diagramas de clases: gestión de datos
- Diagramas de secuencia o interacción.
- Diagramas de estados.
- Estudio de viabilidad del sistema (EVS)
- Análisis de sistema de información (ASI)

7. Plan de acción

Igual que en la tarea EVS 1.3 (Plan de trabajo)

ACTIVIDAD 10.ESPECIFICACIÓN DEL PLAN DE PRUEBAS.

Aquí definiremos el plan de pruebas, que nos servirá como guía. Además nos ayudará a verificar que el sistema de información cumple las necesidades establecidas por nuestro cliente.

Tarea ASI 10.1: Definición del alcance de pruebas.

➤ **Plan de pruebas**

Como en la solución del sistema de información no hemos adquirido ningún tipo de software que lo complemente, nos vemos con la obligación de realizar todas las pruebas de todas las clases, objetos y pantallas.

Tarea ASI 10.2: Definición de requisitos del entorno de pruebas.

➤ Plan de pruebas

El único requisito que se requiere es que la plataforma a realizar las pruebas debe ser Microsoft Windows Xp en adelante (inclusive). Si por ejemplo, el sistema de información debiera de funcionar en cualquier otra plataforma, nos veríamos a realizar las pruebas en distintas plataformas.

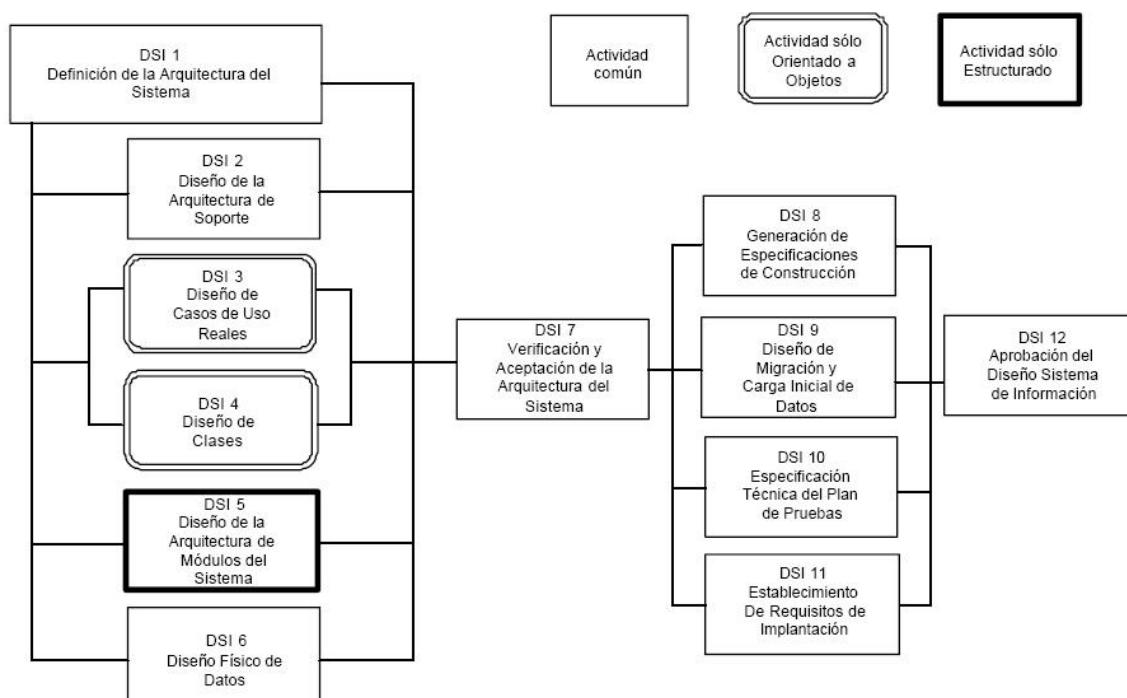
Nota: todos estos pasos aunque parezcan que son una pérdida de tiempo, tienen un efecto totalmente contrario. Ahora tenemos claro cual va a ser el producto final, que es lo que queremos detallar y tenemos detallados todos los posibles peligros que podemos tener. Gracias a todos estos diagramas y estudios tenemos claro las fases del desarrollo y se espera que se realice en un tiempo muy inferior que si nos hubiéramos saltado estos puntos.

2.3 DISEÑO DEL SISTEMA DE INFORMACIÓN

Terminado el análisis, comenzamos con el diseño del sistema. En este proceso abordaremos las actividades necesarias para definir como va a ser la estructura del sistema, su interfaz y el entorno tecnológico del que se dispone. Además se detallarán las especificaciones para el plan de prueba, la migración de datos iniciales y los requisitos de implantación.

Las actividades de este proceso, se pueden dividir en dos grandes bloques. El primero de ellos es el que da cobertura al diseño del sistema. Todas estas actividades tienen las características de que se puede realizar en paralelo y se retroalimentan en mayor o menor magnitud dependiendo del sistema. El segundo es el que ofrece las especificaciones necesarias para la construcción del sistema.

Las actividades a realizar son las siguientes:



Actividad DSI 1: Definición de la arquitectura del sistema.

En esta actividad definiremos la arquitectura general del sistema especificando los subsistemas lógicos posibles.

Tarea DSI 1.1: Definición de niveles de arquitectura.

➤ Diseño de la Arquitectura del Sistema

La arquitectura del sistema está comentada en el ASI 1.2.

Tarea DSI 1.2: Identificación de requisitos de diseño y construcción.

En esta tarea vamos a definir conceptos que están muy ligados con el diseño de la arquitectura, pudiendo interferir en el diseño y construcción del sistema.

- Mediante un ordenador adquirido en un periodo de hace dos años, valdrá para el correcto funcionamiento del software.
- Sobre la base de datos, no pondremos límite porque no se va utilizar todo el potencial de MySql y en el concepto de la capacidad no habrá problema por ser un sistema local.
- Como último rasgo a tener en cuenta será disponer de cualquier tipo de impresora.

Tarea DSI 1.3: Especificación de excepciones.

En esta tarea se busca definir patrones para responder a los comportamientos no habituales del sistema.

- Si el problema proviene de la base de datos, el sistema de información estará incapacitado para realizar operaciones de consulta o modificación.

- Si el problema proviene de la impresora, estará inhabilitado cualquier tipo de impresión de documentos.
- El sistema no ofrecerá al usuario la introducción de datos erróneos. Los campos a introducir estarán limitados los caracteres no válidos.
- En determinados formularios existirán campos que sean obligatoriamente rellenados. Estos errores serán informados al usuario de que falta información para continuar con los procesos.
- Al existir tablas relacionadas entre sí, no se podrá eliminar información si está relacionada con alguna tabla. Se informará al usuario el motivo de la eliminación.

Algunas de estas excepciones se han dado a conocer anteriormente en los escenarios de los casos de uso y aparecen en los diagramas de secuencia.

➤ **Catalogo de requisitos**

Se mantiene como el punto anterior.

Tarea DSI 1.4: Especificación de estándares y normas de diseño y construcción.

En esta tarea se va a definir los patrones a mantener en el diseño del sistema. Esta tarea cobra más sentido cuando existe un grupo de trabajo numeroso que indirectamente incide a realizar diferentes tipos de diseño.

➤ **Catalogo de normas**

El catálogo de normas se mantiene como el ASI 1.3 pero se añade los estándares y las normativas de elaboración de software

Tarea DSI 1.5: Identificación de subsistemas de diseño.

El objetivo de esta tarea es dividir de forma lógica el sistema en subsistemas de diseño, para aprovechar todas las características de la programación en objetos.

La partición necesaria coincide con la subdivisión que se hizo en el ASI, y en los diagramas de transición de estados se puede apreciar los formularios que están relacionados con cada caso de uso.

Tarea DSI 1.6: Especificación del entorno tecnológico.

Aquí definimos con detalle los diferentes dispositivos de la infraestructura técnica.

Como no existe ningún cambio el sistema se queda como en el ASI 1.1

Tarea DSI 1.7: Especificación de requisitos de operaciones y seguridad.

➤ Procedimientos de seguridad

Para asegurar todos los riesgos que conlleva el almacenamiento de la información protegida, solo se podrá tener acceso al sistema mediante un registro en el sistema.

➤ Copias de seguridad

El usuario podrá realizar tanto copia de seguridad de la información como restaurar la última copia hecha.

➤ **Seguridad en los datos**

Como se ha comentado anteriormente el sistema proporcionará un entorno estable, en el que esté anulada la introducción de datos no válidos.

Actividad DSI 2: Diseño de la arquitectura de soporte.

Esta actividad es realizada cuando comprende con un diseño de soporte. Por lo tanto esta actividad queda fuera del alcance de nuestro sistema de información y no es necesaria su elaboración.

Actividad DSI 3: Diseño de casos de uso reales.

Esta actividad, que es exclusiva para cuando el sistema se va a realizar en orientación a objetos, define el comportamiento del sistema para un caso de uso y se determina las operaciones de las clases e interfaces para los diferentes subsistemas.

Tarea DSI 3.1: Identificación de clases asociadas a un caso de uso.

Como no ha sufrido ninguna variación, la relación se mantiene con en el ASI 4.2

Tarea DSI 3.2: diseño de la realización de los casos de uso.

En esta tarea se describe como están relacionadas las clases que se puede comprobar en:

Ver diagrama de interacción en el capítulo de diagramas.

Tarea DSI 3.3: Revisión de la interfaz de usuario.

Todos los elementos referentes a la interfaz gráfica como formatos de impresión y ventanas fueron detallados en el ASI y no han sido modificados.

Actividad DSI 4: Diseño de clases.

Esta actividad, que al igual que la anterior solo se realiza para la orientación en objetos, pretende revisar y completar las clases además de sus relaciones, atributos y operaciones. Para ello se transforma el modelo lógico en el modelo de clases de diseño.

Tarea DSI 4.1: Identificación de clases adicionales.

Tras realizar una revisión de las clases no se han encontrado.

Tarea DSI 4.2: Diseño de asociaciones y agregaciones.

Las diferentes asociaciones y agregaciones que existen se pueden comprobar en el diagrama de clases que se expuso en el ASI.

Tarea DSI 4.3: Identificación de atributos de las clases.

Como las clases no se han modificado, se mantiene con los mismos atributos que en el punto anterior.

Tarea DSI 4.4: Identificación de operaciones de las clases.

En esta tarea se identifican las operaciones de cada clase que se pueden apreciar en el diagrama de clases.

Tarea DSI 4.5: Diseño de la jerarquía.

Esta tarea la podemos excluir porque nuestro diagrama de clases no posee ninguna relación de herencia

Tarea DSI 4.6: Descripción de métodos de las operaciones.

En esta tarea se realiza la descripción mediante un lenguaje natural la definición de cada método. Como por nomenclatura de la programación los métodos reciben nombres de acuerdo con su función por lo que todos los nombres de los métodos deducen cual es su funcionalidad. Por lo tanto el diagrama de clases se queda tal como lo teníamos en el apartado 4.4

Actividad DSI 6: Diseño físico de datos.

En esta actividad se destina a la creación de la estructura física de los datos (base de datos) partiendo desde el diagrama de clases. Esta transformación de clases a modelado de datos, en la programación orientada a objetos sigue unas pautas muy metódicas y sin nivel de dificultad.

Tarea DSI 6.1: Diseño del modelo físico de datos.

En el diagrama de gestión de datos podemos apreciar el resultado final de la base de datos. En un principio se muestra el nombre de las tablas y sus relaciones. A continuación se muestra el contenido de cada tabla.

[Ver diagrama de la base de datos en el capítulo de diagramas.](#)

Actividad DSi 7: Diseño físico de datos.

Llegados a este punto debemos comprobar la calidad de las especificaciones del diseño del sistema, y como hicimos en el ASI, se tienen que garantizar la coherencia entre los diferentes modelos y sus especificaciones.

Tarea DSi 7.1: Diseño del modelo físico de datos.

Revisando todos los diagramas y modelos elaborados podemos continuar con los restantes puntos al mantenerse en coherencia los distintos modelos.

Tarea DSi 7.2: Análisis de consistencia de las especificaciones de diseño.

Además de lo comentado en el punto anterior, podemos asegurar que no los diagramas y modelos elaborados, no poseen ni ambigüedades ni duplicación de información. Estos diagramas y modelos son válidos para la realización del sistema de información.

Tarea DSi 7.3: Análisis de consistencia de las especificaciones.

He realizado una comprobación completa del sistema, intentando buscar que no haya incoherencias ni ambigüedades. Buscando que todos los elementos del sistema se integren correctamente y exista una correlación directa con el interfaz de usuario. De forma que quede implementados de forma clara y concisa todos los diagramas y modelos que he realizado en el proyecto.

Actividad DSI 8: Generación de especificaciones de construcción.

En esta actividad se definen las características para la construcción del sistema de información

Tarea DSI 8.1: Especificación del entorno de construcción.

A continuación vamos a describir el conjunto de equipo con el que se trabaja.

Para un entorno tecnológico:

- Un PC portátil centrino 1800 MHz, dispone de 1Gb de memoria Ram y bajo plataforma de Windows Xp.
- Impresora HP láser para la impresión de la memoria.

Este equipo puede dar respuesta a cualquier necesidad que requiera las aplicaciones necesarias para el desarrollo del proyecto.

Para las herramientas de construcción necesarias son:

- Visual Studio 2005 con el compilador de C#
- Magic Draw para la generación de diagramas.
- MySql para la gestión de la base de datos.
- Aqua Data Studio como entorno de trabajo para la base de datos.
- Microsoft Project para la planificación
- HelpCreator para la elaboración del glosario de ayuda.
- Adobe Acrobat y Microsoft Word para editar y consultar información de texto.

➤ Calendario de planificación estimado

Se mantiene como el de la tarea EVS 1.3 (plan de trabajo)

Tarea DSI 8.2: Definición de componentes y subsistemas de construcción.

La aplicación es orientada desde el enfoque de los casos de uso, manteniendo los mismos subsistemas.

Es importante destacar la importancia de incluir una componente para poder comunicar la base de datos de MySql con el compilador C#. Desde esta clase podremos solicitar información de la base de datos y realizar las operaciones de inserción, modificación y eliminación.

Tarea DSI 8.3: Elaboración de especificaciones de construcción.

En el manual de usuario está disponible el comportamiento de cada formulario. Además quedan disponibles los diagramas de transición de estado para consultar los diferentes eventos que ocurren entre formularios.

Tarea DSI 8.4: Elaboración de especificaciones del modelo físico de datos.

La base de datos se aloja en un gestor de MySql. Para no utilizar la consola que trae MySql he utilizado Aqua Data Studio que es más visual. En el diagrama de gestión de base de datos se puede observar como están compuestas y relacionadas la base de datos.

Ver diagrama de base de datos en el capítulo de diagramas.

Actividad DSI 9: Diseño de la migración y carga inicial de datos.

Como se argumentó en el apartado 4.7 de esta sección, no se realiza la migración de base de datos ajena a nuestro sistema. Recordamos que en el EVS actividad 2 hablábamos de que no poseía ningún software de almacenamiento de datos.

Actividad DSI 10: Especificación técnica del plan de pruebas.

En esta actividad se elabora con detalle el plan de pruebas al que se va a someter el sistema de información.

Tarea DSI 10.1: Especificación del entorno de pruebas.

Por los recursos disponibles, el entorno de pruebas coincide con los entornos de construcción.

He realizado las distintas pruebas unitarias, de integración, de implantación y aceptación.

Se han introducidos todas las posibles posibilidades de datos en los diferentes formularios para verificar el estado del sistema de información.

Tarea DSI 10.2: Especificación técnica de niveles de prueba.

En esta tarea se definen los diferentes niveles de pruebas existentes en el sistema

- Prueba unitaria:

Estas pruebas deben tener en cuenta:

- Los campos obligatorios, no dejando pasar el control del formulario a otro evento sin dejar ningún campo obligatorio vacío.
- Excepciones: En estas pruebas se busca asegurar la integridad de cada unidad de la aplicación. Se debe controlar todas las posibles excepciones que puedan ocurrir dentro de un formulario. Estas excepciones vienen provocadas por errores en la conversión de datos, problemas de la base de datos etc.
- Funcionamiento: como último punto debe asegurar que el elemento implementado realiza la misión que tenía encomendada.

- Pruebas de integración:

Estas pruebas consisten en integrar los diferentes elementos, partiendo de los niveles más internos hacia fuera. Una vez integrado todo en el conjunto se realizan las pruebas unitarias necesarias. Al final de la integración tendremos una aplicación que funciona correctamente.

- Pruebas de implantación:

En estas pruebas no provocarán problemas debido a que el entorno tecnológico al que va destinado el sistema es muy parecido al entorno de construcción.

Tarea DSI 10.3: Revisión de planificación de pruebas.

Desde el plan de pruebas del punto anterior se realiza una planificación estimada para las pruebas. Como solo tenemos un perfil bastará con realizar las pruebas pertinentes solo con este. En el caso de que tuviéramos más de un usuario diferente, tendríamos que realizar todas las pruebas necesarias en todos los diferentes usuarios.

Para la planificación se detallada además la preparación y ejecución de las diferentes pruebas, la evaluación de los resultados y el tiempo estimado.

El tiempo que se ha dedicado se puede observar en la planificación del sistema.

Actividad DSI 11: Establecimiento de requisitos de implantación.

Ahora se tiene que definir unos requisitos para que el usuario pueda manipular el sistema con el menor riesgo y problemas posibles.

Tarea DSI 11.1: Especificación de requisitos de documentación de usuario.

En esta tarea describiremos las condiciones que debe tener la documentación que recibirá el usuario junto al sistema de información.

La documentación que entregáramos al usuario será el manual de usuario.

El manual será entregado en soporte electrónico e impreso a pesar de las condiciones del volumen.

Una segunda documentación que recibirá será el tipo de ayuda con el que está provista la aplicación, que siempre podrá acceder a ella más rápidamente al estar desglosada por términos.

Tarea DSI 11.2: Especificación de requisitos de documentación de implantación.

En esta tarea haremos una descripción de los requisitos que requiere la implantación del sistema.

Una vez implantado el sistema, se procederá con una pequeña demostración de cómo funciona el sistema. Debido a que la interfaz es intuitiva y sencilla se realizará las pruebas necesarias hasta que el usuario final logre ver el fin de la aplicación.

Actividad DSI 12: Aprobación del diseño del sistema de información.

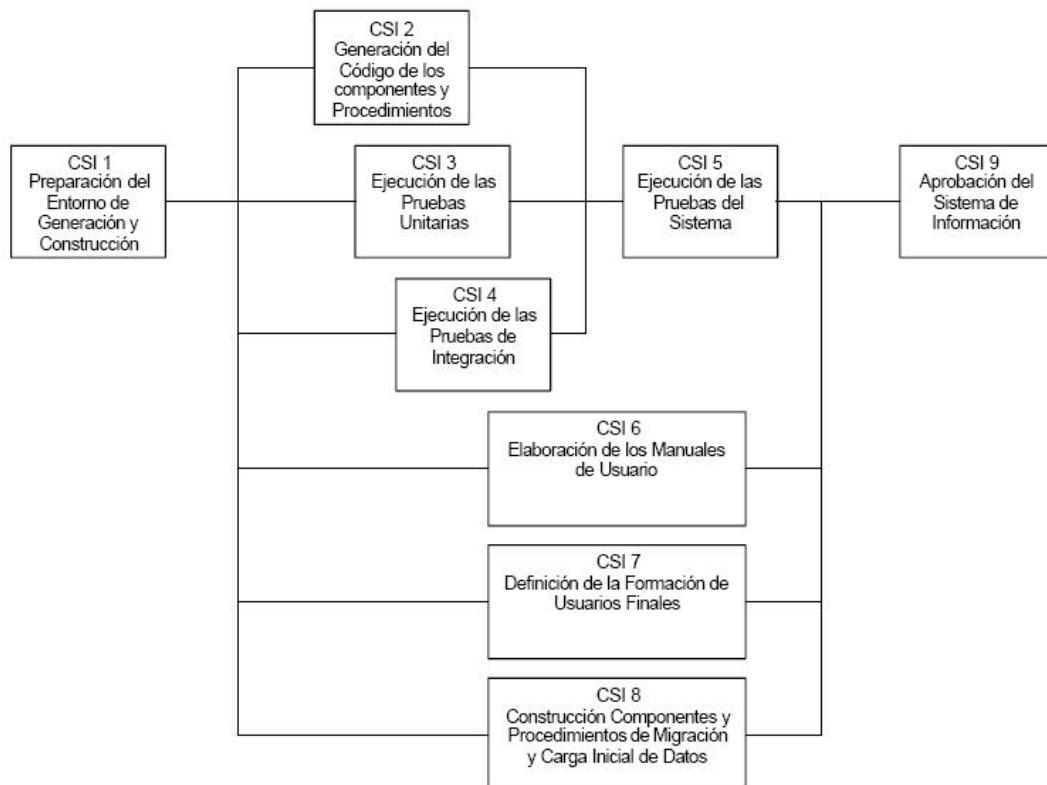
Tarea DSI 12.1: Presentación y aprobación del diseño del sistema de información.

Esta tarea está destinada a realizar la presentación que hemos cubierto en el apartado anterior, pero ahora delante del comité de directivos que nos encargaron el producto. De este modo obtendremos las gratificaciones del resultado final.

2.5 CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

Llegados a este proceso, nos disponemos a concluir nuestro ciclo de metodología de métrica. En este proceso hay que elaborar todos los componentes y procedimientos necesarios para el funcionamiento del sistema y el manual de usuario correspondiente.

Las actividades a realizar son las siguientes:



Actividad CSI 1: Preparación del entorno de generación y construcción.

En esta actividad hay que confirmar disponemos de todos los medios y facilidades para llevar a cabo la construcción del sistema. Comprobar el entorno de construcción y sus requisitos.

Tarea CSI 1.1: Implantación de la base de datos física o ficheros.

En esta tarea se tiene que crear la base de datos, reservar espacio de almacenamiento e iniciar la base de datos.

Esto lo hemos realizado con la aplicación Aqua Data Studio que nos permite tener una visión más cómoda sobre la base de datos.

La arquitectura del sistema está comentada en el ASI 1.2.

Tarea CSI 1.2: Preparación del entorno de construcción.

En esta tarea preparamos el entorno necesario para construir las componentes.

Ya que tenemos implantados la base de datos, pasamos a implantar el Visual Studio y enlazamos con la librería necesaria para conectar la base de datos con C#.

Actividad CSI 2: Generación del código de los componentes y procedimientos.

En esta actividad elaboramos la codificación de los componentes según están establecidas las especificaciones de construcción en el DSI.

Tarea CSI 2.1: Generación del código de componentes.

En esta tarea se genera el código de cada componente, respetando la nomenclatura y calidad recogido en el catálogo de normas.

Tarea CSI 2.2: Generación del código de los procedimientos de operación y seguridad.

La diferencia de esta tarea respecto al punto anterior, es que ahora se desea elaborar los procedimientos de seguridad para acceder al sistema una vez esté realizado. En nuestro sistema esto coincide con el formulario de Registro.

Actividad CSI 3: Ejecución de las pruebas unitarias.

En esta actividad se llevaran a cabo las pruebas unitarias de cada componente generada. Esto lo realizaremos para verificar que se ajusta a la funcionalidad por la que se creo.

Tarea CSI 3.1: Preparación del entorno de las pruebas unitarias.

En esta tarea hay que preparar el entorno de pruebas. Para ello recopilaremos los recursos y datos necesarios para poder realizar las pruebas unitarias.

Tarea CSI 3.2: Realización y evaluación de las pruebas unitarias.

El resultado final del punto anterior no ha concluido con las satisfacciones esperadas, por lo que hemos tenido que retocar parte de la aplicación para ajustarla según el DSI.

Al haber modificadas dichas acciones y verificar que no existen errores, podemos concluir con que la aplicación es completa y correcta.

Actividad CSI 4: Ejecución de las pruebas de integración.

En esta actividad nos daremos cuenta al realizar las pruebas de integración, si todos los componentes entre sí se relacionan como esperábamos y verificamos que la transmisión de los mensajes entre las interfaces es el correcto.

Tarea CSI 4.1: Preparación del entorno de las pruebas de integración.

Debido al sistema de información tan peculiar que poseemos, el entorno necesario para las pruebas de integración va a ser el mismo que para las pruebas unitarias.

Tarea CSI 4.2: Realización de las pruebas de integración.

En esta tarea tenemos que realizar todas las pruebas de integración que se comentó en el DSI con el entorno establecido en el punto anterior.

Tarea CSI 4.3: Evaluación del resultado de las pruebas de integración.

El objetivo de estas pruebas es recoger el grado de satisfacción del resultado de las pruebas frente al esperado.

En nuestro caso el programa ha respondido con bastante bien. Debido a esta respuesta, no tenemos que generar nuevas pruebas y volver a realizar el plan de pruebas, ni parcial ni totalmente.

Actividad CSI 5: Ejecución de las pruebas del sistema.

El objetivo de esta actividad es comprobar la integración del sistema, verificar el funcionamiento correcto de las interfaces de los diferentes subsistemas y con el resto del sistema.

En estas pruebas hay que tener en cuenta los requisitos, debido a que el incumpliendo de estos puede comprometer la aceptación del sistema desde el comité. Al final de la metodología, dicho comité dará la aceptación del sistema elaborado a partir de estos requisitos. Este último paso comentado se realiza en la implantación y aceptación del sistema.

Tarea CSI 5.1: Preparación del entorno de las pruebas del sistema.

Como en la tarea anterior, el entorno de pruebas del que se dispone coincide con el de las pruebas unitarias.

Tarea CSI 5.2: Realización de las pruebas del sistema.

En esta tarea debemos realizar las pruebas establecidas en el proceso anterior. Debemos comprobar la integración de todos los subsistemas, al igual que la interacción con otros sistemas si se dieran el caso.

Tarea CSI 5.3: Evaluación del resultado de las pruebas del sistema.

En esta tarea evaluaremos los resultados de las pruebas y el grado de cumplimiento que consiste en:

- Comparar los resultados esperados con los obtenidos.
- Investigar el origen del problema detectado y determinar las acciones necesarias para resolverlo.
- Indicar si el plan de pruebas a finalizado o por el contrario si se debe de realizar total o parcialmente.

En nuestro caso, el resultado ha sido satisfactorio y hemos concluido la actividad, deshaciendo la idea de rehacer el plan de pruebas.

Actividad CSI 6: Elaboración de los manuales de usuario.

Tarea CSI 6.1: Elaboración de los manuales de usuario.

A partir de los requisitos de documentación de usuario y la aplicación, debemos de crear el manual de usuario que ira destinado al usuario final.

- **Manual de usuario**

[Ir al capítulo de manual de usuario.](#)

Actividad CSI 7: definición de la formación de usuarios finales.

En esta actividad tenemos que asociar la dificultad de utilización de la aplicación con los niveles necesarios de formación del usuario final. En nuestro caso no será necesario realizar este estudio debido a que por parte de la aplicación no existe ninguna funcionalidad difícil de entender y por parte del usuario final no existe requisito de la formación.

De todos modos, el usuario tiene a disposición el manual de usuario y la ayuda de la aplicación.

Actividad CSI 8: Construcción de los componentes y procedimientos de migración y carga inicial de datos.

Esta actividad comprende la importación de datos iniciales desde otra fuente de información ajena a nuestra base de datos. Como en el DSI comentamos que esta posibilidad no se podía dar, avanzamos a la siguiente actividad.

Actividad CSI 9: Aprobación del sistema de información.

Tarea CSI 9.1: Presentación y aprobación del sistema de información.

Concluido este proceso, damos por finalizado el ciclo de construcción de la aplicación. Nos encontramos con todos los modelos desarrollados, la aplicación ha superado las pruebas propuestas y disponemos del manual de usuario. Solo falta demostrar al comité de seguimiento para la aprobación del sistema.

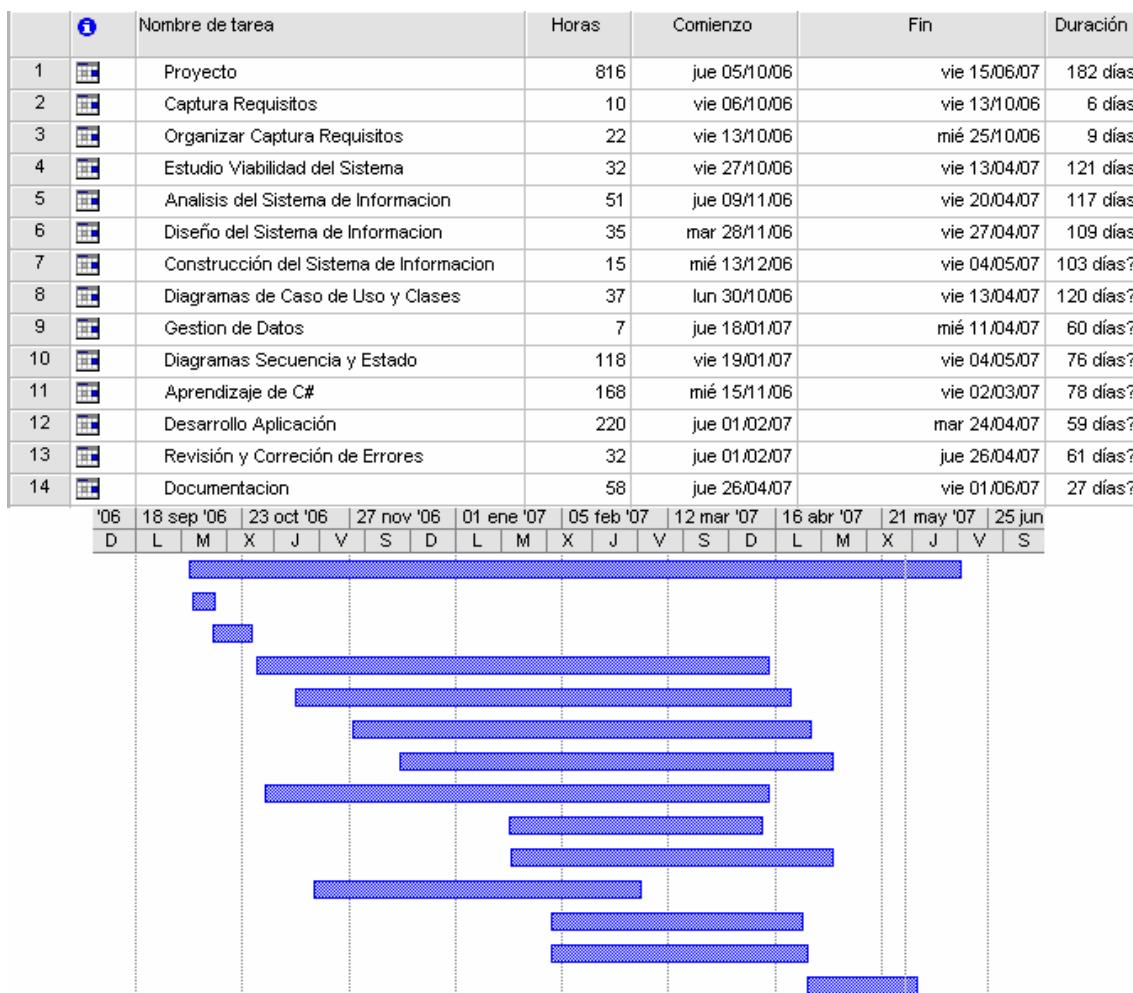
Debido a la complejidad del sistema, concluimos el desarrollo de métrica.

En comparación con un proyecto más complejo, con diversidad de personal para la función de las distintas tareas y objetivos de métrica, nos faltaría detallar el mantenimiento que requiere el software. Además como este sistema corresponde a un proyecto fin de carrera, debemos de excluir esta posibilidad al ser el mantenimiento la operación más cara y con mayor necesidad de tiempo que las operaciones de análisis, diseño y construcción.

2.6: Planificación

Planificación real:

A continuación se muestra la planificación real del proyecto. La duración en días es de 182 días, pero este dato no tiene valor significativo, debido a que a lo largo de esta duración, el número de horas es muy variado, pasando de días dedicados por completo al proyecto, otros en cambio no se ha realizado ningún avance y también existen días que se le ha dedicado algunas horas sueltas. El número de horas totales es de 816 y en la gráfica se realiza un desglose de cada tarea.



Capítulo 3: Diagramas

A continuación se muestran los diagramas realizados por la metodología Métrica. Los diagramas que van a aparecer son los de caso de uso, diagrama de clases, diagrama de base de datos, diagramas de secuencia y diagramas de interacción. Todos estos diagramas son necesarios para la implantación y los dos últimos modelos, son necesarios para la programación orientada a objetos.

3.1: Diagrama de Casos de uso del Proyecto

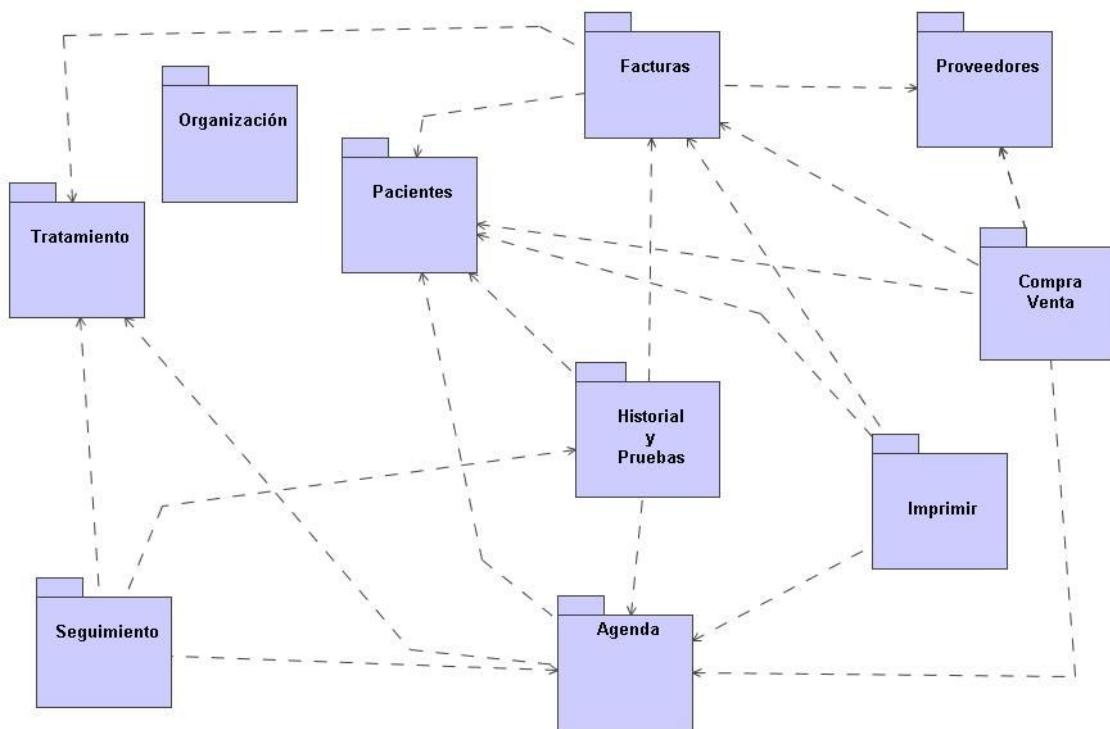


Fig1. Diagrama de paquetes de casos de uso

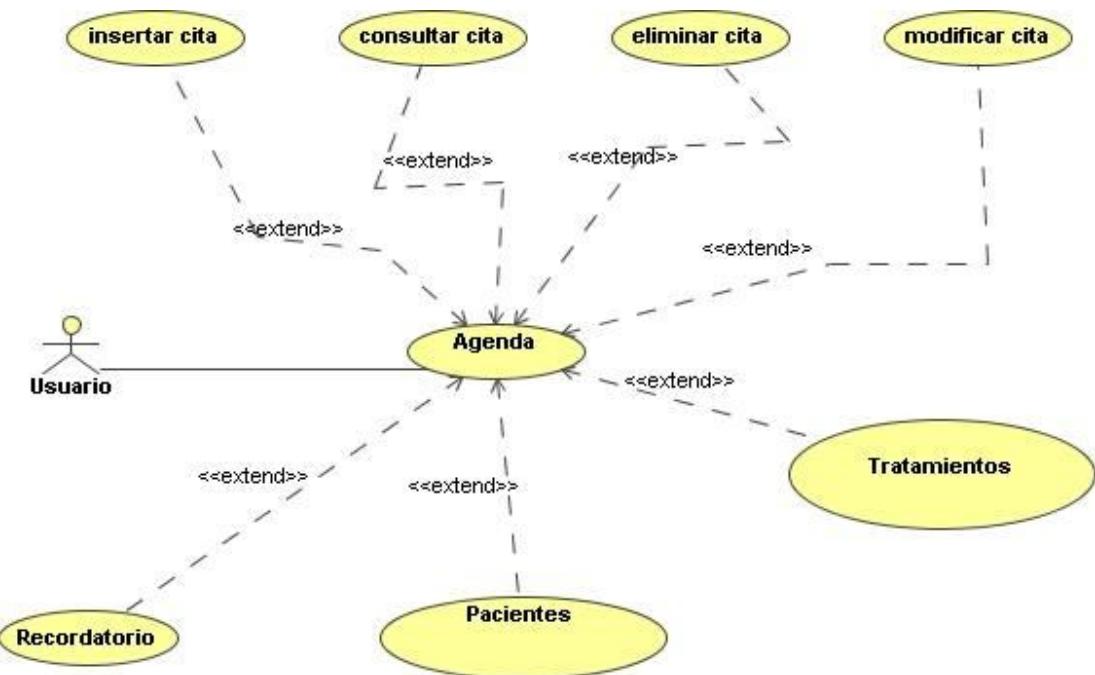


Fig2.Diagrama de caso de uso de Agenda

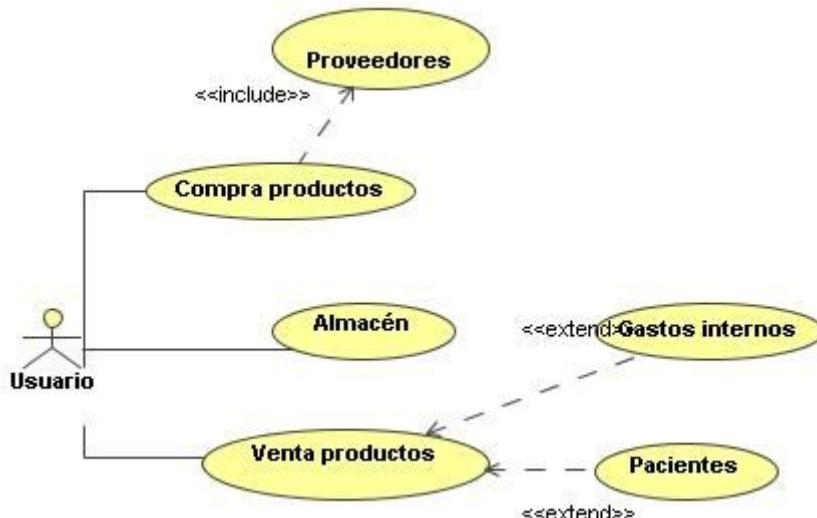


Fig3.Diagrama de caso de uso de Compra Venta

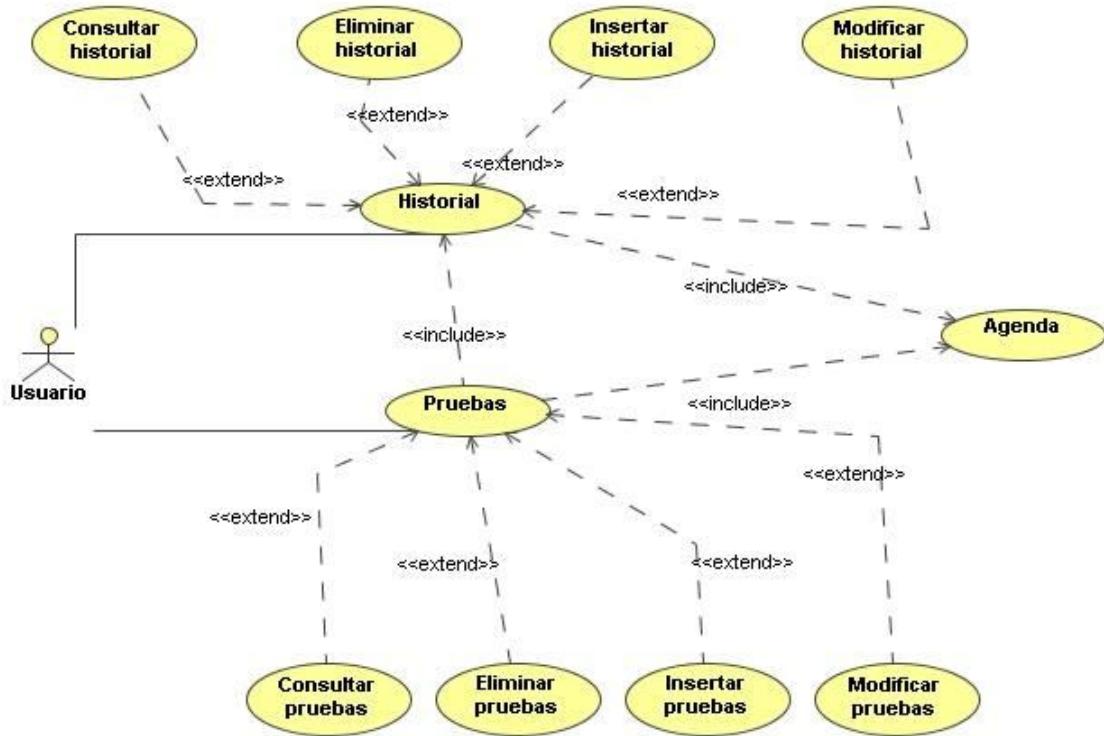


Fig4. Diagrama de caso de uso de Pruebas e Historiales

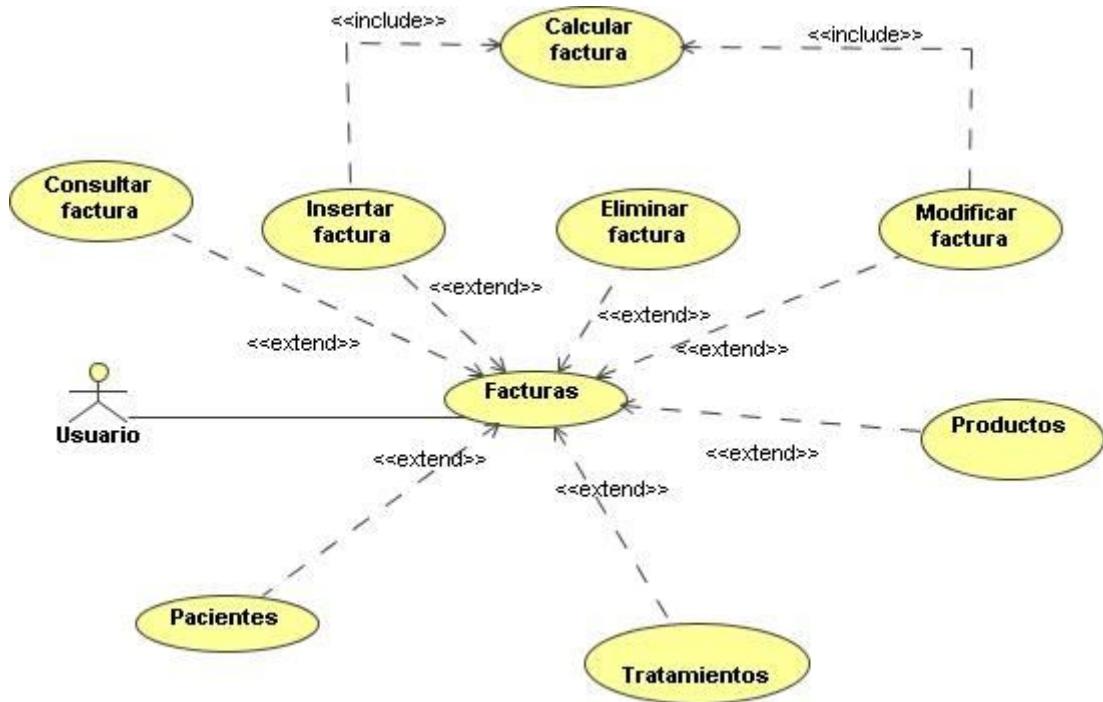


Fig5. Diagrama de caso de uso de Facturas

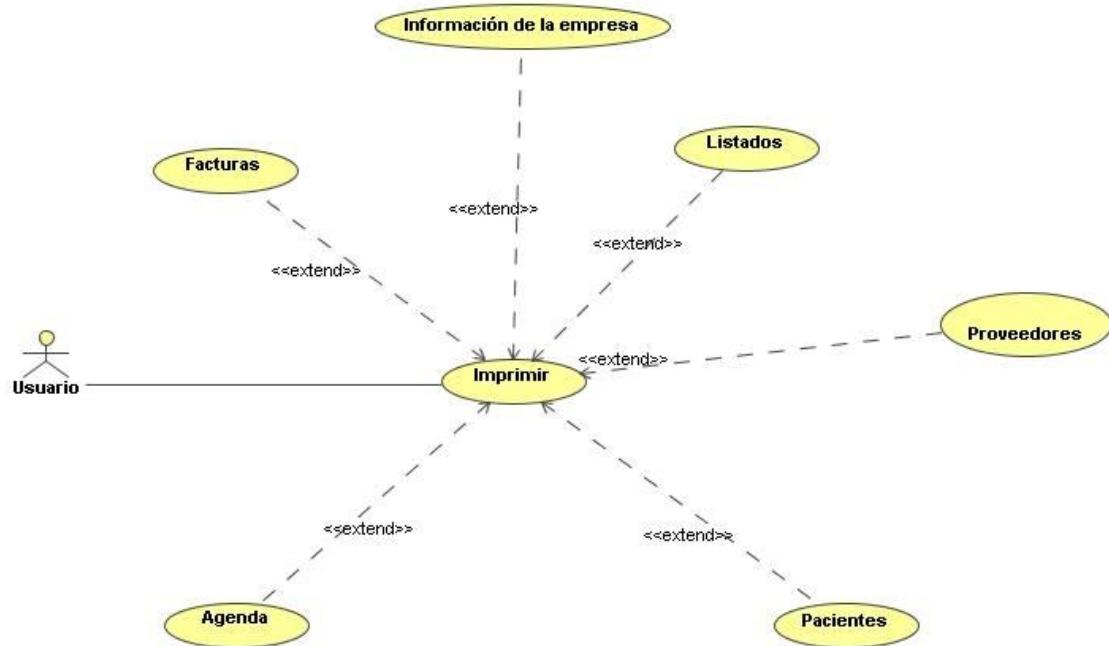


Fig6.Diagrama de caso de uso de Imprimir

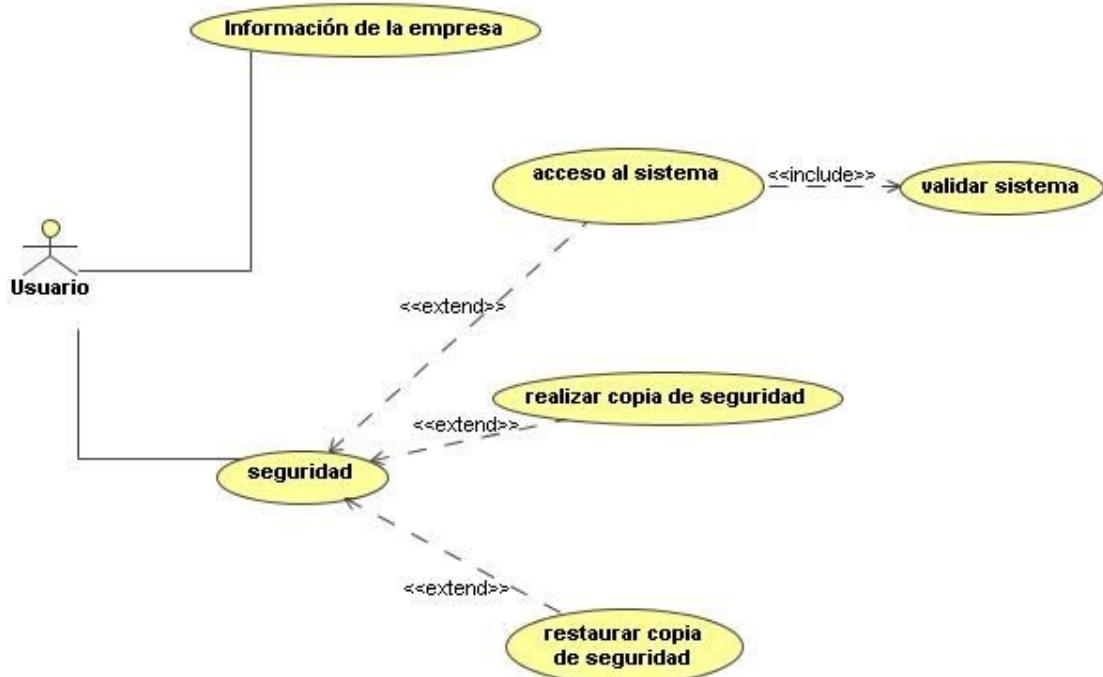


Fig7.Diagrama de caso de uso de Organización

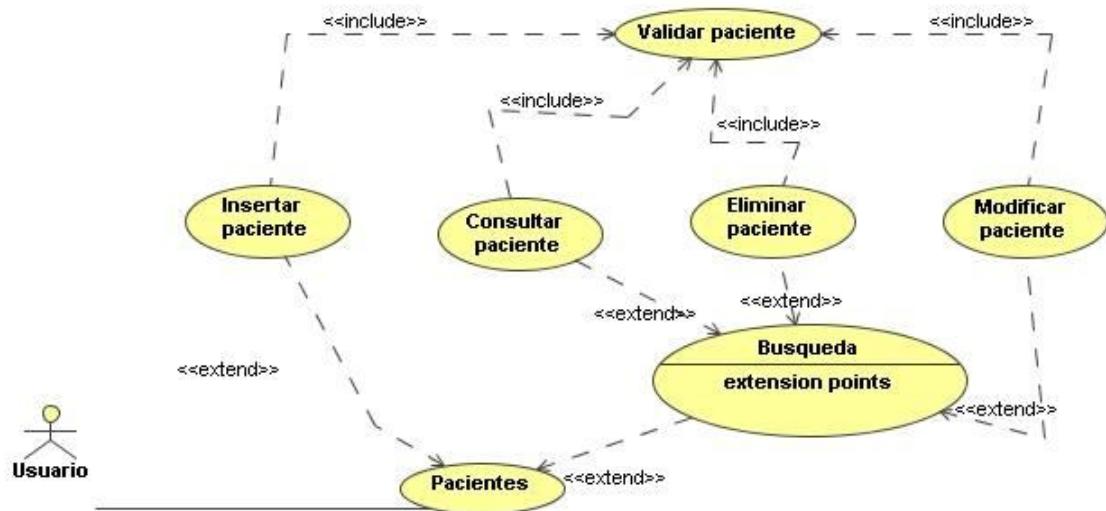


Fig8.Diagrama de caso de uso de Pacientes

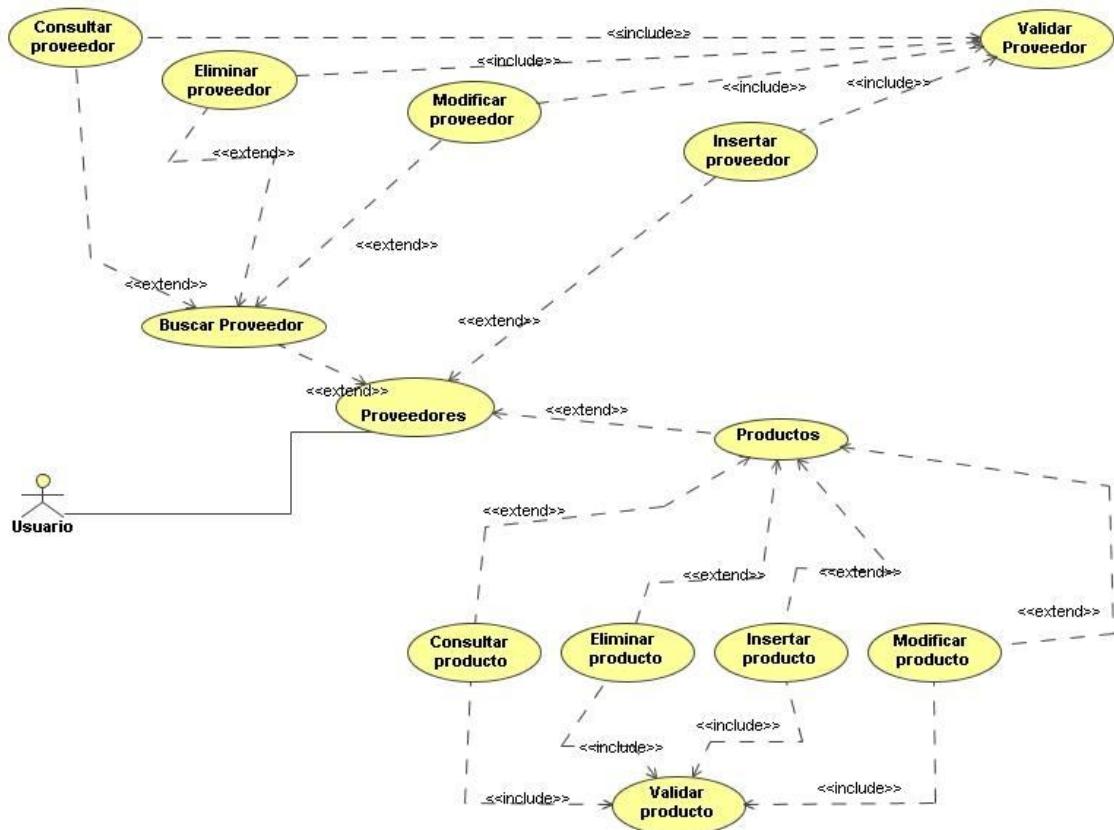


Fig9.Diagrama de caso de uso de Proveedores

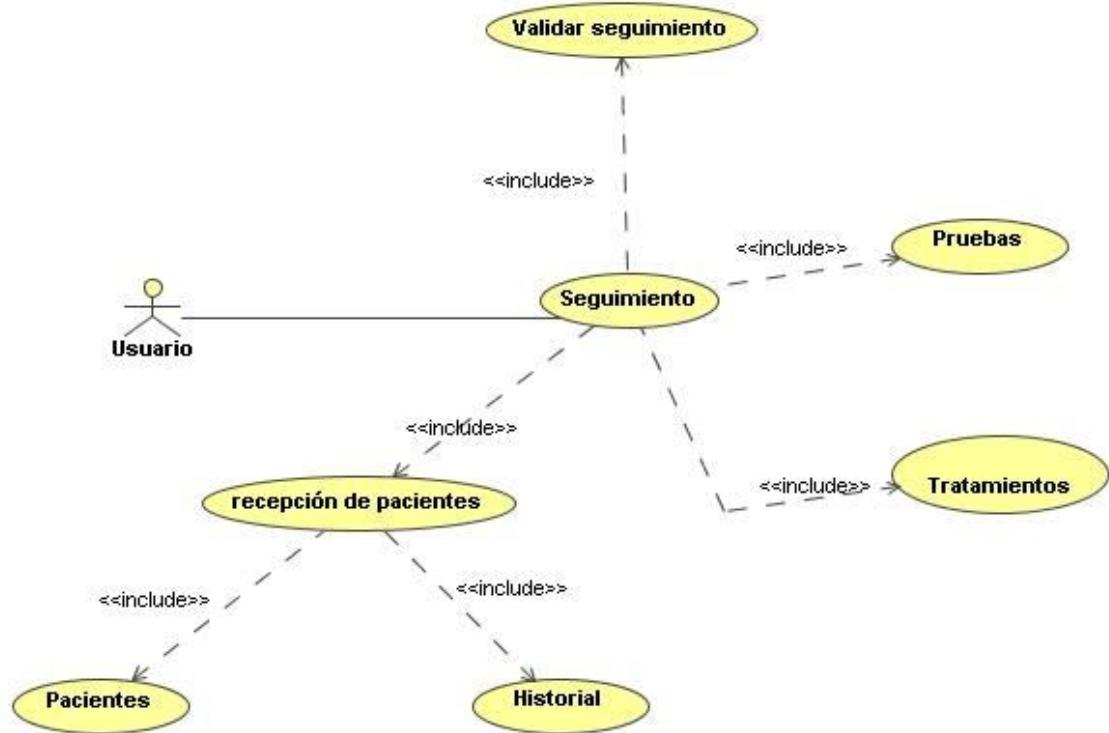


Fig10.Diagrama de caso de uso de Seguimiento

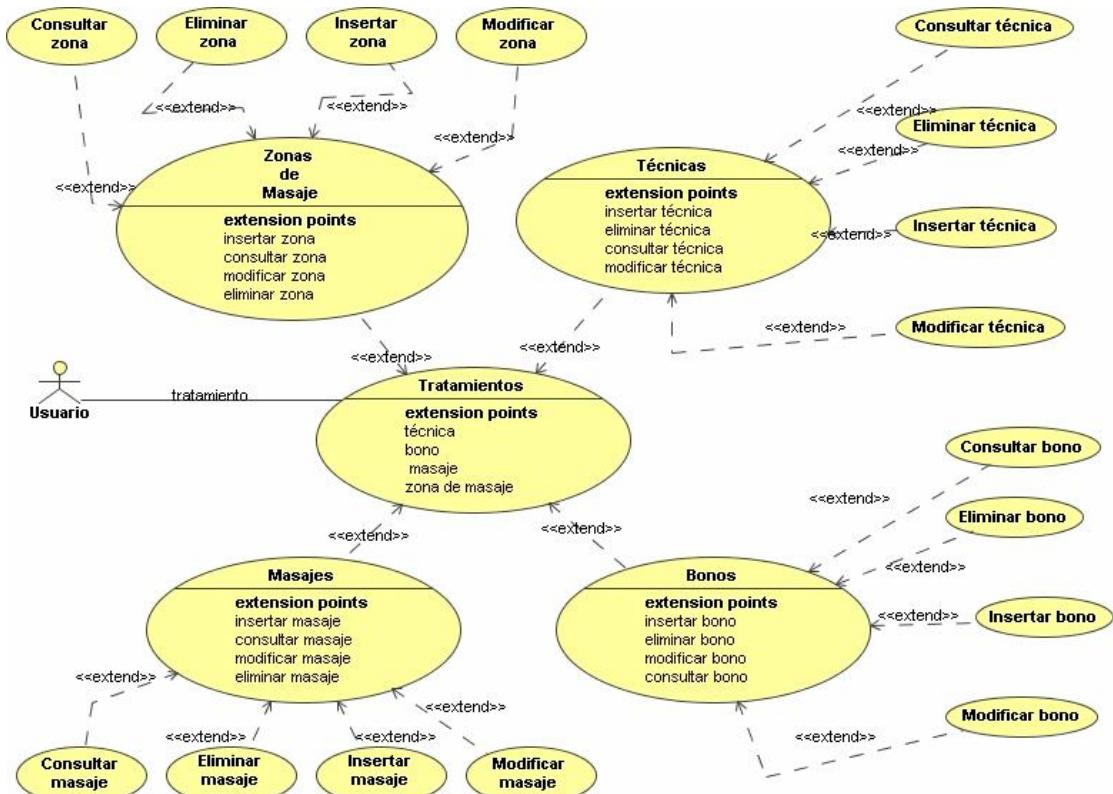


Fig11.Diagrama de caso de uso de Tratamientos

3.2: Diagrama de Clases del proyecto

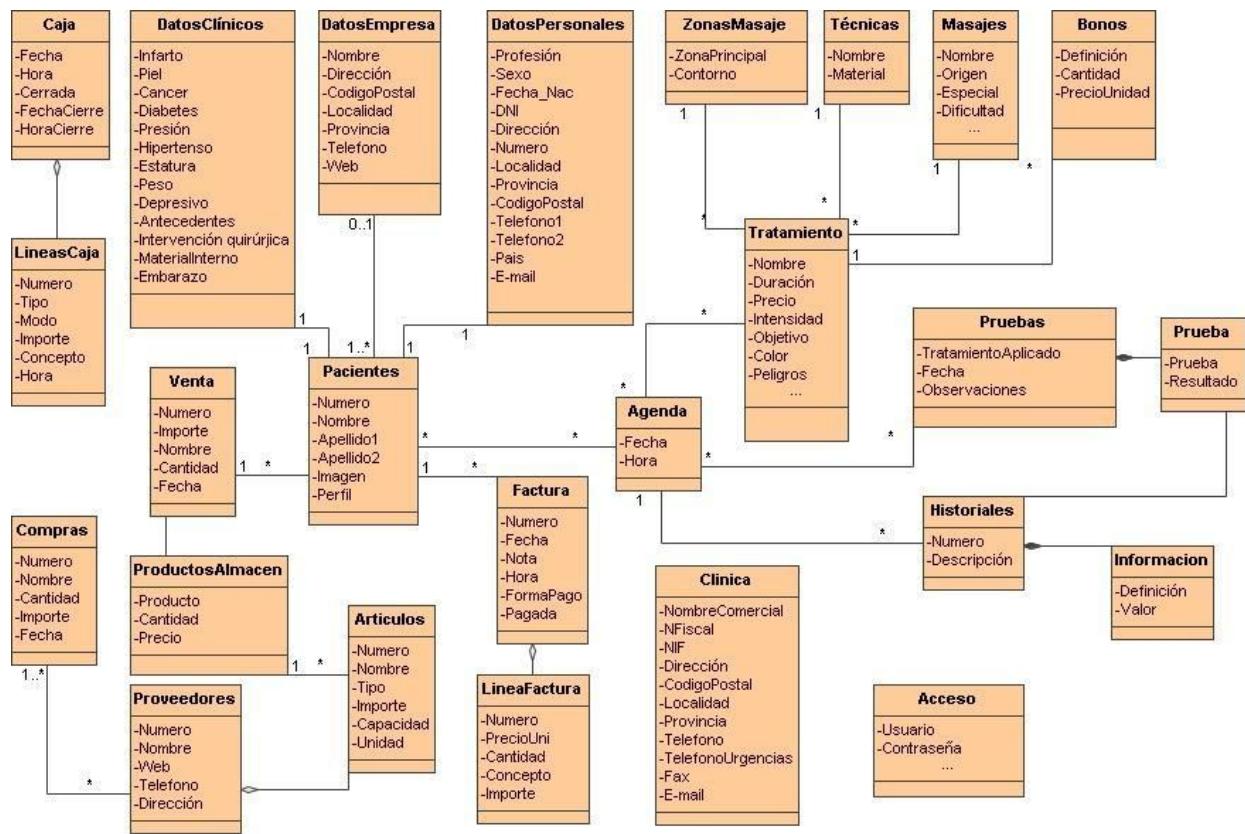


Fig12. Diagrama de clases

3.3: Diagrama de Gestión de Datos del Proyecto

Para realizar la base de datos hemos utilizado el modelo de entidad-relación. Este es el modelo que más se utiliza para el diseño de base de datos. El modelo está formado por las siguientes características:

- Entidad: Es cualquier objeto sobre el que se almacena información. Hay dos tipos de entidades. Está la entidad débil que significa que su existencia depende de otra, y la entidad fuerte que es una entidad que no es débil.
- Relacion: Es una asociación entre dos entidades. Cada relación tiene un nombre que define la función. Existen varios tipos de relación pero lo más común es el de asociación. La asociación puede ser de 1-1, que significa que una tupla de una entidad está asociada con una tupla de otra entidad, de 1-muchos, que significa que una tupla de la primera entidad está relacionada con muchas de la segunda y de esta forma se construye la de muchos-muchos,0-muchos.
- Atributo: Es una característica de una entidad o sobre una relación. Los atributos definen las propiedades más básicas. Por cada atributo se tiene un conjunto de valores
- Identificador: Es un atributo o varios atributos que determina de forma única la tupla de esa entidad. Para esto el identificador debe cumplir dos reglas. Una es que no puede existir dos identificadores iguales, y la otra es que si se elimina un identificador del conjunto la regla anterior deja de cumplirse

En la siguiente imagen se muestra como se compone la base de datos. Para mayor legibilidad se muestra primero como están conectadas las tablas y después de forma independiente los atributos de cada tabla.

Software para la gestión de una consulta de masajes terapéuticos

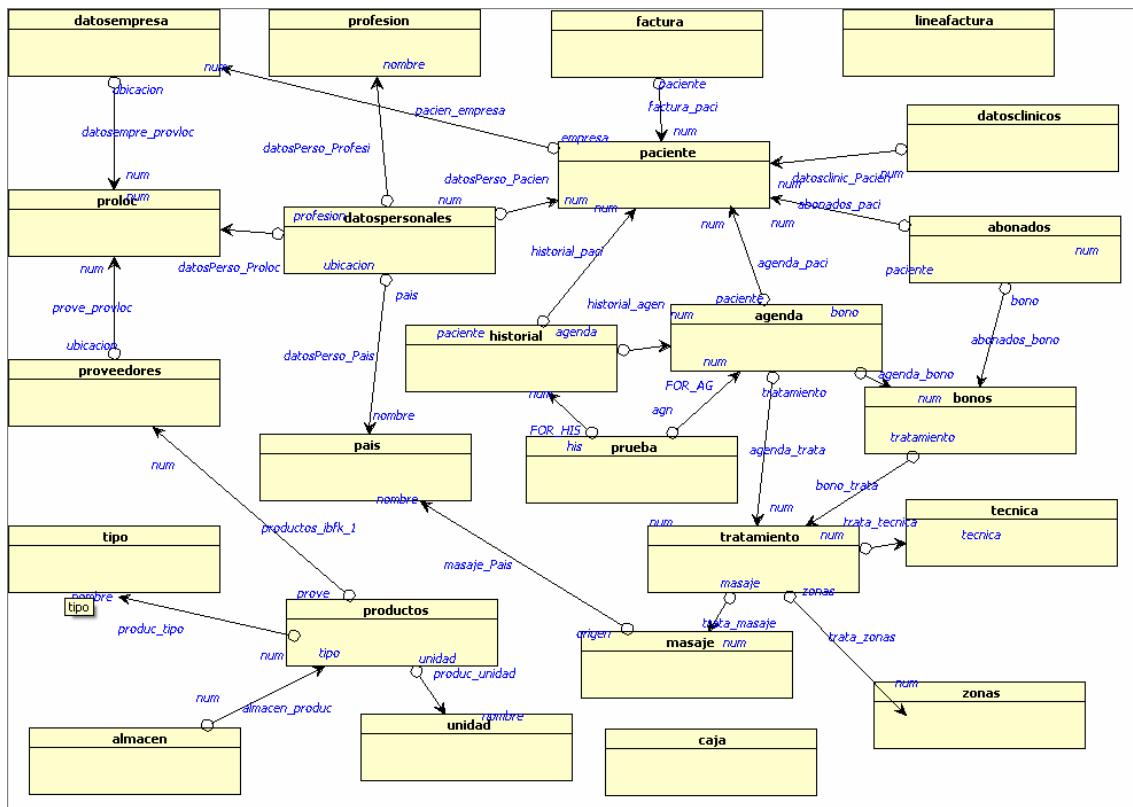
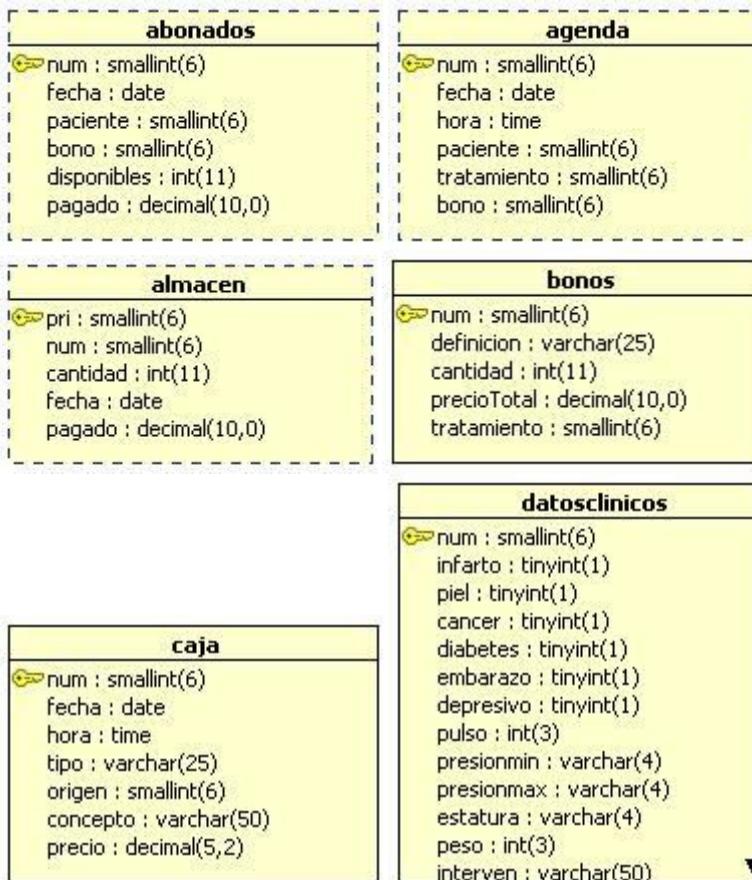


Fig13.Diagrama de Base de Datos General



datosempresa	datospersonales
<p>num : smallint(6) nombre : varchar(25) direccion : varchar(30) codigoP : int(5) telefono : int(12) ubicacion : smallint(6)</p>	<p>num : smallint(6) profesion : varchar(15) sexo : varchar(15) FechaNaci : date direccion : varchar(30) codigoP : int(5) telefono : varchar(12) movil : int(9) pais : varchar(15) estado : varchar(25) ubicacion : smallint(6)</p>
factura	historial
<p>num : smallint(6) fecha : date hora : time paciente : smallint(6) anotacio : text</p>	<p>num : smallint(6) paciente : smallint(6) tb1 : varchar(200) tb2 : varchar(100) tb3 : varchar(100) tb4 : varchar(200) tb5 : varchar(100) tb6 : varchar(300) tb7 : varchar(100) tb8 : varchar(100) tb9 : varchar(100) tb10 : varchar(100) tb11 : varchar(100)</p>
lineafactura	masaje
<p>num : smallint(6) nombre : varchar(100) precio : decimal(10,0)</p>	<p>num : smallint(6) nombre : varchar(25) origen : varchar(10) especial : varchar(100) dificultad : varchar(25) precaucion : varchar(100)</p>
paciente	productos
<p>num : smallint(6) nombre : varchar(25) apellido1 : varchar(30) apellido2 : varchar(30) empresa : smallint(6) dni : varchar(10) imagen : text email : varchar(100) activo : tinyint(4)</p>	<p>num : smallint(6) nombre : varchar(50) tipo : varchar(15) importe : int(11) capacidad : int(11) unidad : varchar(15) prove : smallint(6)</p>

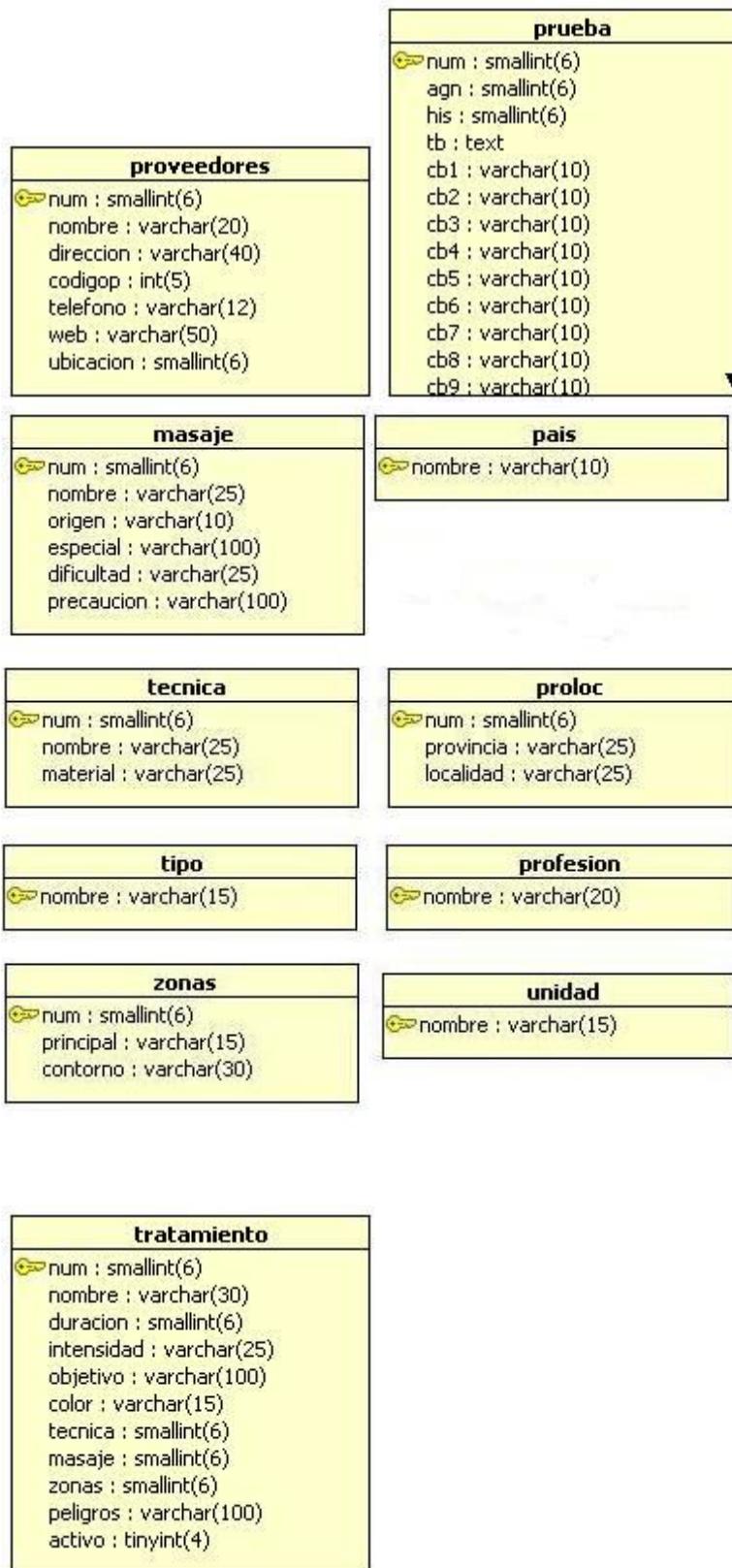


Fig14.Diagrama de Base de Datos Expandido

3.4: Diagrama de Interacción del Proyecto

Definición:

Un diagrama de secuencia es de los diagramas más útiles para el modelo de interacción entre los objetos de un sistema de información. Hay que realizar un diagrama de secuencia para cada caso de uso. Sabemos que en un caso de uso permite crear una visión del escenario, en cambio para el diagrama de secuencia contiene características de implementación del escenario, puede incluir objetos y clases que se necesitan para implementar el escenario, así como los mensajes entre objetos.

Para el diseño de un diagrama de secuencia, se examina el comportamiento de un caso de uso para concretar cuales son los objetos que se necesitan para la implementación. Si se posee un modelo de la descripción para cada caso de uso a través de una secuencia de varios pasos, entonces podemos viajar sobre esos pasos para deducir los objetos necesarios.

Por lo tanto en un diagrama de secuencia se muestra los objetos que actúan en el escenario con líneas discontinuas verticales, llamadas líneas de vida, mientras que los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se muestran cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos no tiene orden.

Diagramas elaborados:

A continuación se muestran los diagramas elaborados. Debido a que muchas ventanas parten desde la misma idea y se desea la misma implementación se procede a omitir los diagramas que sean idénticos.

Los diagramas omitidos son los siguientes:

- Todas las ventanas que sean de mantenimiento de tablas del estilo profesión, unidad, ubicación, países... tendrán el comportamiento del diagrama de “Tabla”.

- Se ha expuesto el diagrama de Nuevo Tratamiento y Nuevo Proveedor. Los diagramas de Buscar, Consultar, Editar y Eliminar Tratamiento o Proveedor siguen el mismo comportamiento que el de Paciente respectivamente.

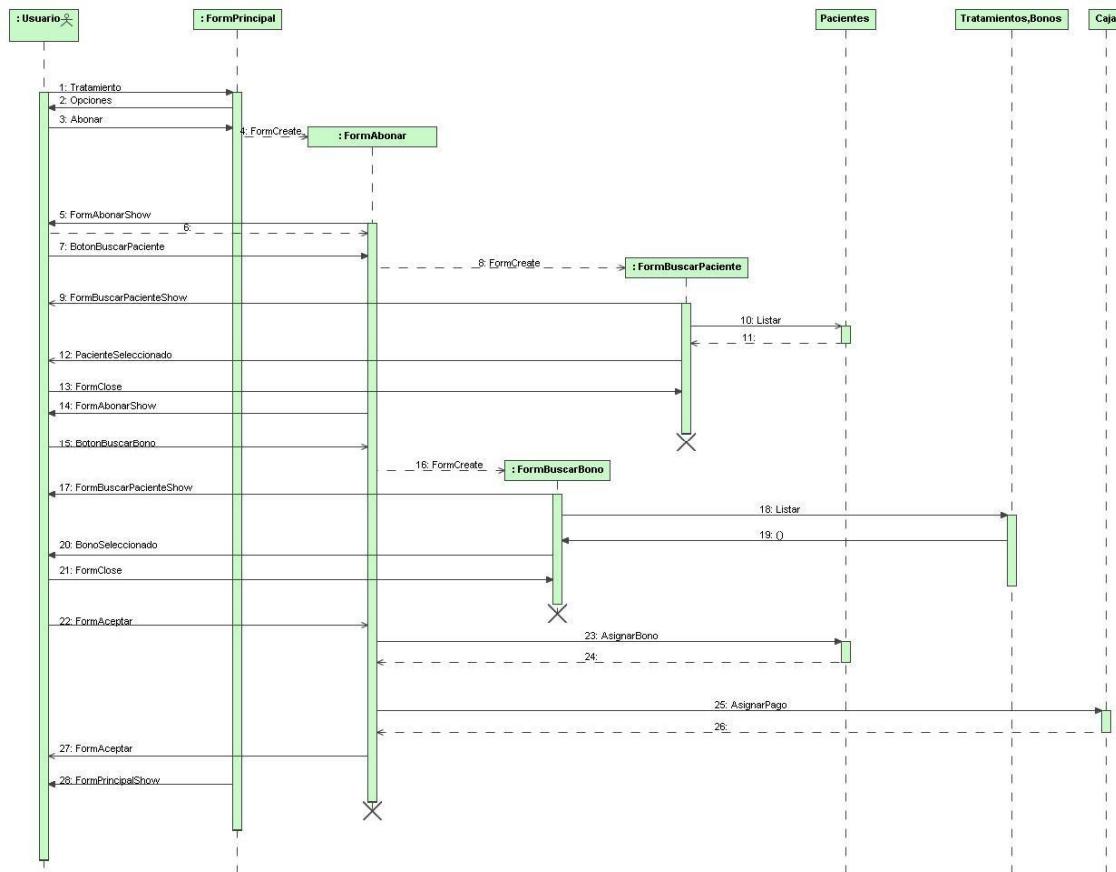


Fig15. Diagrama de secuencia de Abonar Paciente

Software para la gestión de una consulta de masajes terapéuticos

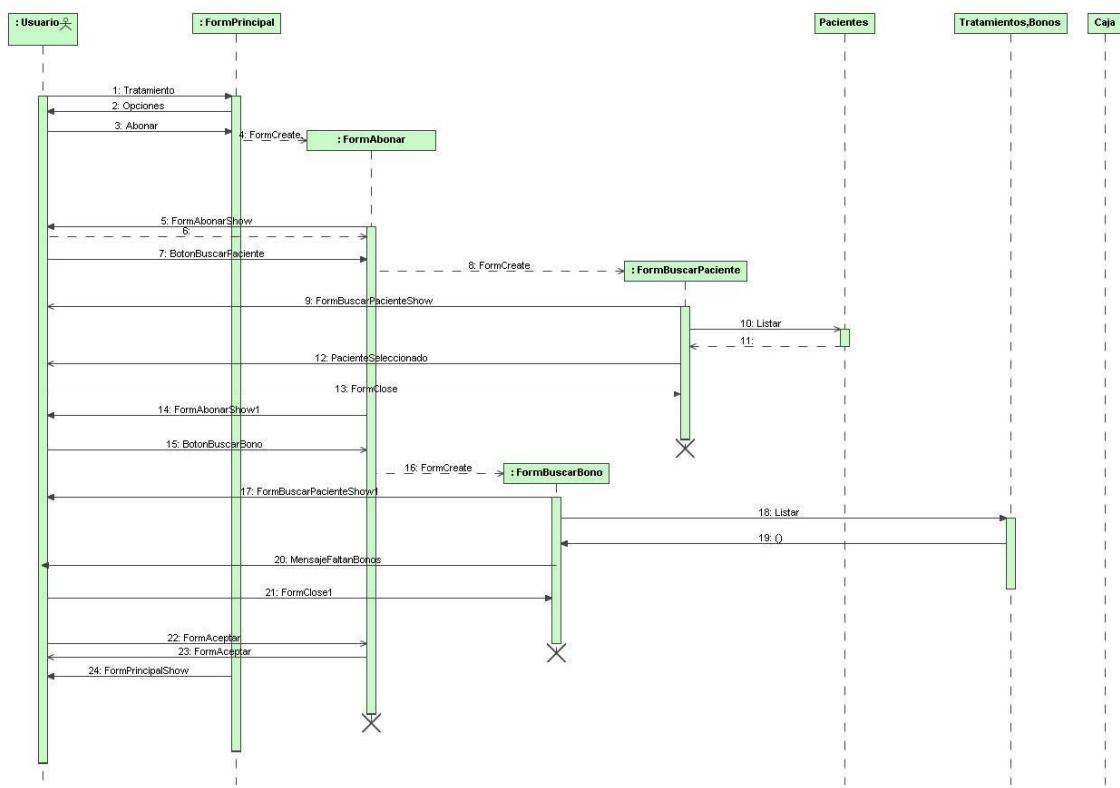
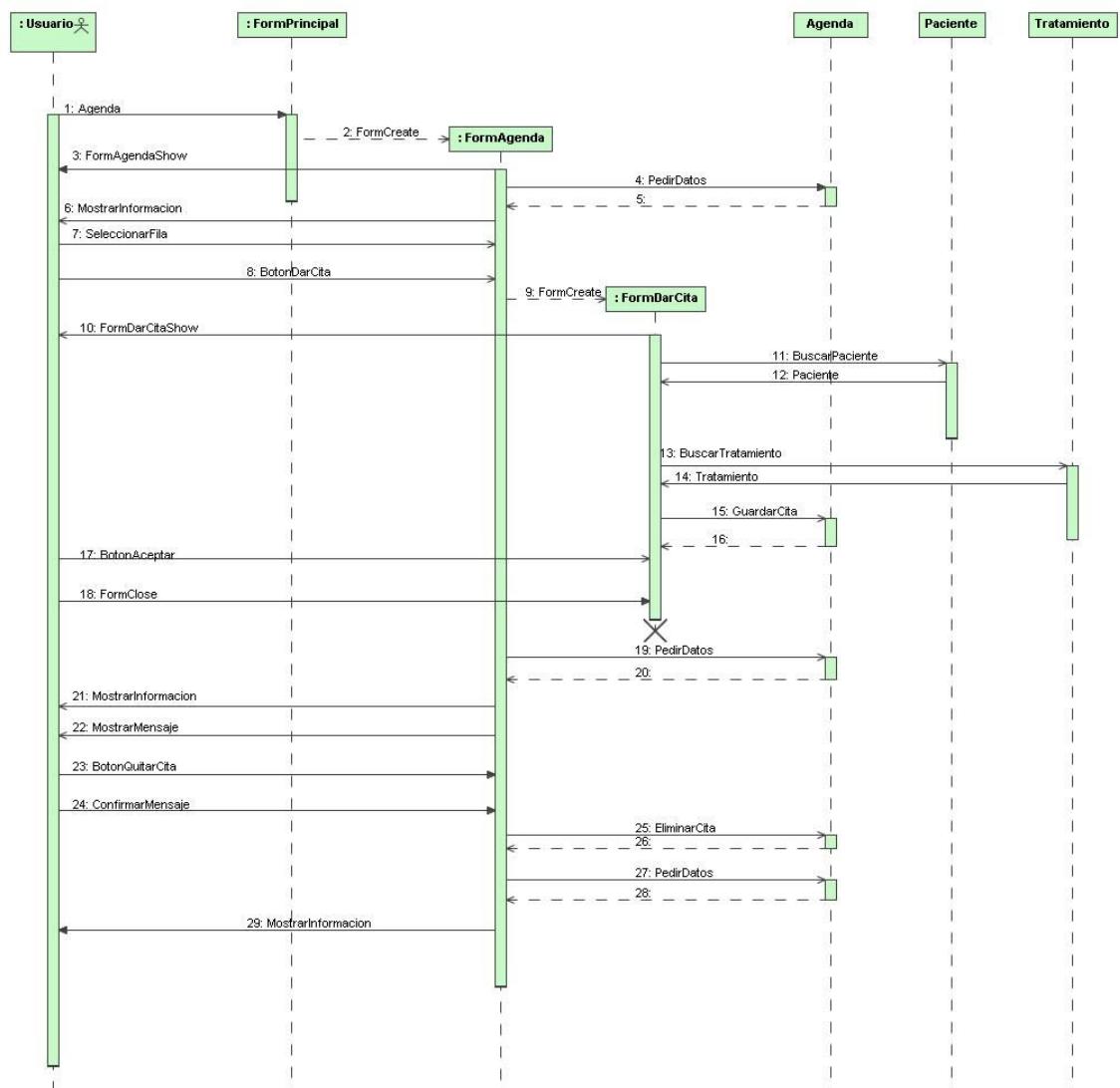


Fig 16. Diagrama de secuencia de Abonar Paciente Excepción

Software para la gestión de una consulta de masajes terapéuticos



Software para la gestión de una consulta de mensajes terapéuticos

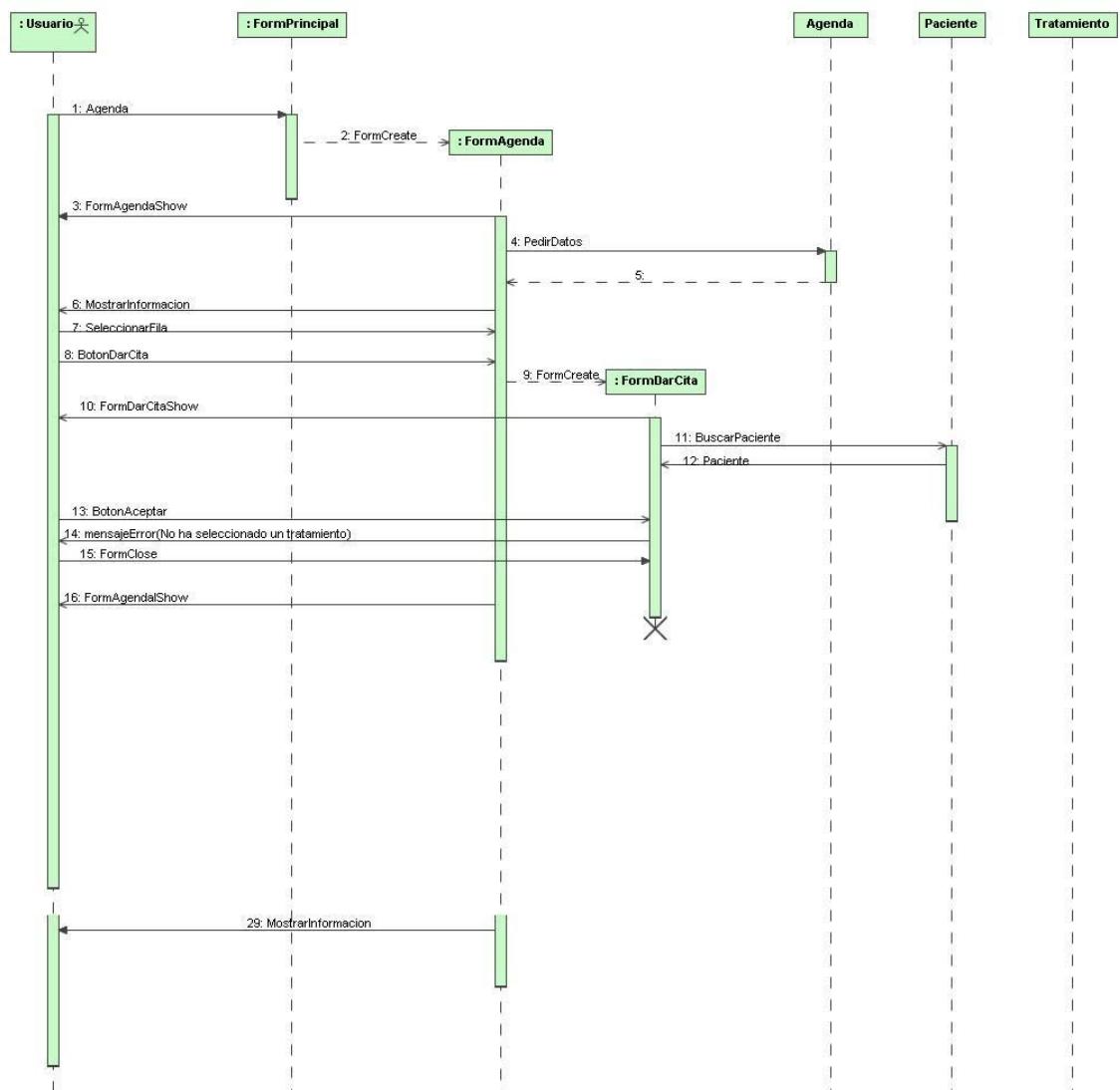


Fig18.Diagrama de secuencia de Agenda Excepción

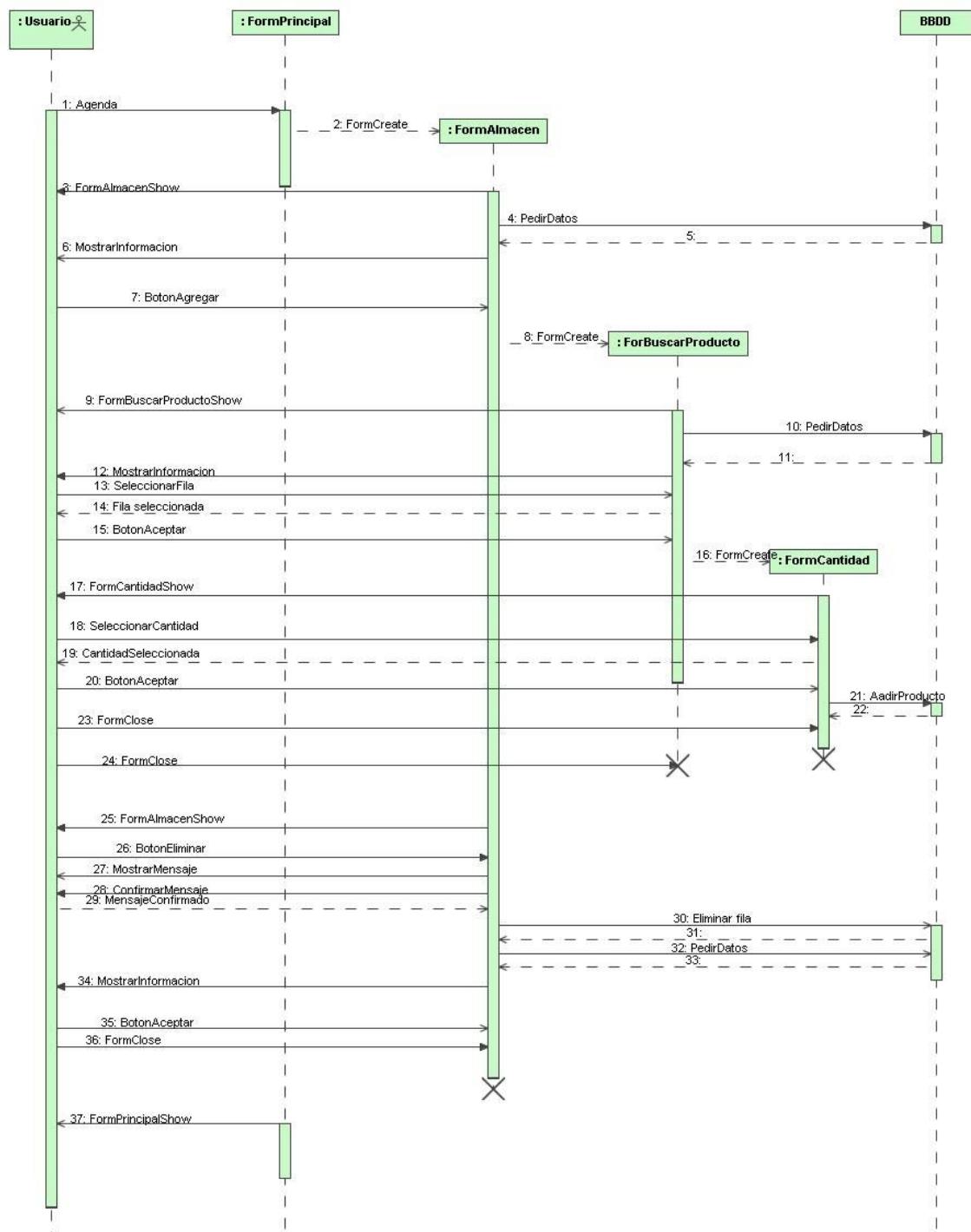


Fig19.Diagrama de secuencia de Almacén

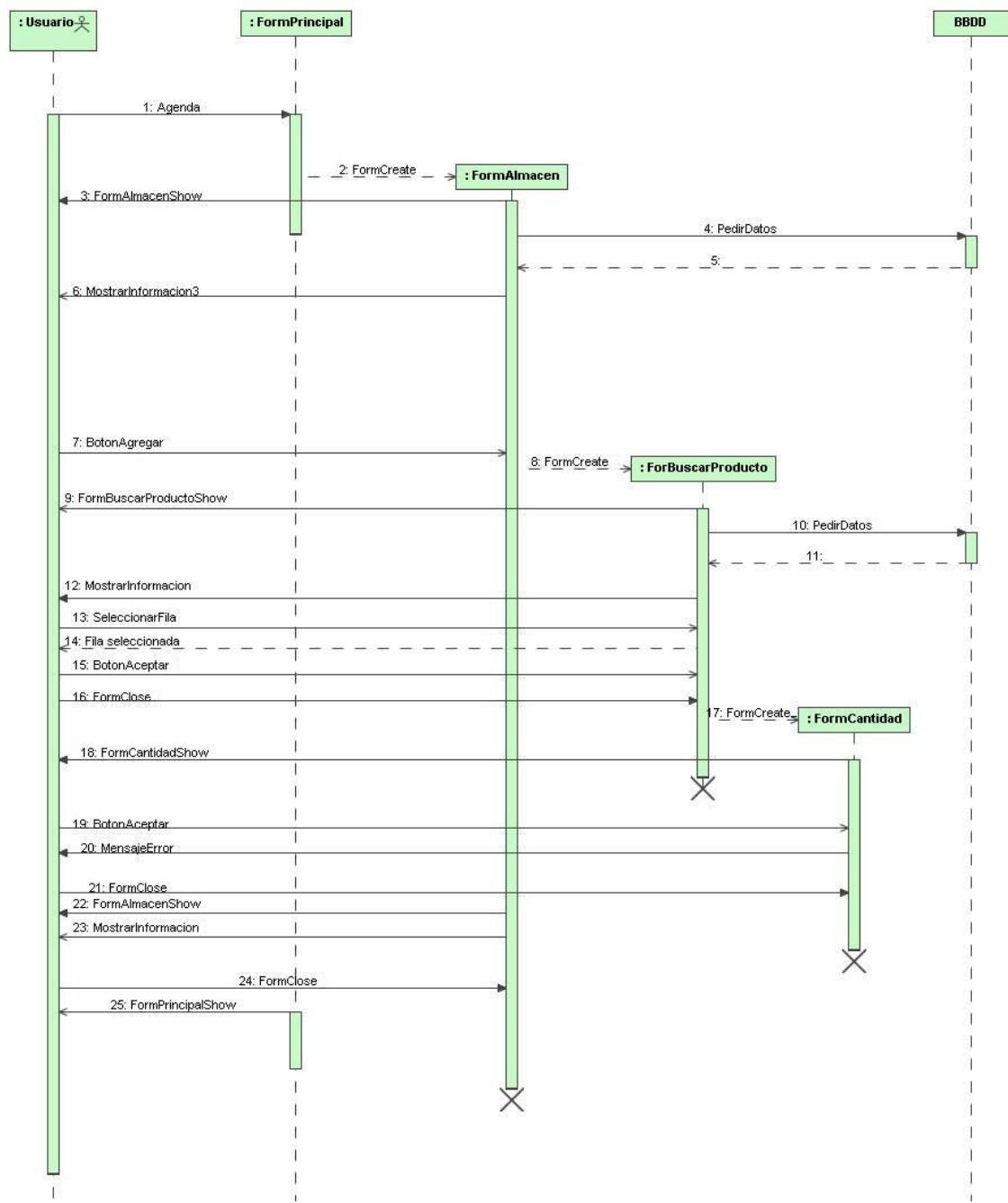


Fig20. Diagrama de secuencia de Almacén Excepción

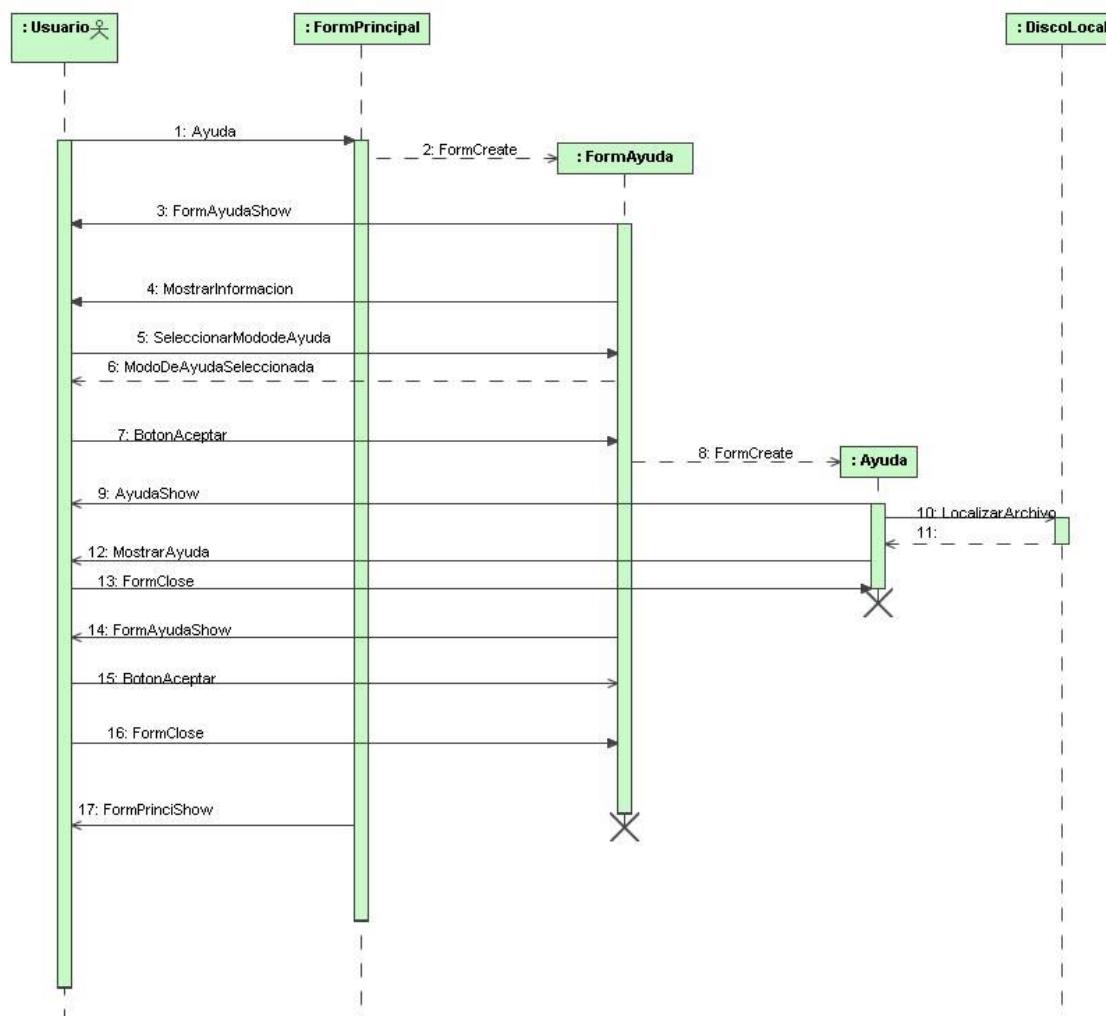


Fig21. Diagrama de secuencia de Ayuda

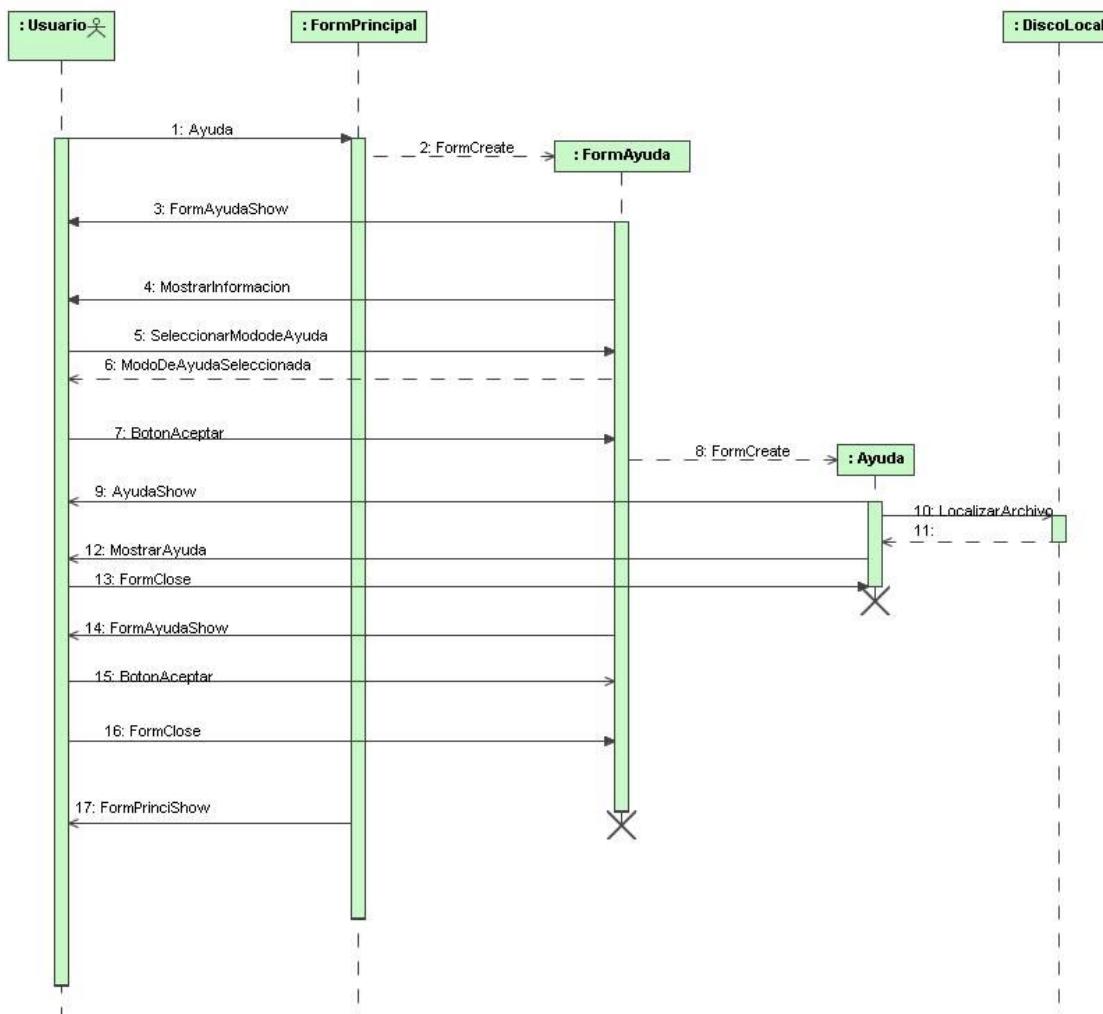


Fig22.Diagrama de secuencia de Ayuda Excepción

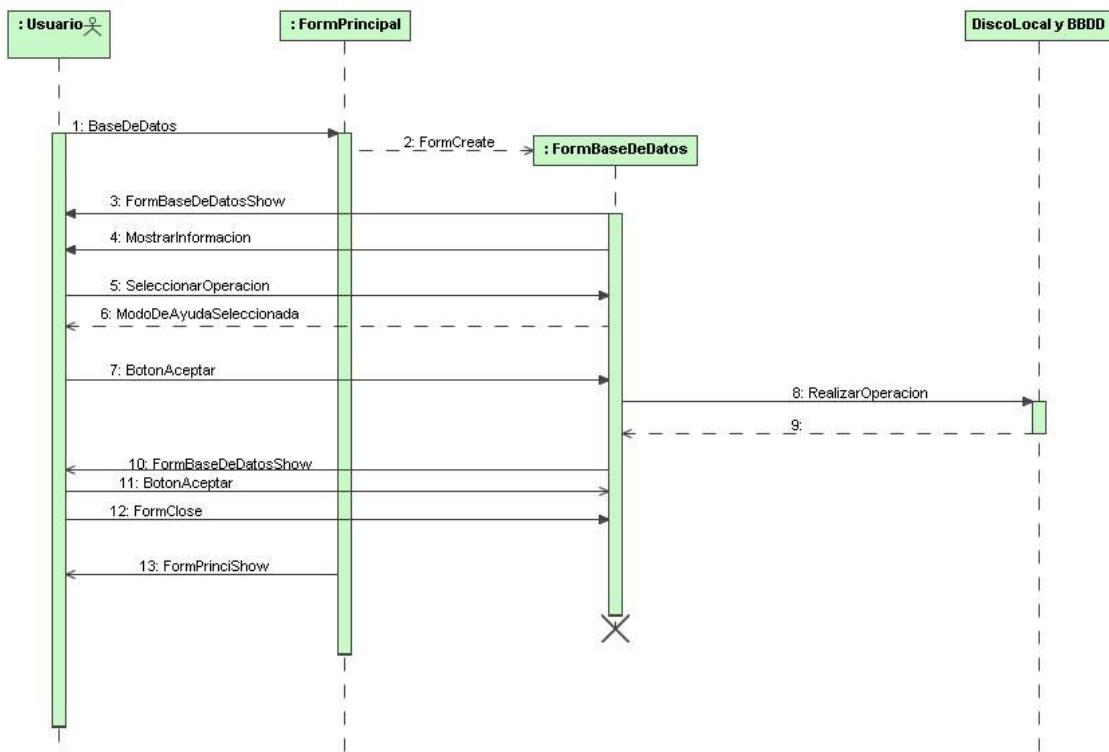


Fig23.Diagrama de secuencia de Base de Datos

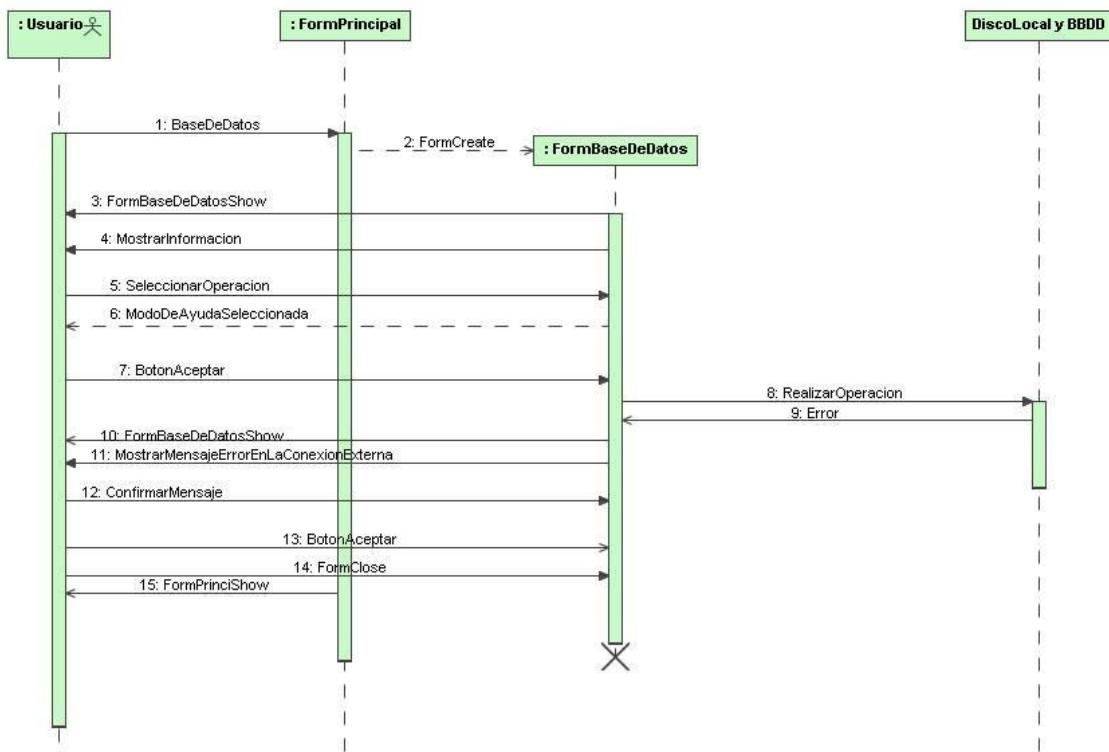


Fig24.Diagrama de secuencia de Base de Datos Excepción

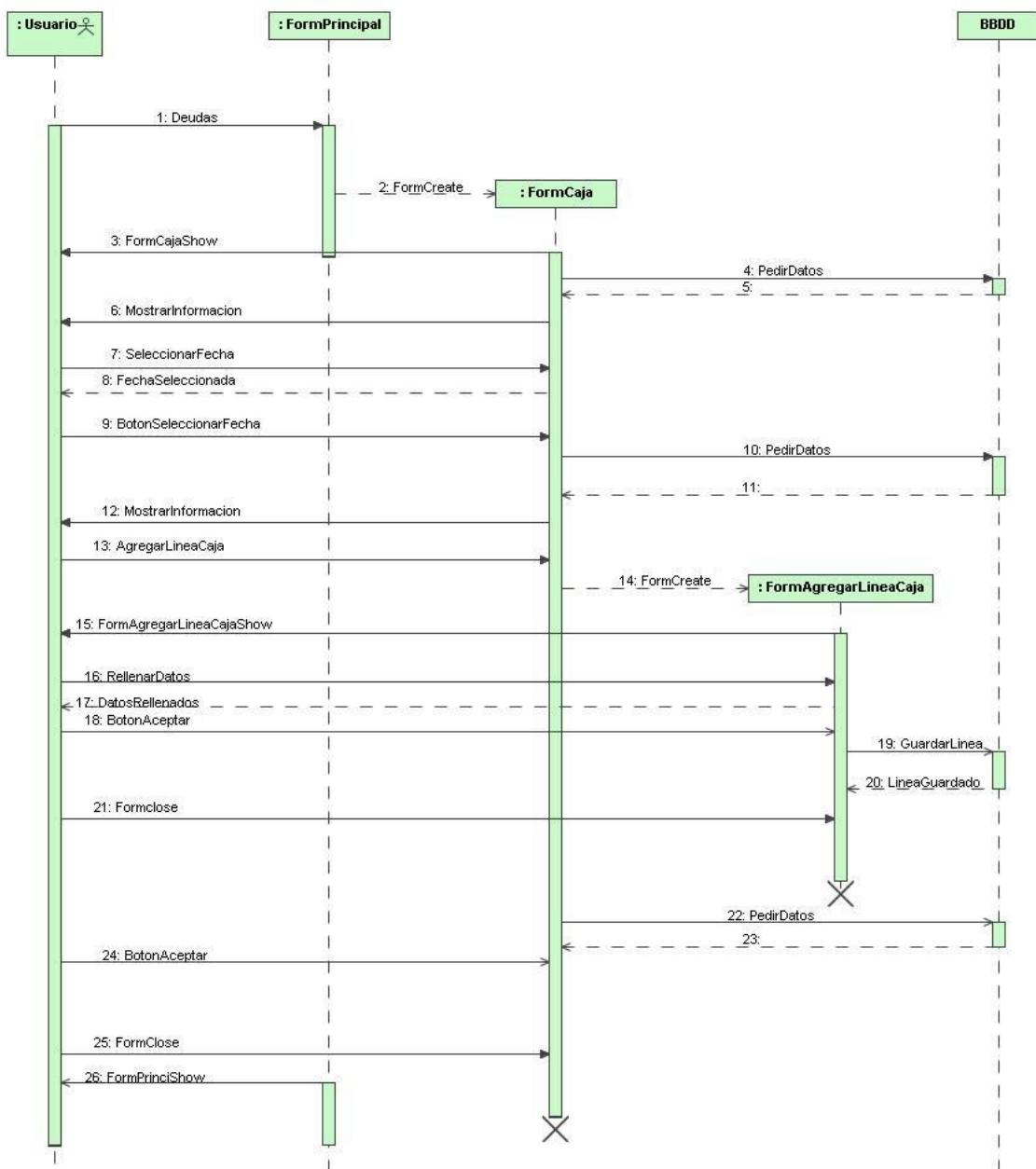


Fig25.Diagrama de secuencia de Caja

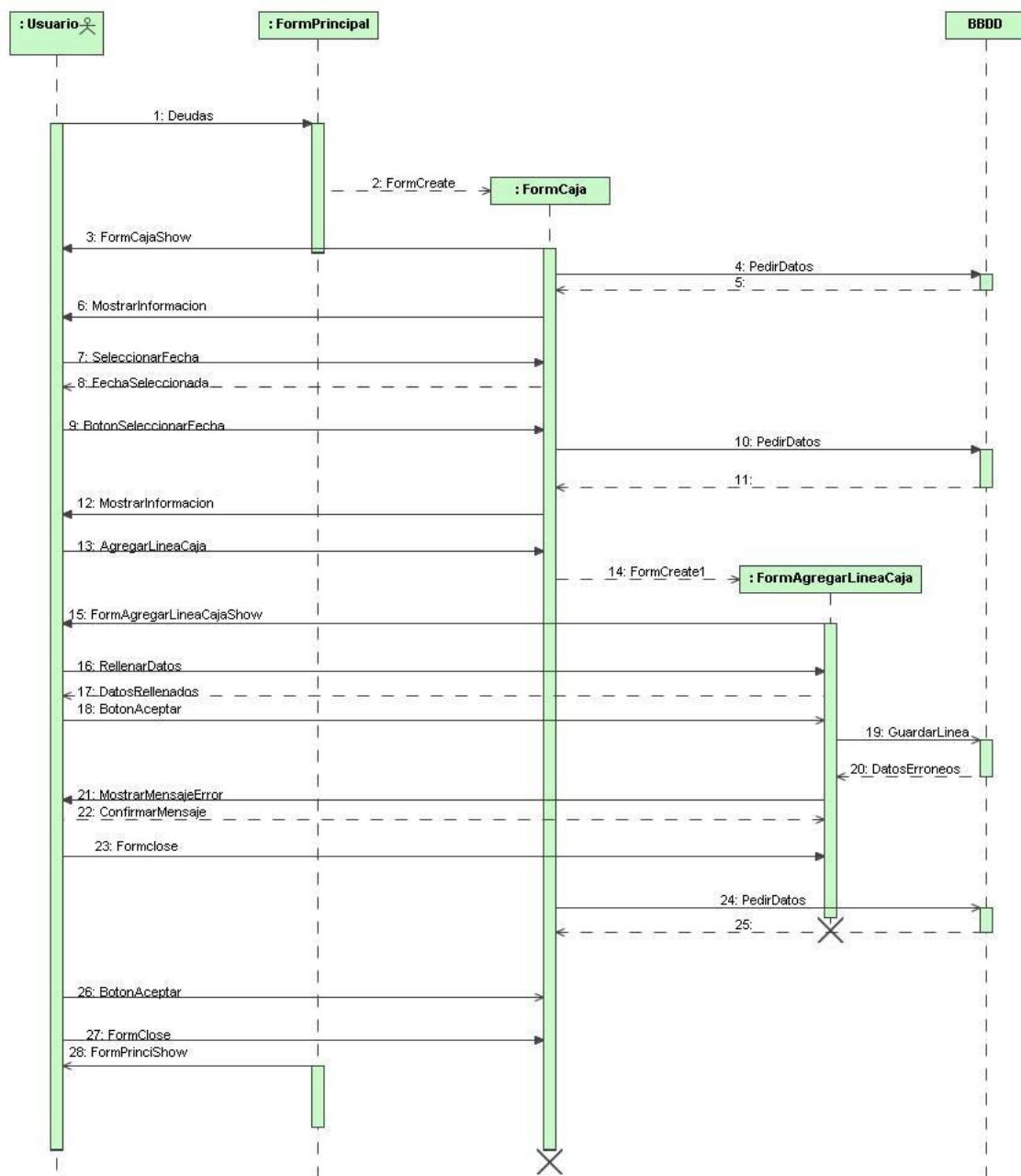


Fig26.Diagrama de secuencia de Caja Excepción

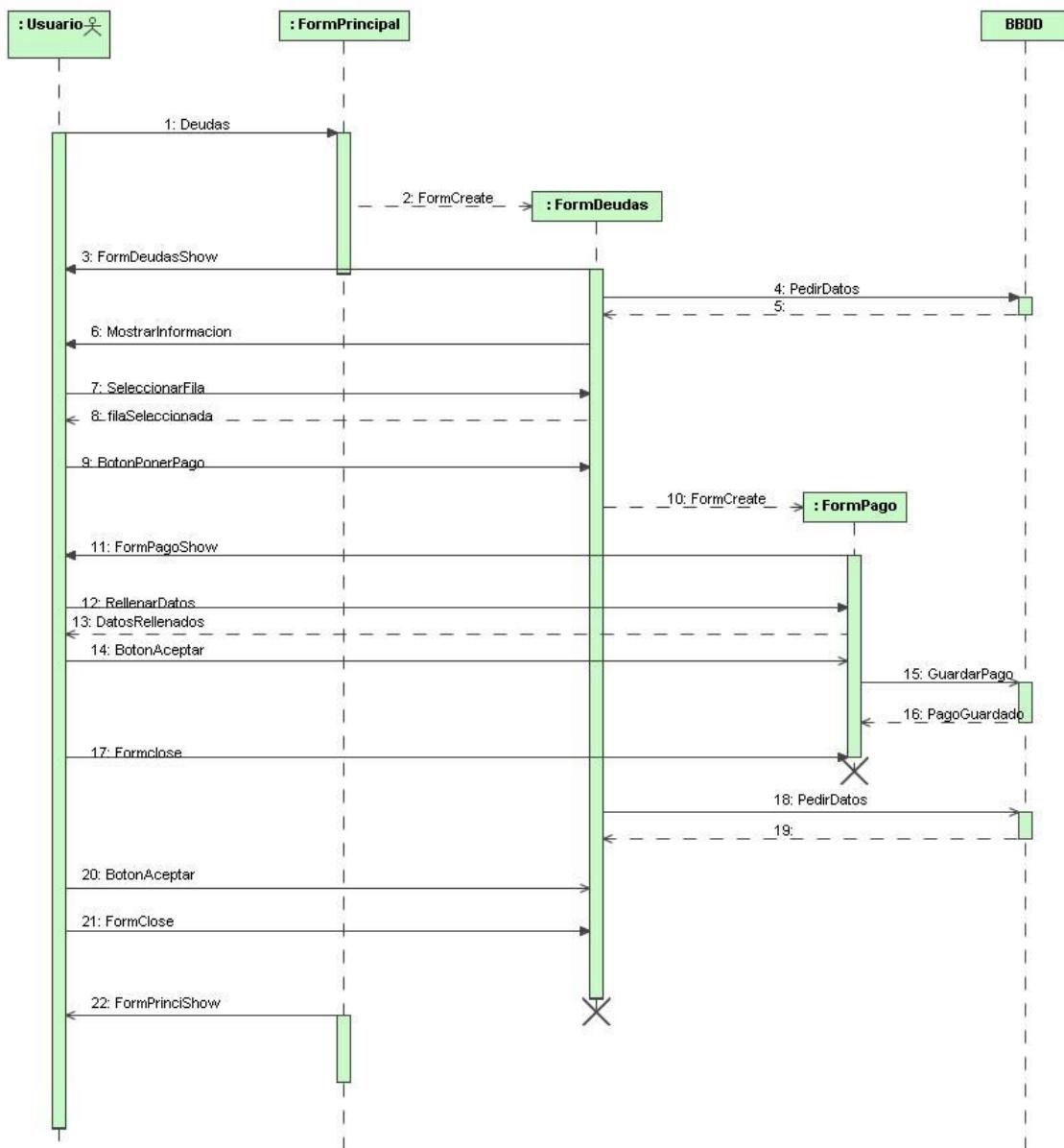
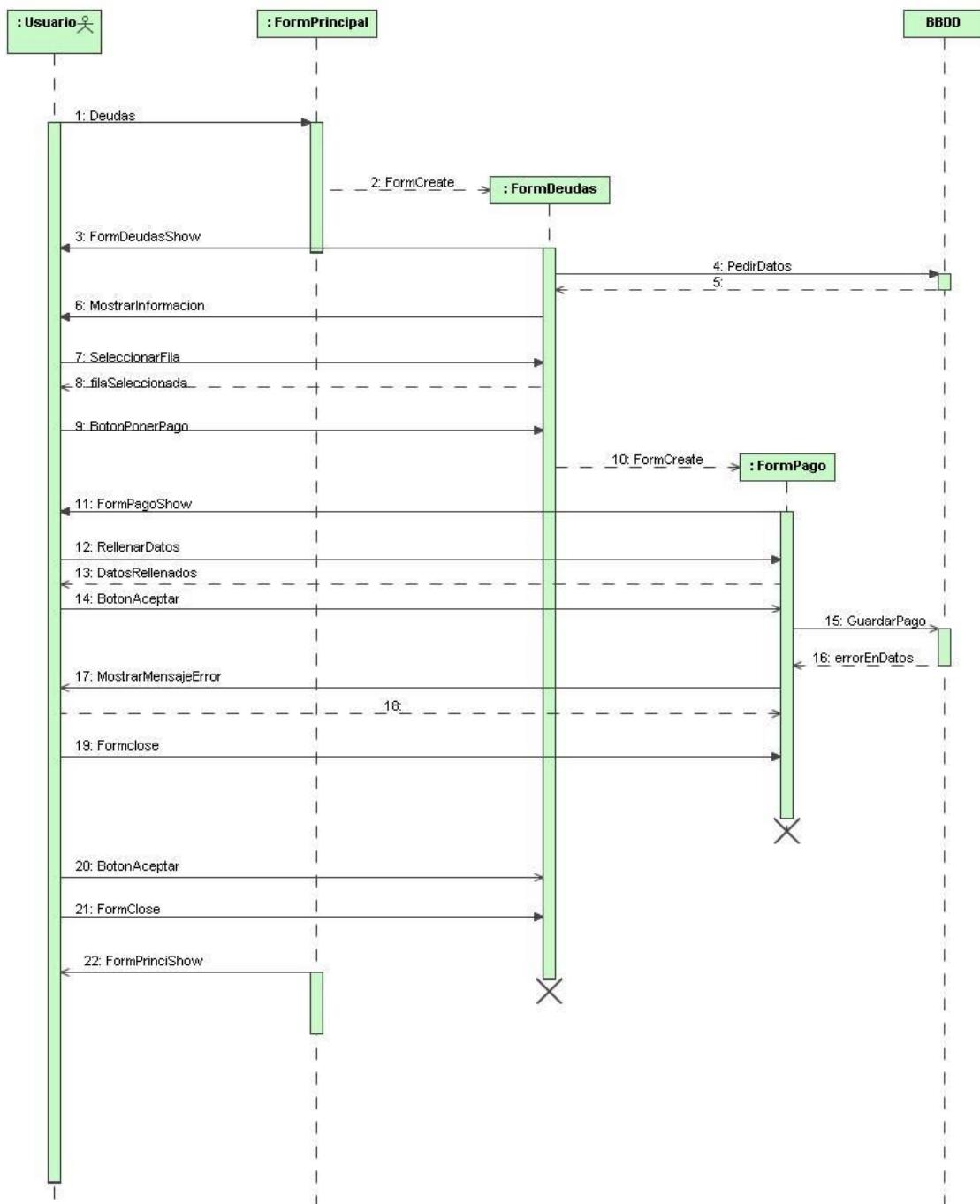


Fig27. Diagrama de secuencia de Deudas



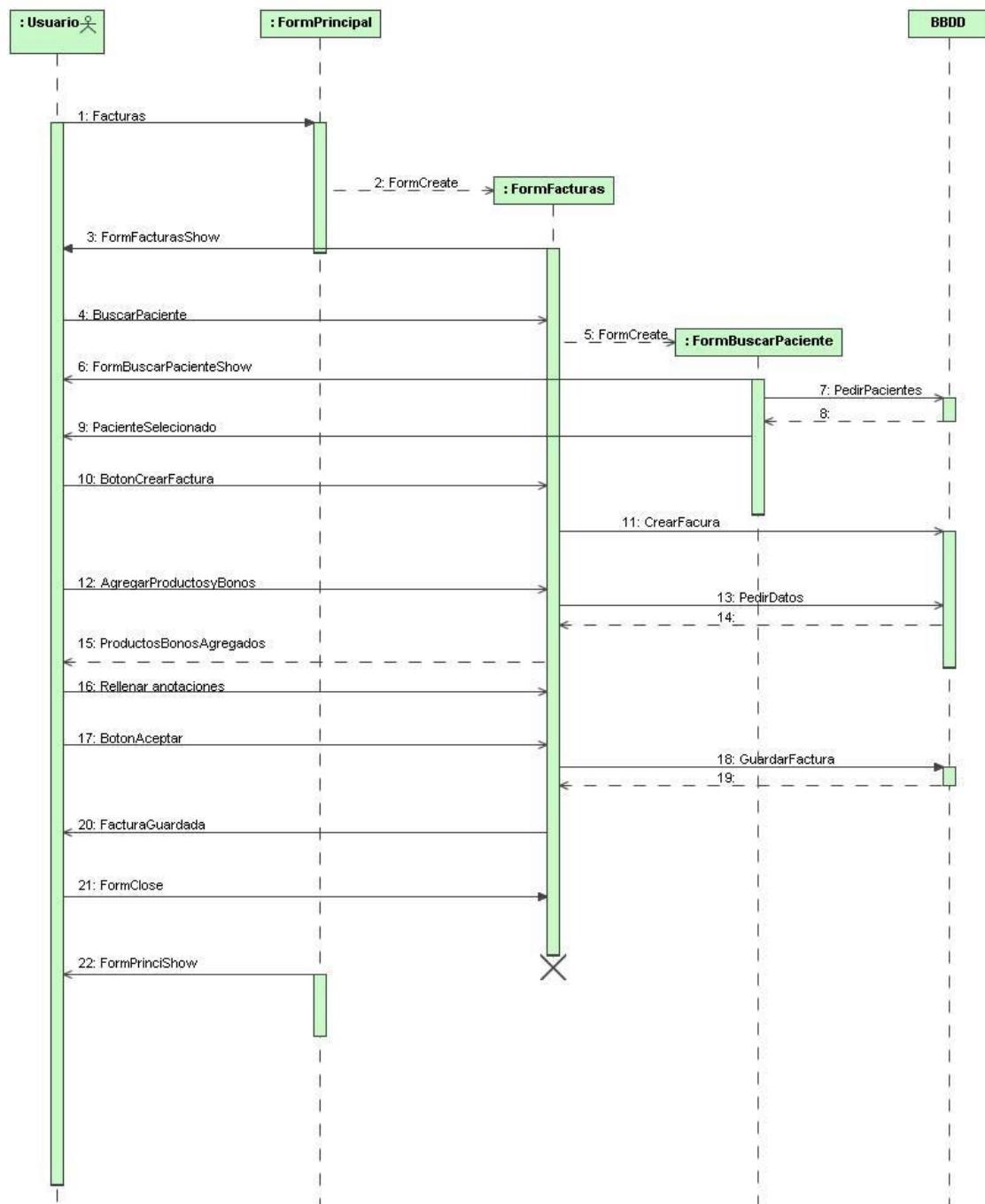


Fig29. Diagrama de secuencia de facturas

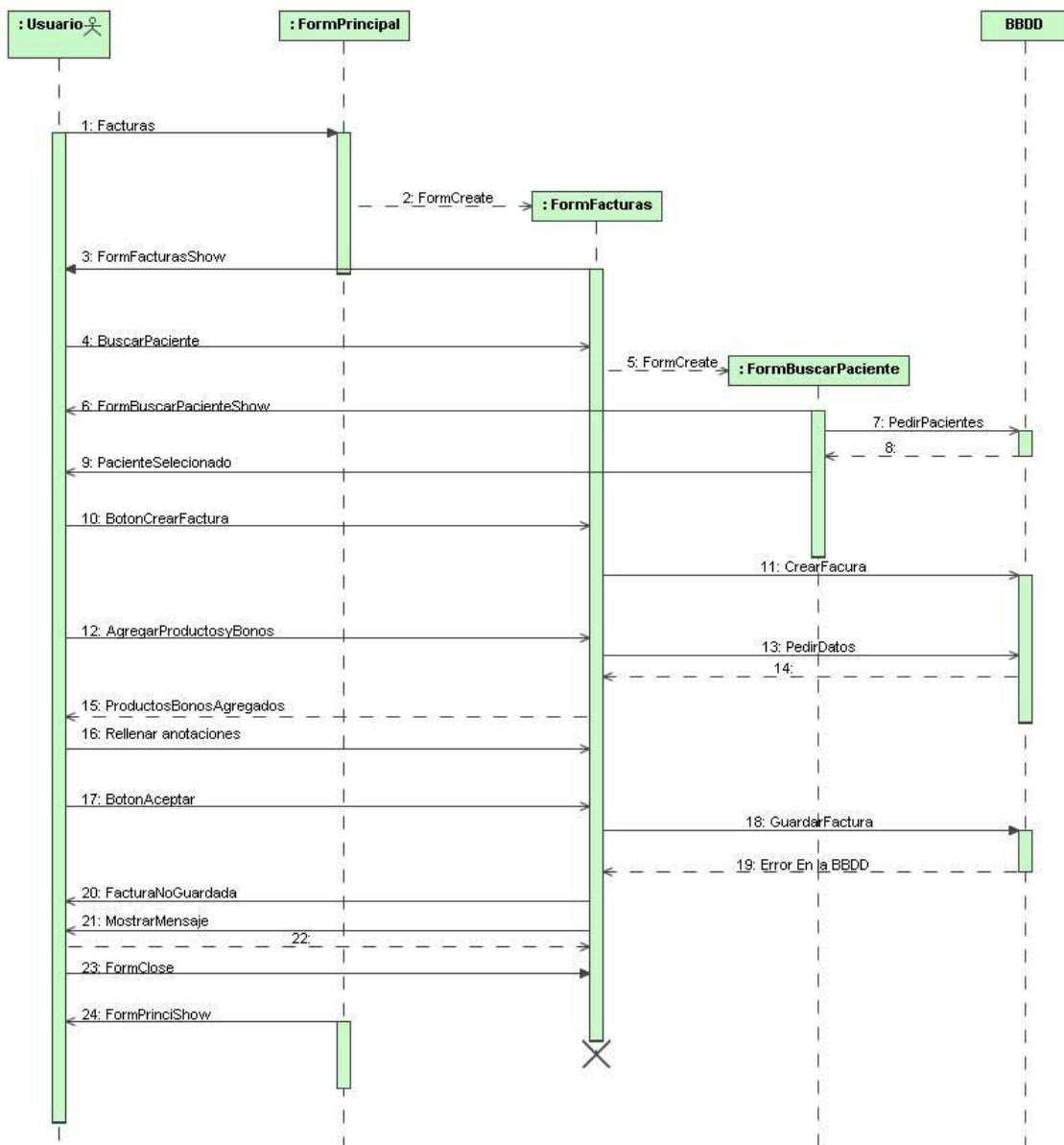


Fig30.Diagrama de secuencia de Facturas Excepción

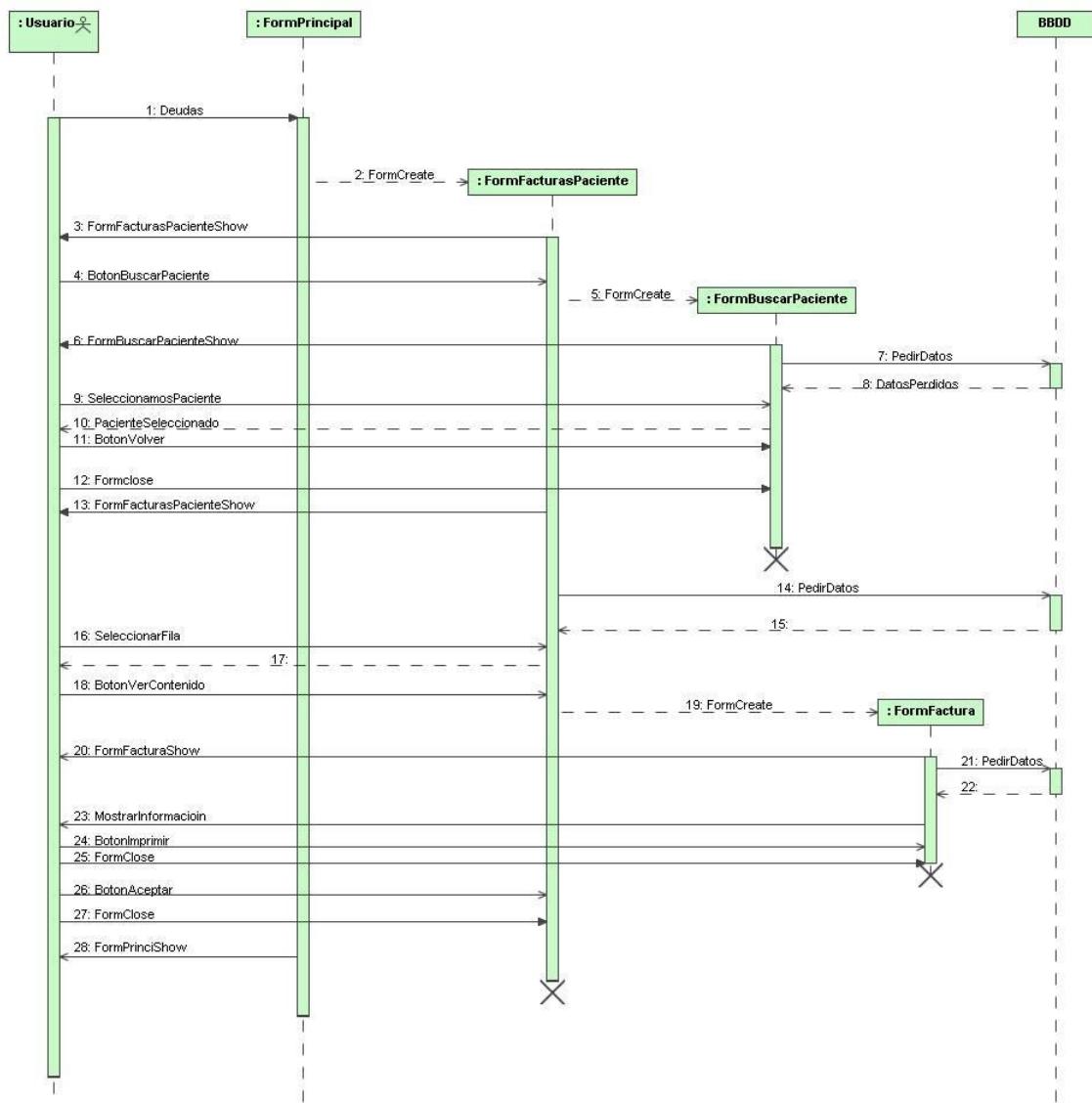


Fig31.Diagrama de secuencia de Facturas Por Paciente

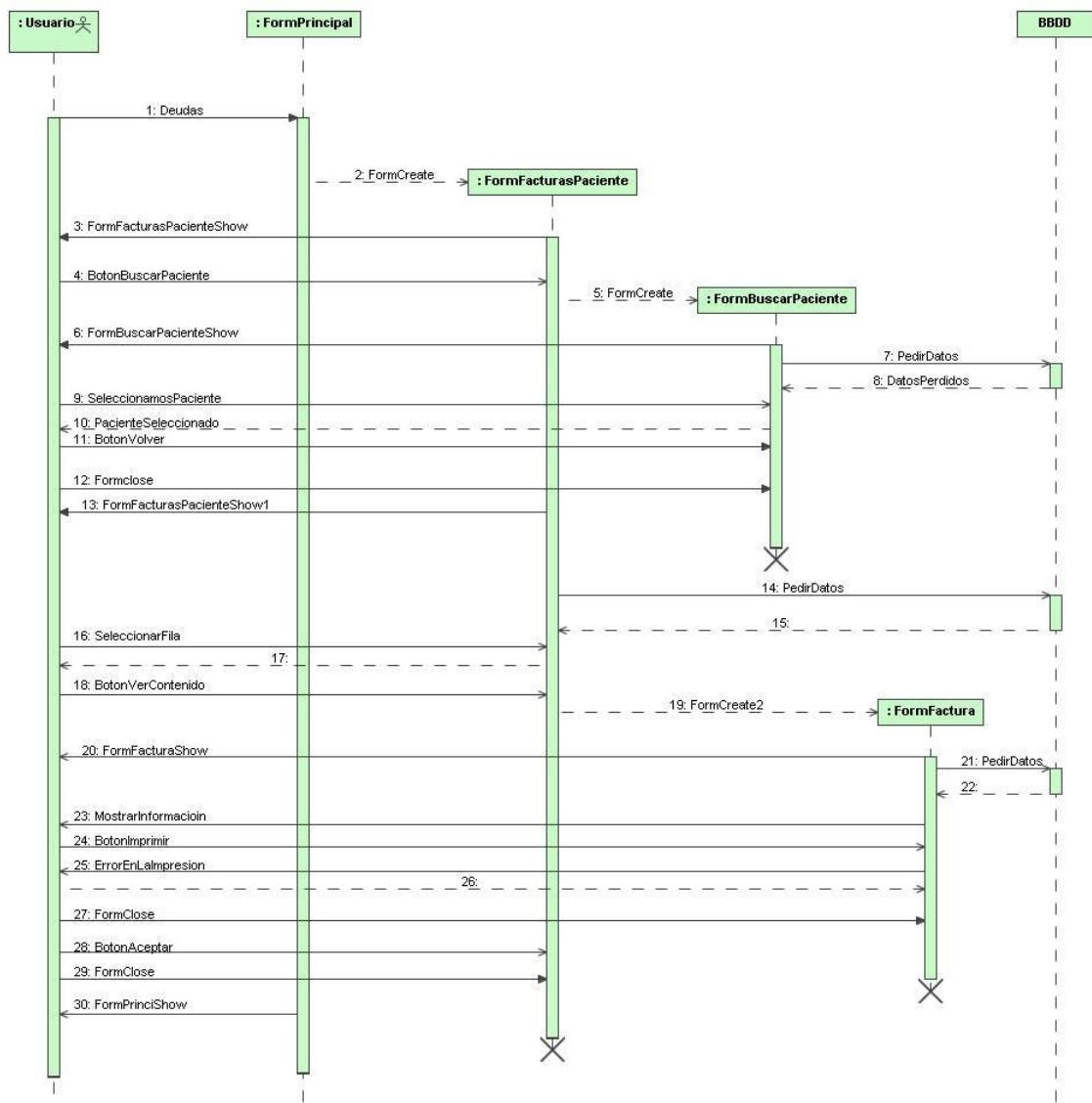


Fig32.Diagrama de secuencia de Facturas Por Paciente Excepción

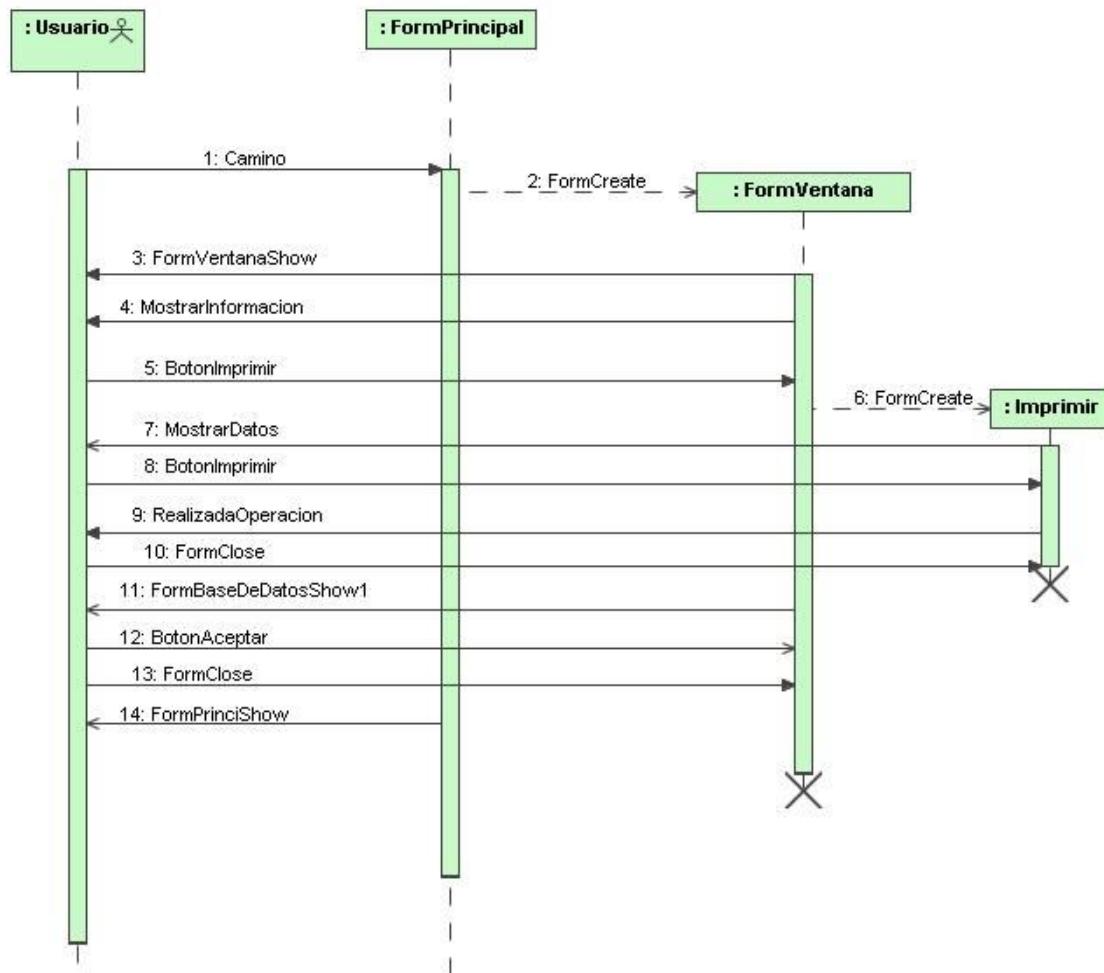


Fig33. Diagrama de secuencia de Imprimir

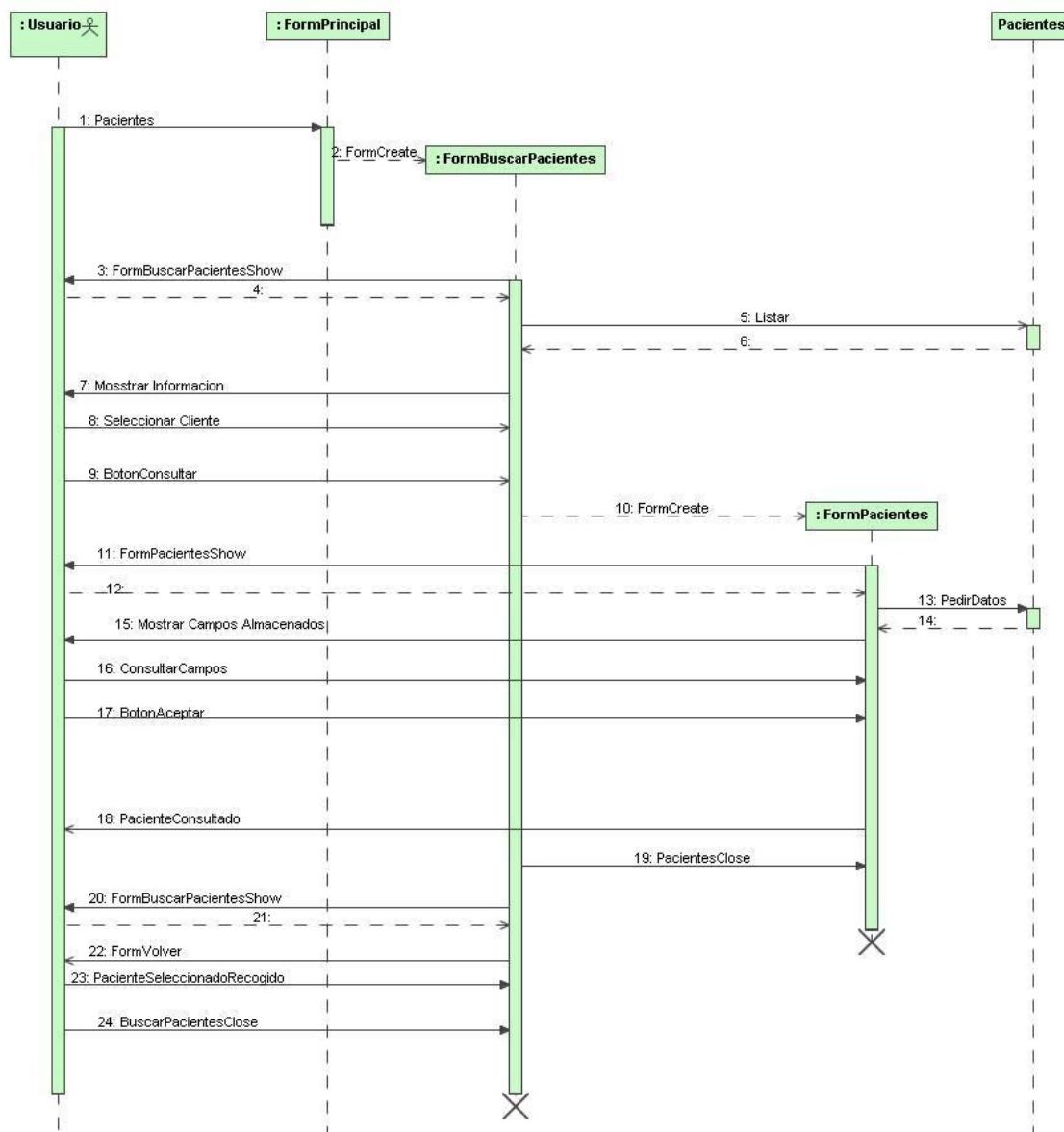


Fig34. Diagrama de secuencia de Paciente Consultar

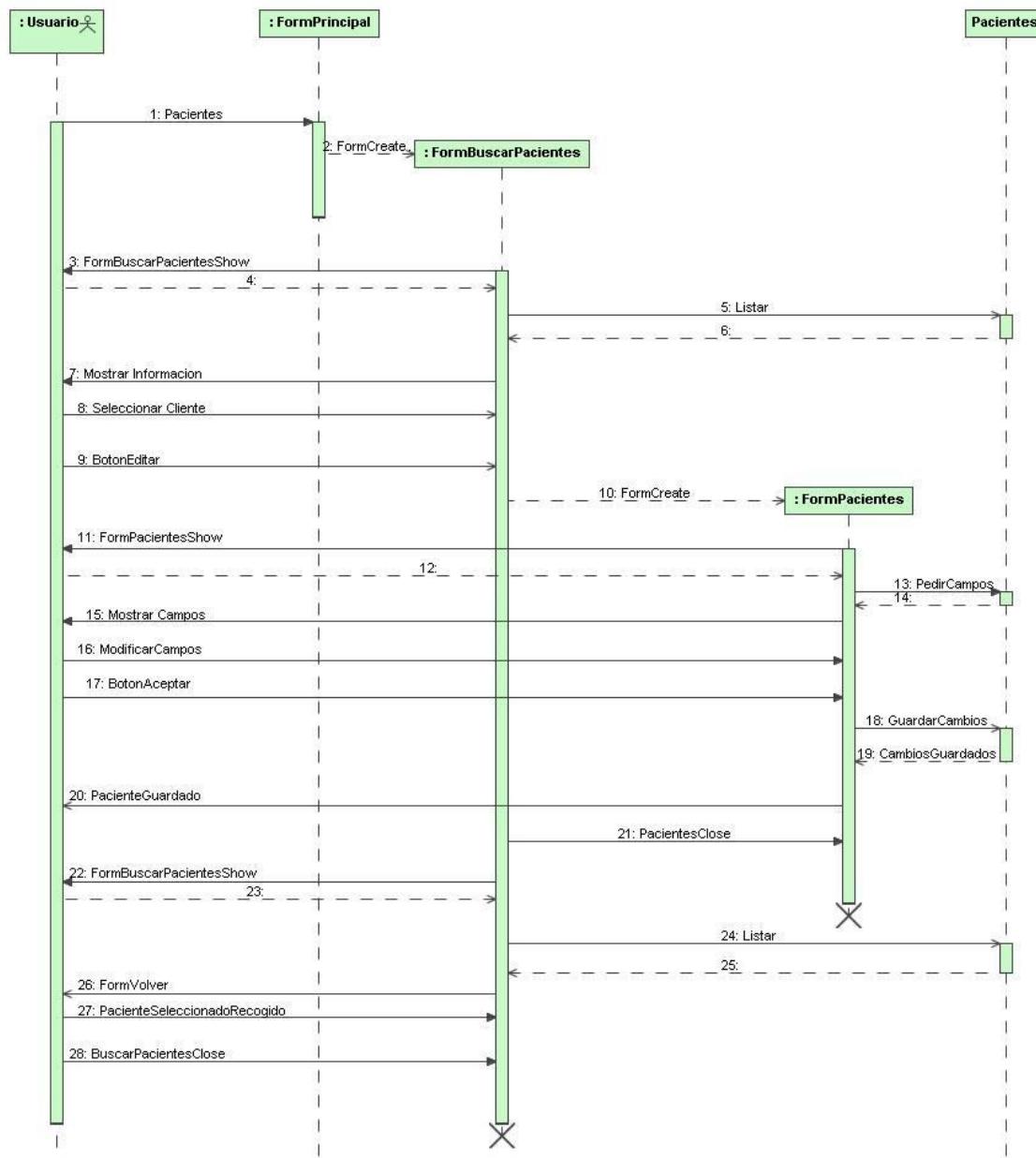


Fig35.Diagrama de secuencia de Paciente Editar

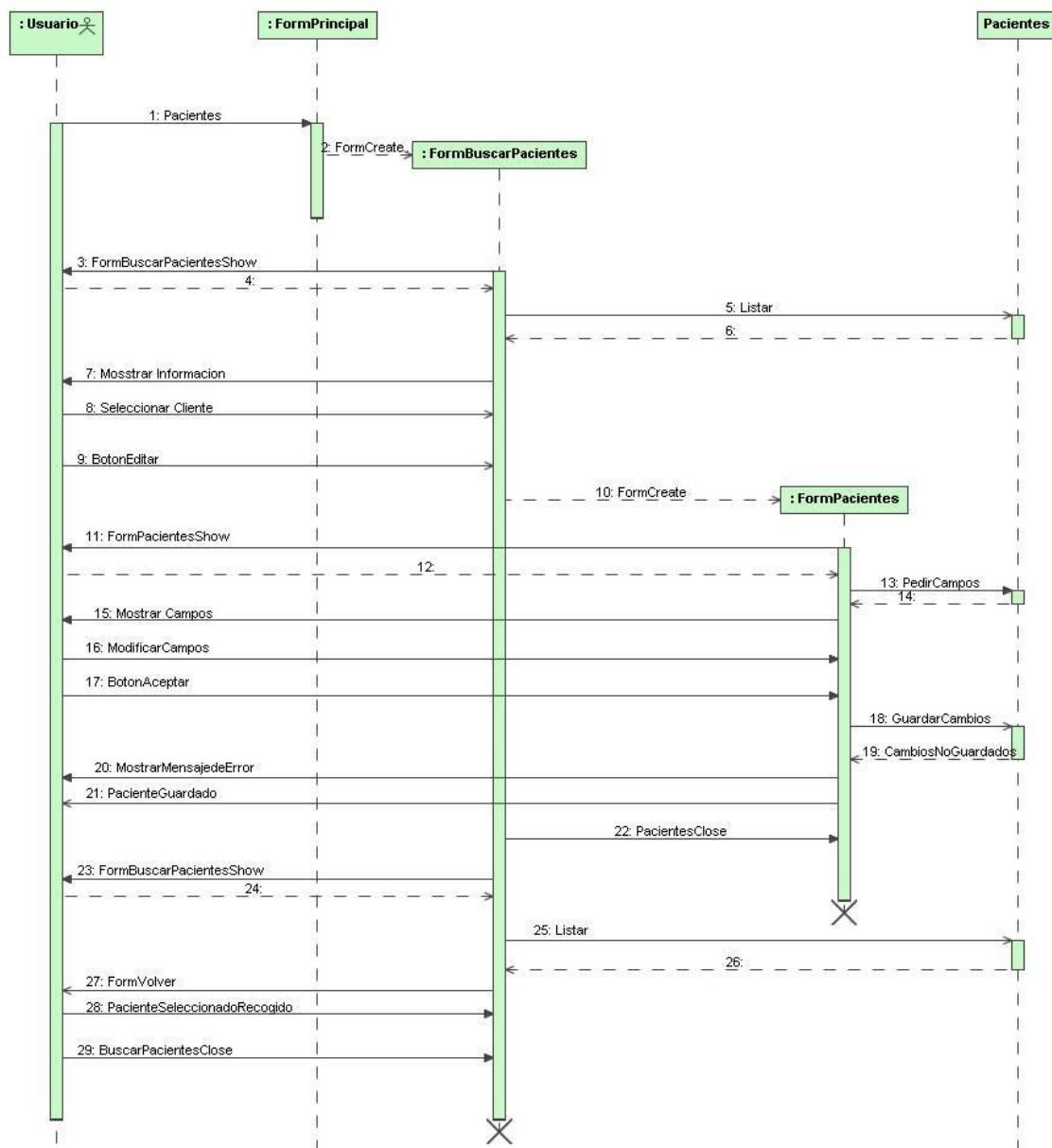


Fig36Diagrama de secuencia de Paciente Editar Excepción

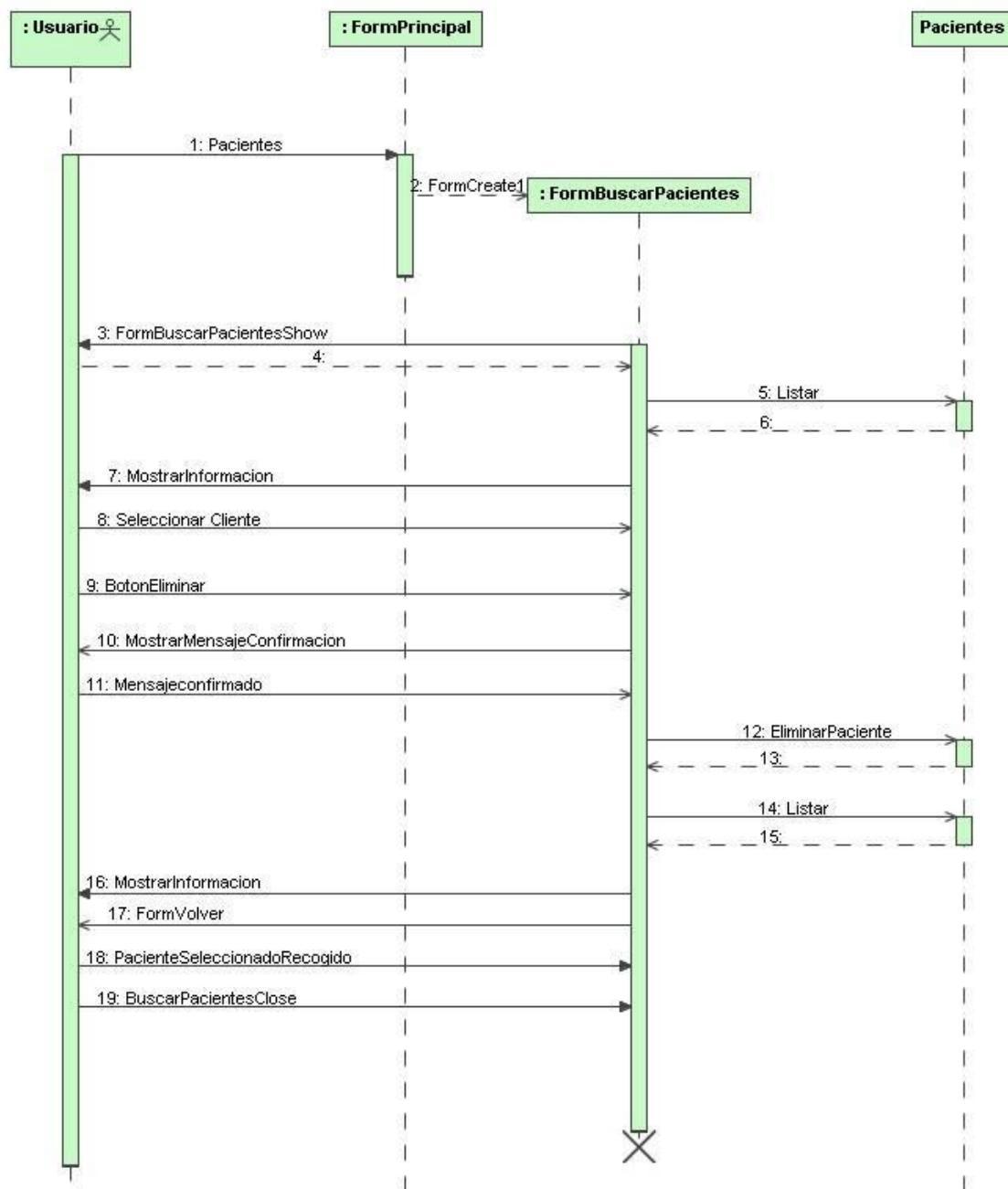


Fig37. Diagrama de secuencia de Paciente Eliminar

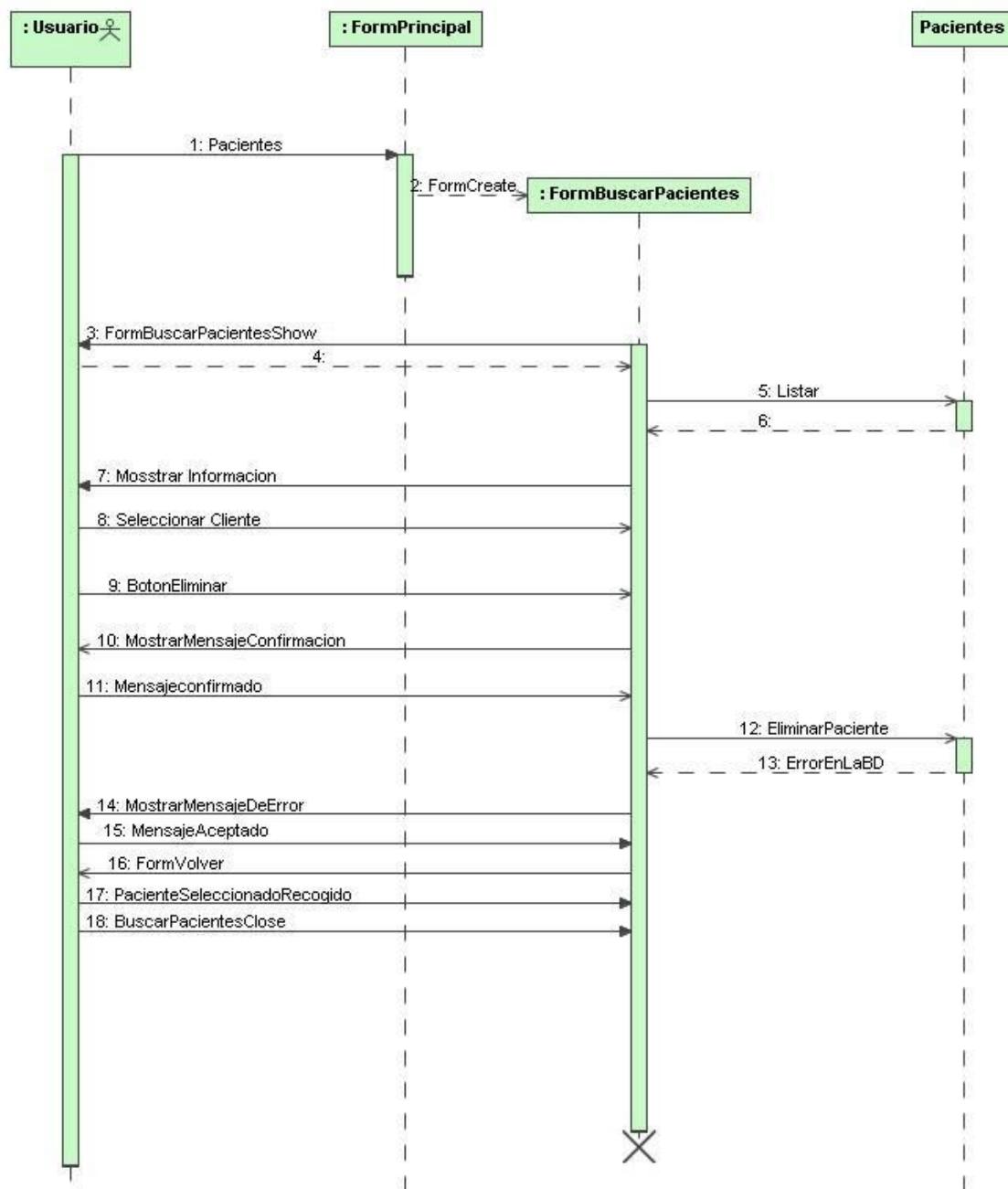


Fig38.Diagrama de secuencia de Paciente eliminar Excepción

Software para la gestión de una consulta de mensajes terapéuticos

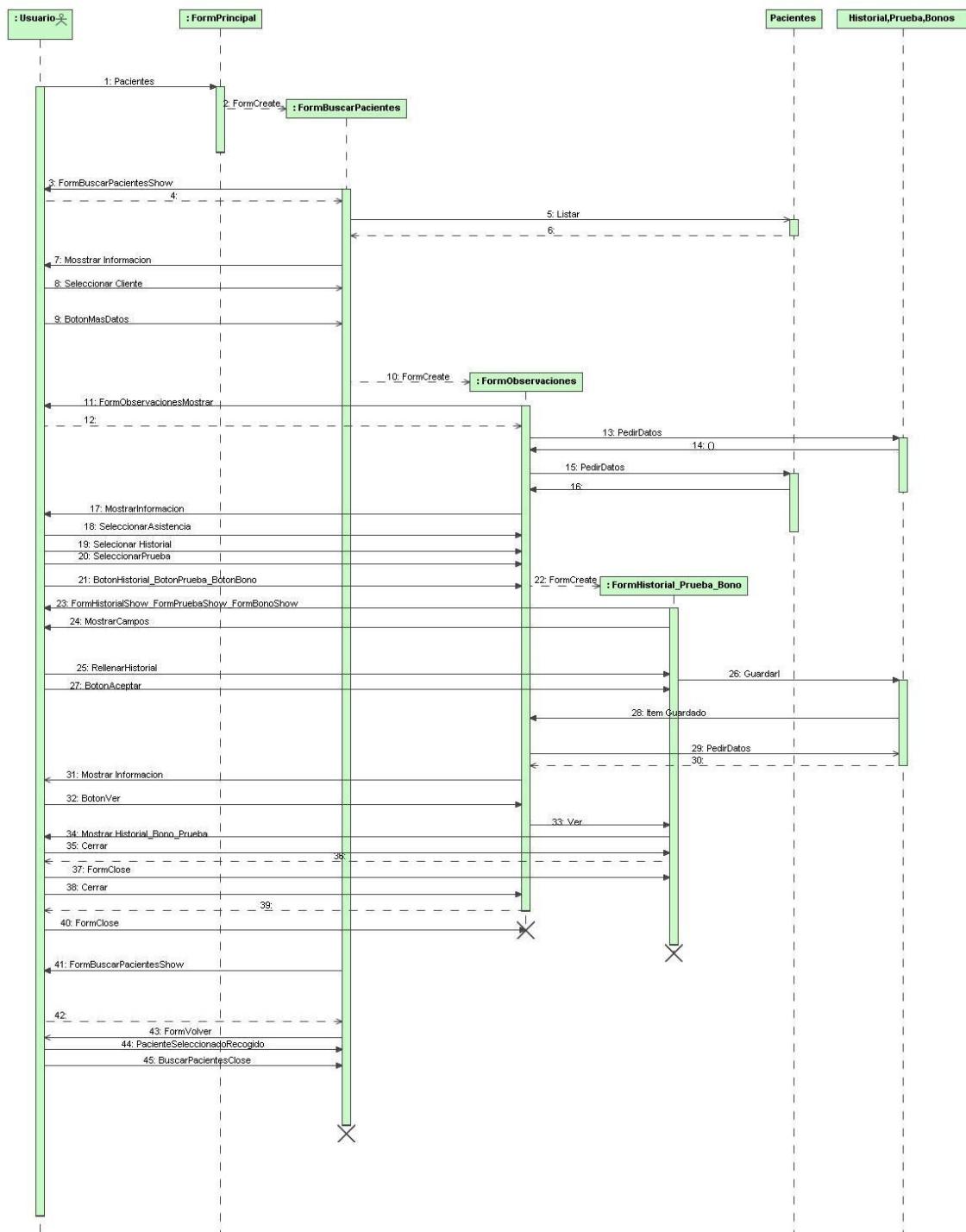


Fig40. Diagrama de secuencia de Paciente Mas Datos

Software para la gestión de una consulta de mensajes terapéuticos

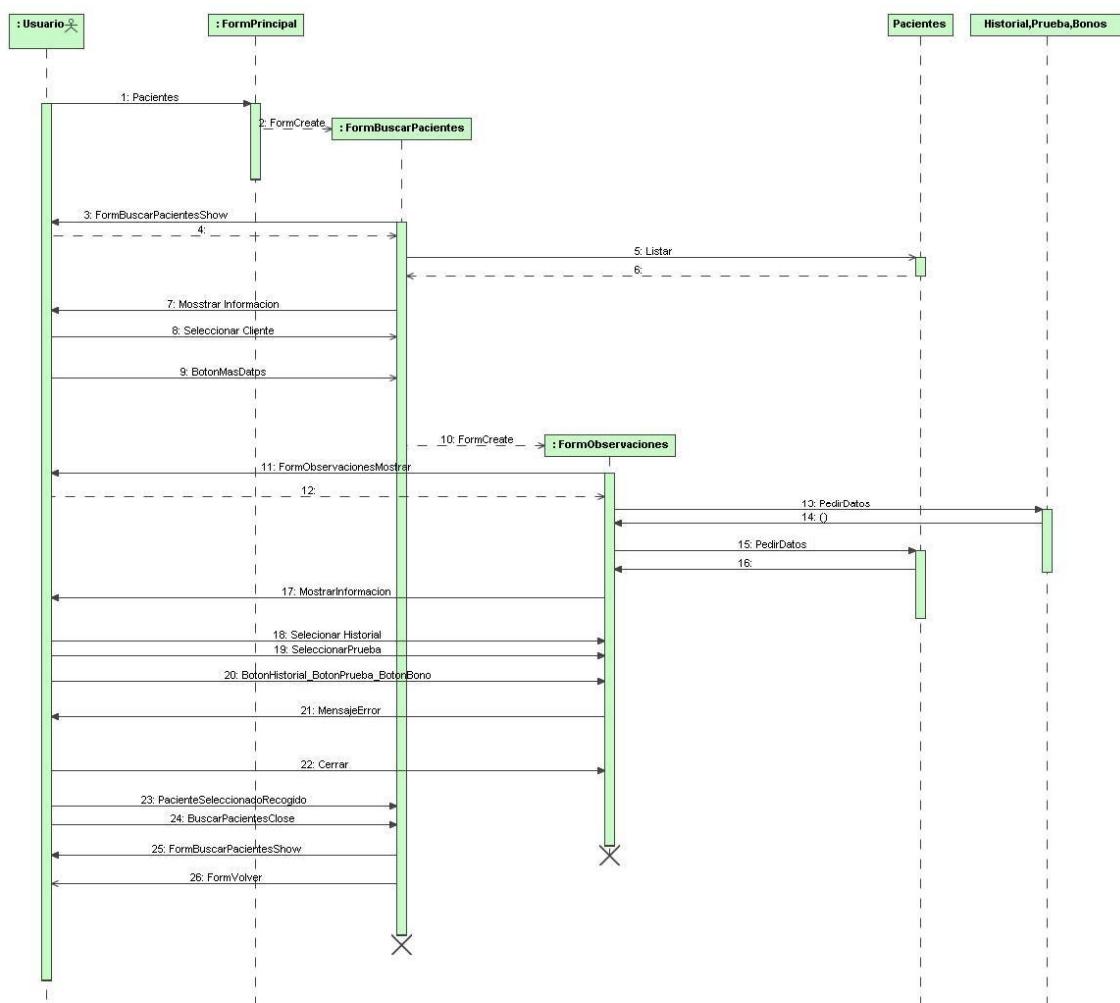
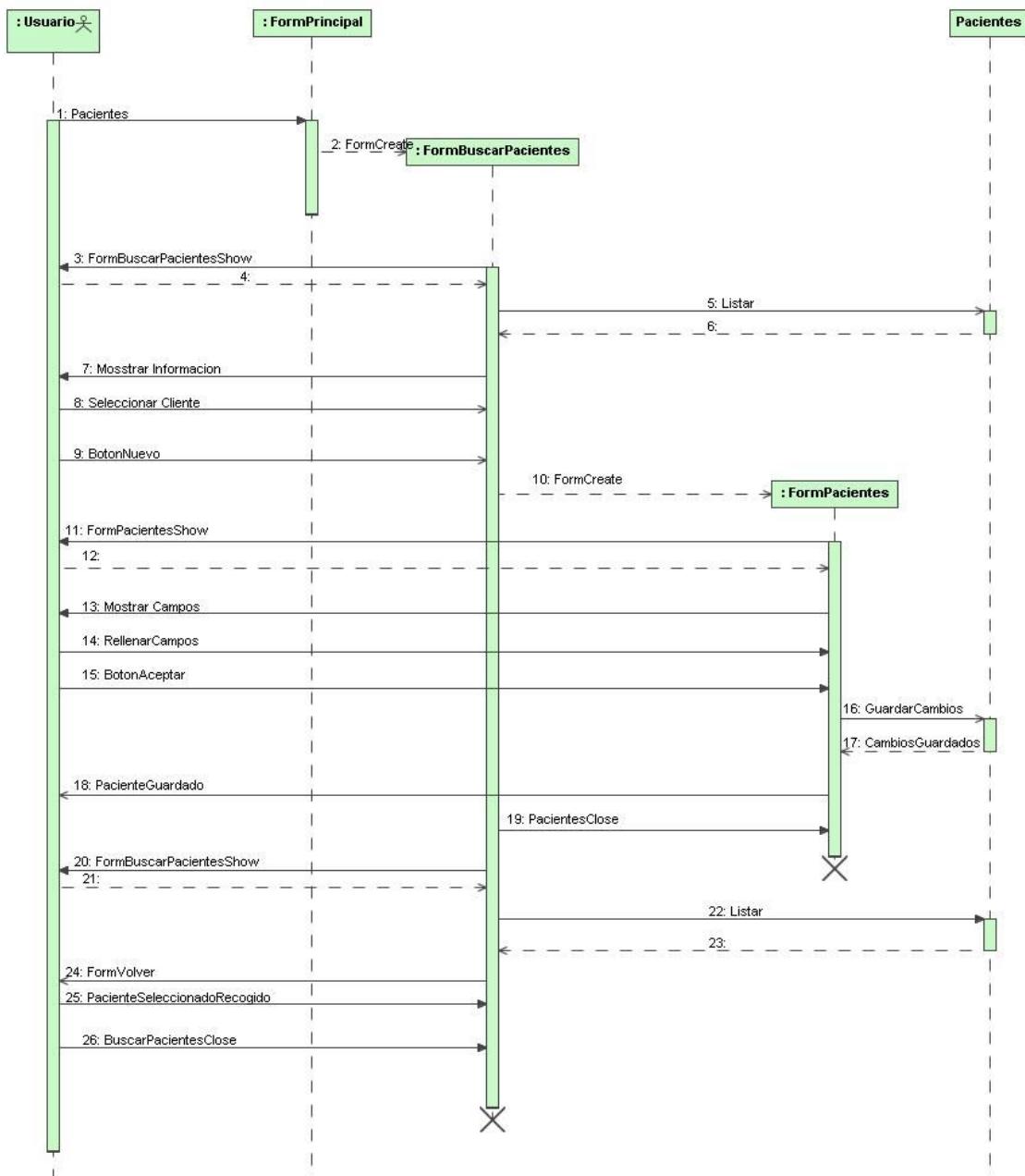


Fig41. Diagrama de secuencia de Paciente Mas Datos Excepcion



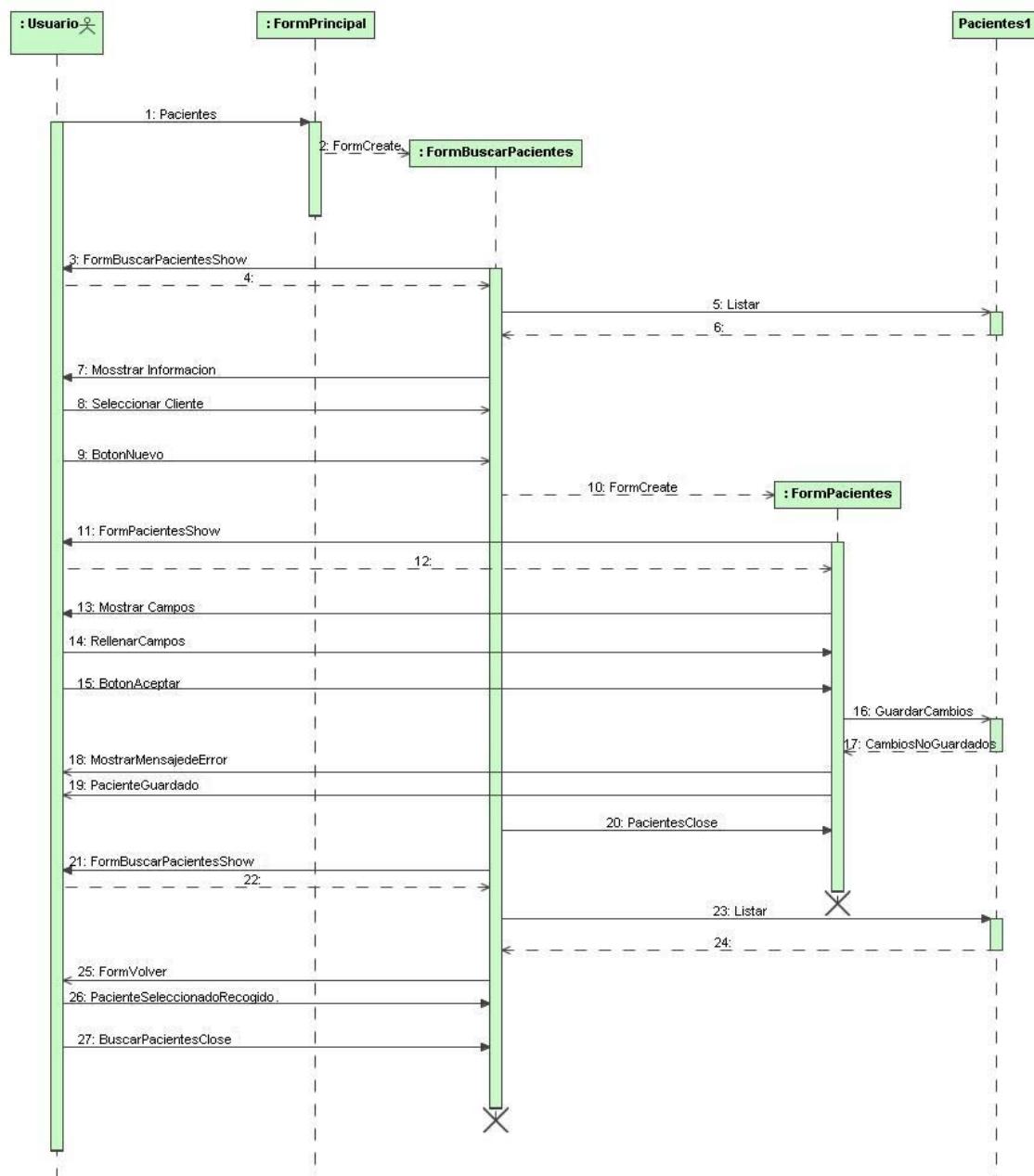


Fig43.Diagrama de secuencia de Paciente Nuevo Excepción

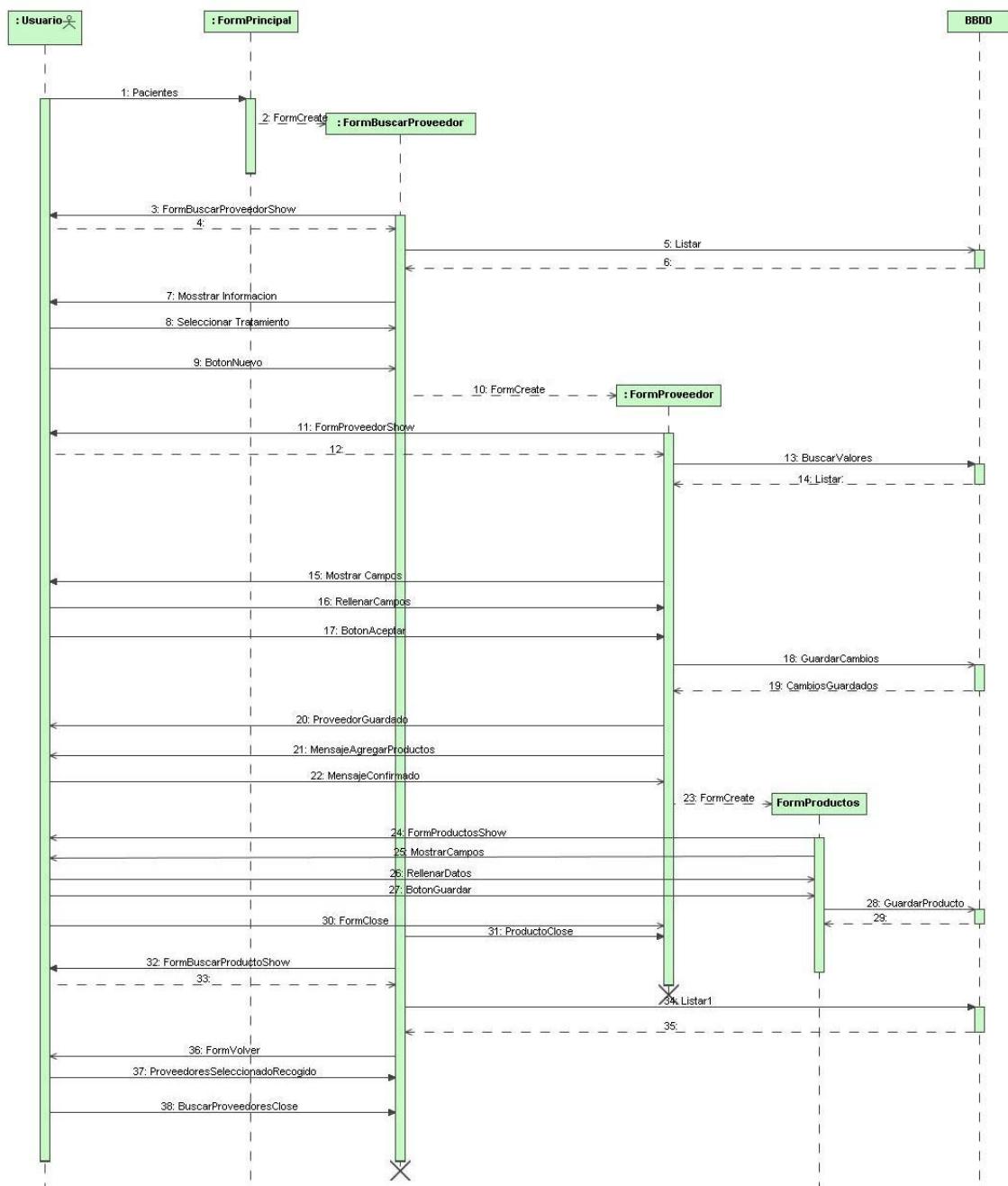


Fig44. Diagrama de secuencia de Proveedor Nuevo

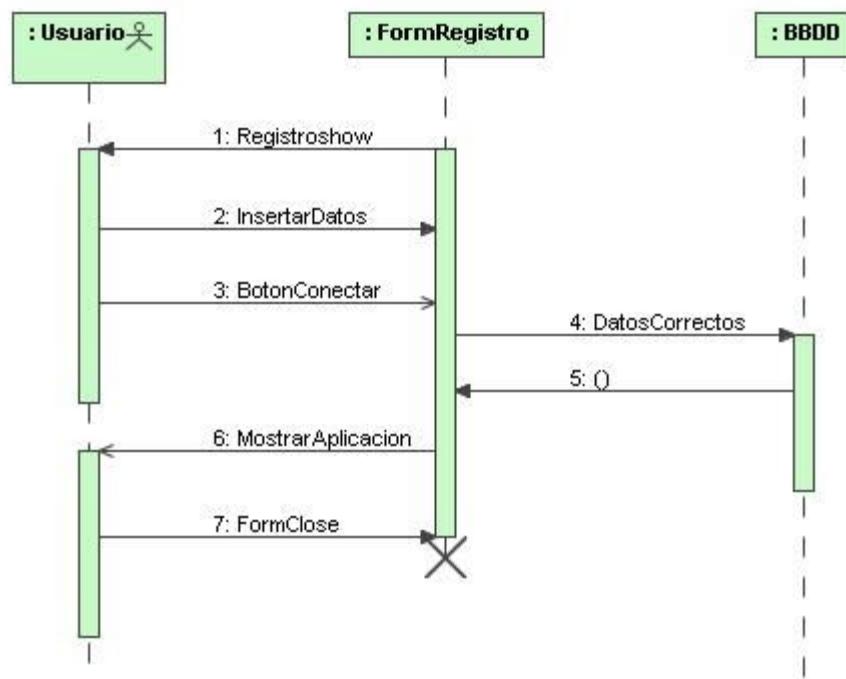


Fig45.Diagrama de secuencia de Registro

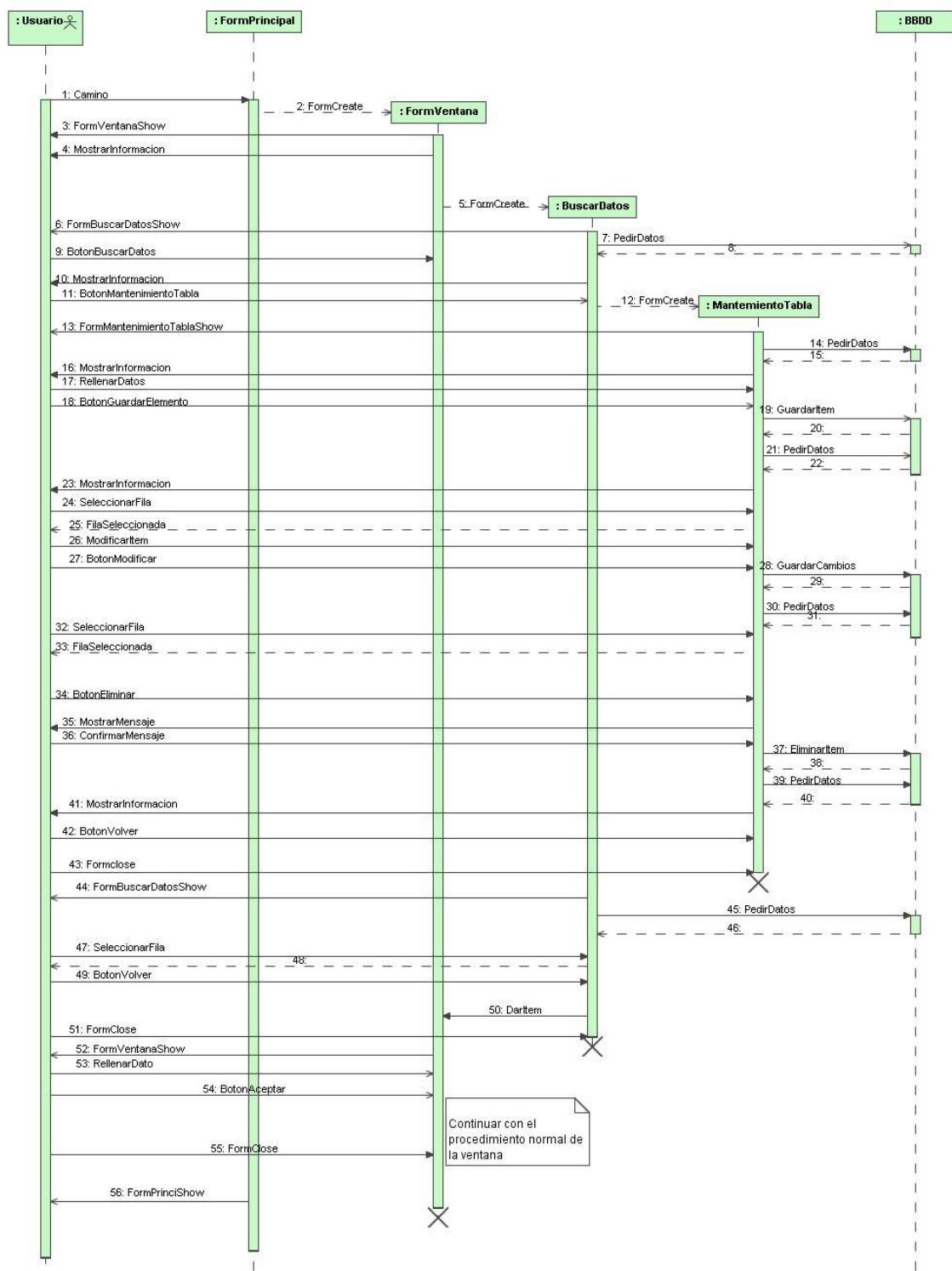


Fig46.Diagrama de secuencia de Tabla

Software para la gestión de una consulta de mensajes terapéuticos

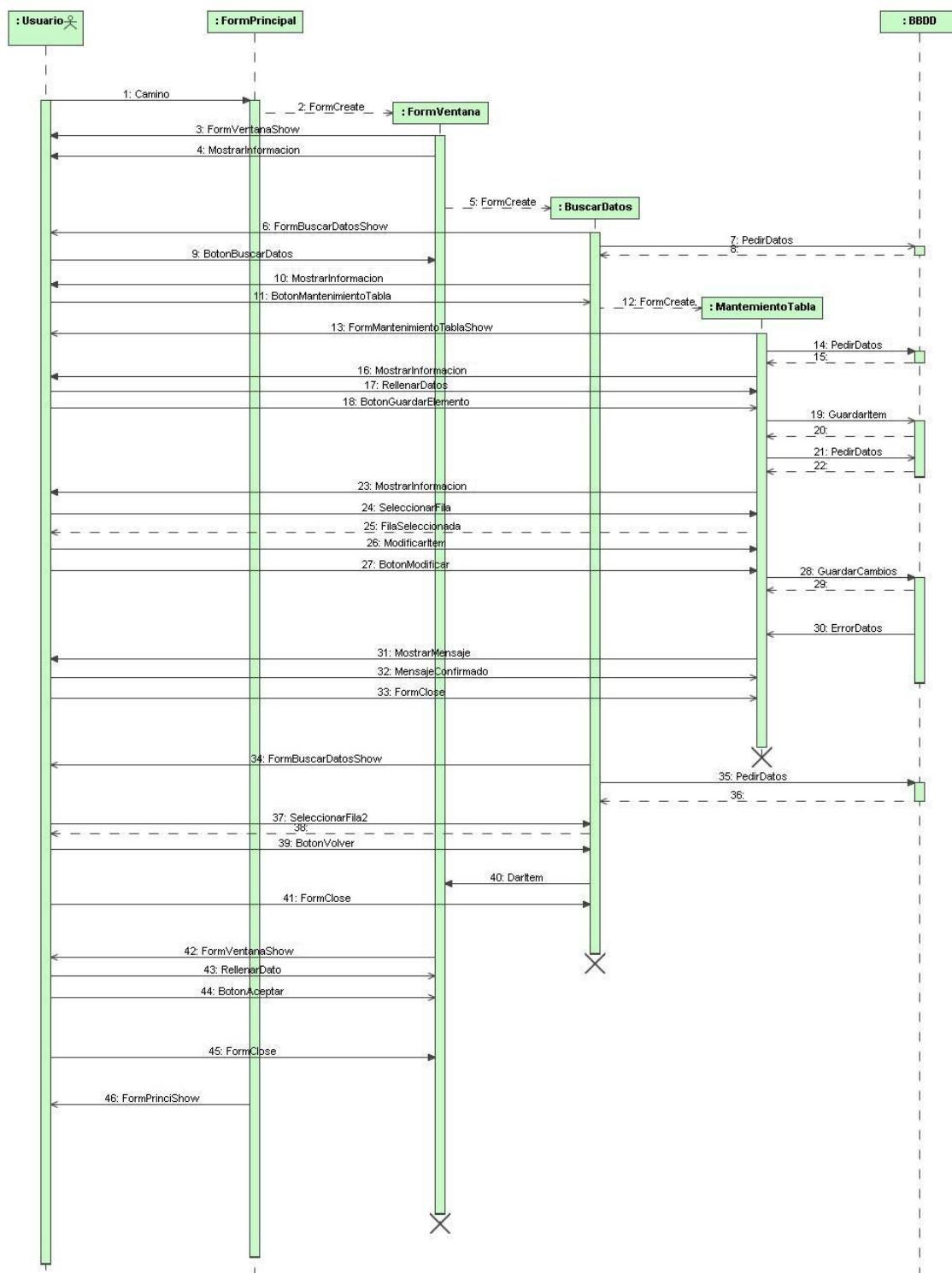


Fig47.Diagrama de secuencia de Tabla Excepción

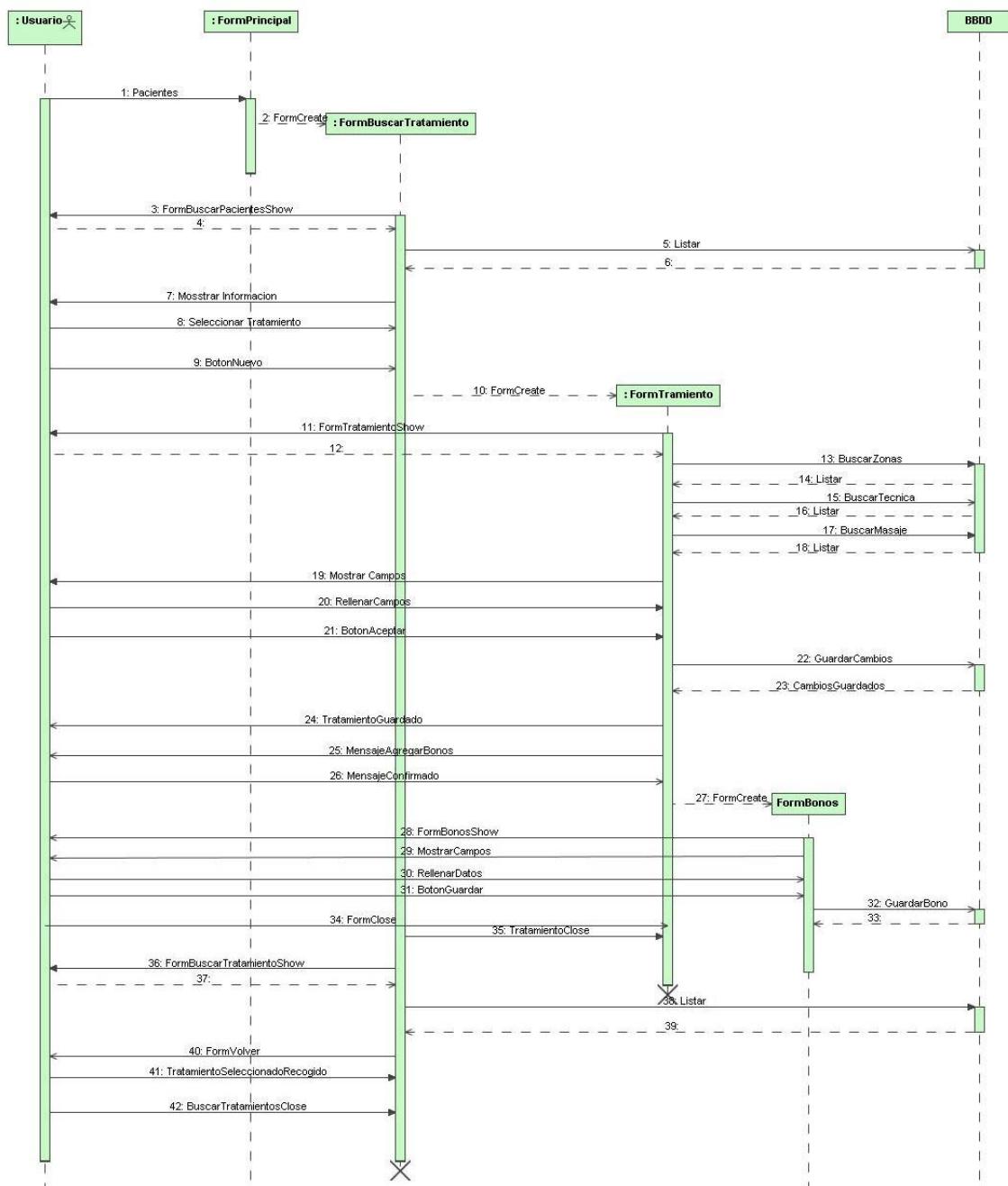


Fig48.Diagrama de secuencia de Tratamiento Nuevo

3.5: Diagrama de Transición de Estados del Proyecto

Definición:

En los diagramas de estado se observan los distintos estados que tiene un objeto durante su vida, y los estímulos que originan los cambios del estado en un objeto.

En estos diagramas se ven a los objetos como máquinas de estado o autómatas finitos, que pueden estar dentro de un conjunto de estados finitos y que pueden alterar su estado a través de un estímulo desde un conjunto finito de estados.

Estos diagramas están compuestos por estados, que sirven para identificar un periodo de tiempo en el que un objeto está esperando una operación. El estado está representado por un rectángulo con bordes redondeados con tres compartimientos, uno para el nombre, otro para el valor del atributo y otro para determinar las acciones. El diagrama también tiene eventos sirve para cambiar de estado. También se puede representar el cambio de estados a través de envío de mensajes, esto se representa por medio de una línea terminada en punta en el estado receptor.

Diagramas elaborados:

Al igual que con los diagramas de secuencia, en este apartado se ha elaborado los mismos diagramas correspondientes según la implementación deseada, agrupando los de tratamiento y proveedores con paciente y reuniendo en uno mismo (Tabla) los correspondientes a profesión, país, ubicación etc.

Software para la gestión de una consulta de masajes terapéuticos

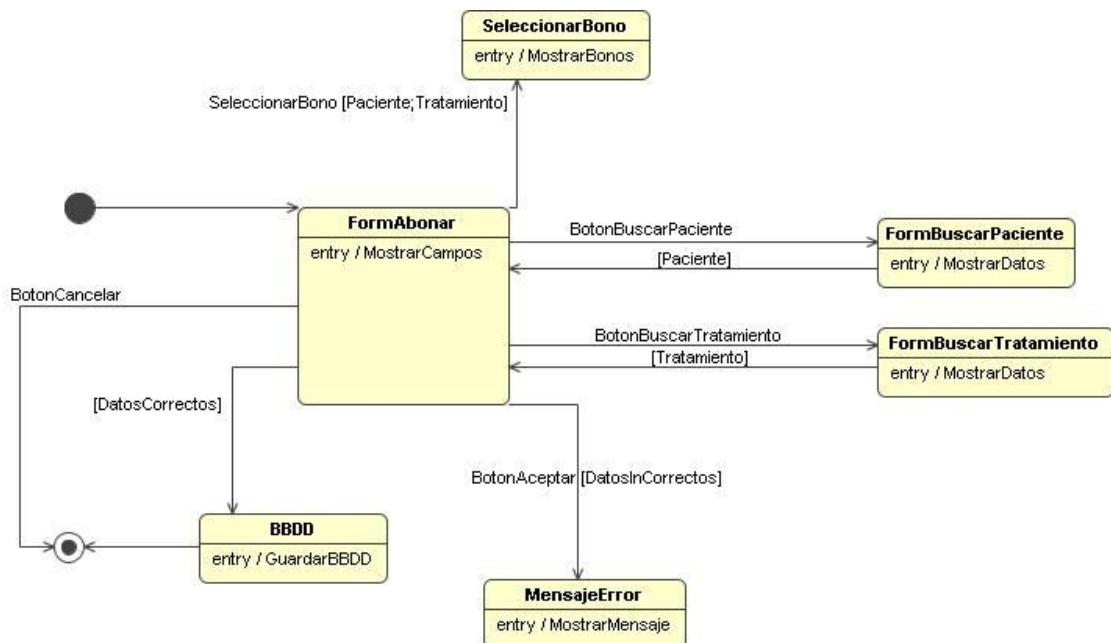


Fig49.Diagrama de transición de estados de Abonar

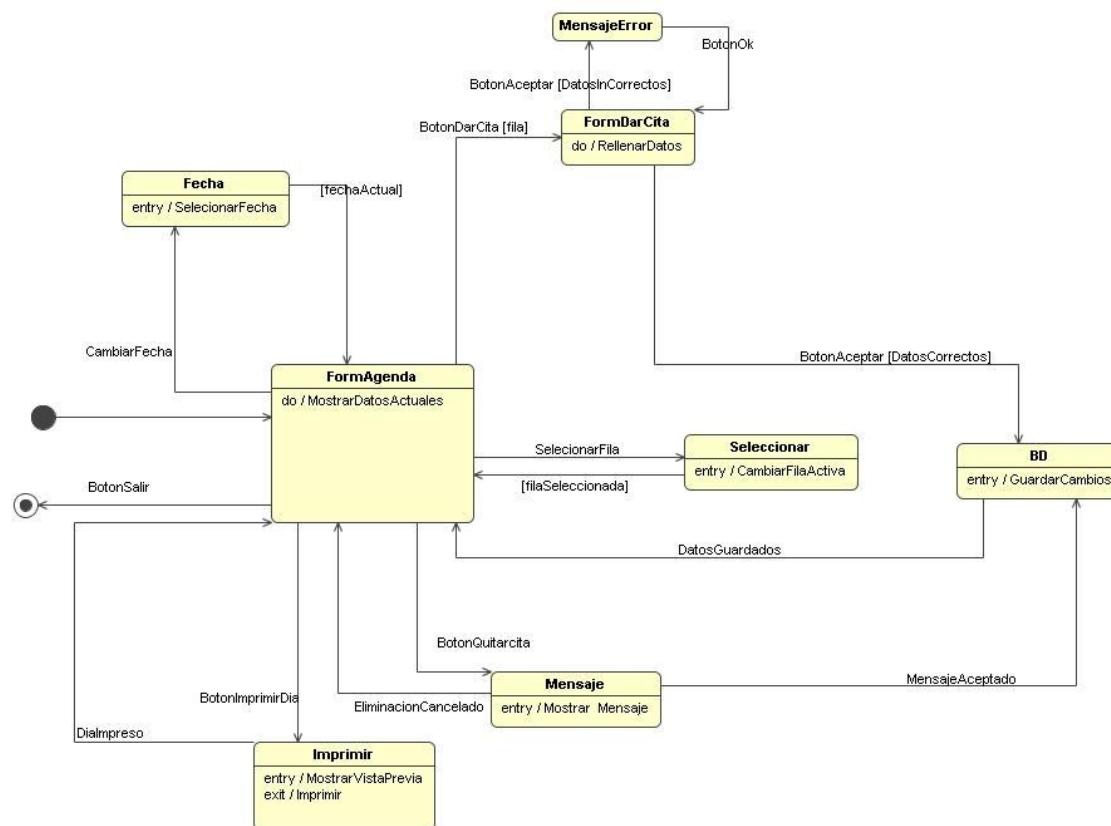


Fig50.Diagrama de transición de estados de Agenda

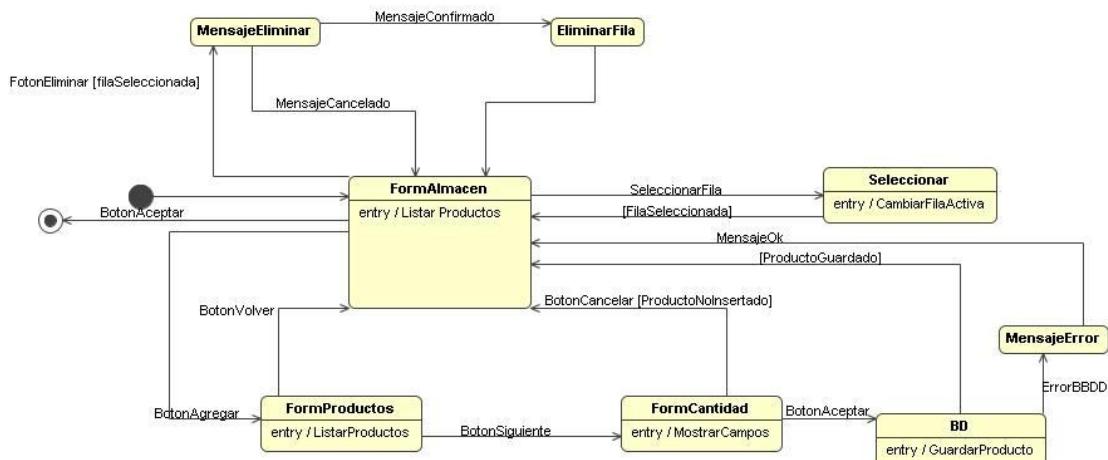


Fig51. Diagrama de transición de estados de Almacén

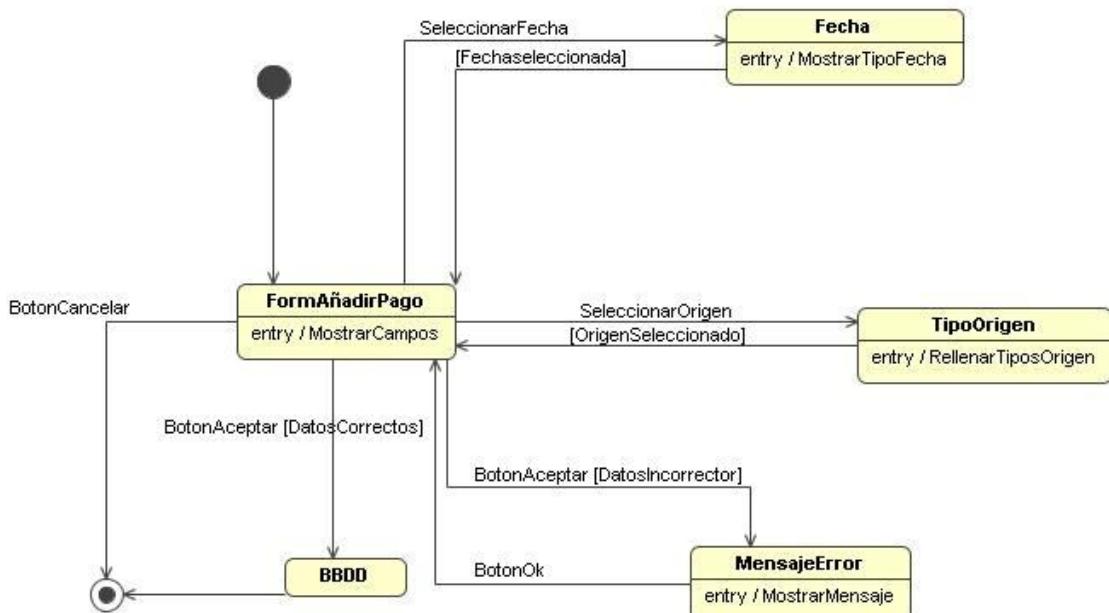


Fig52.Diagrama de transición de estados de Añadir Pago

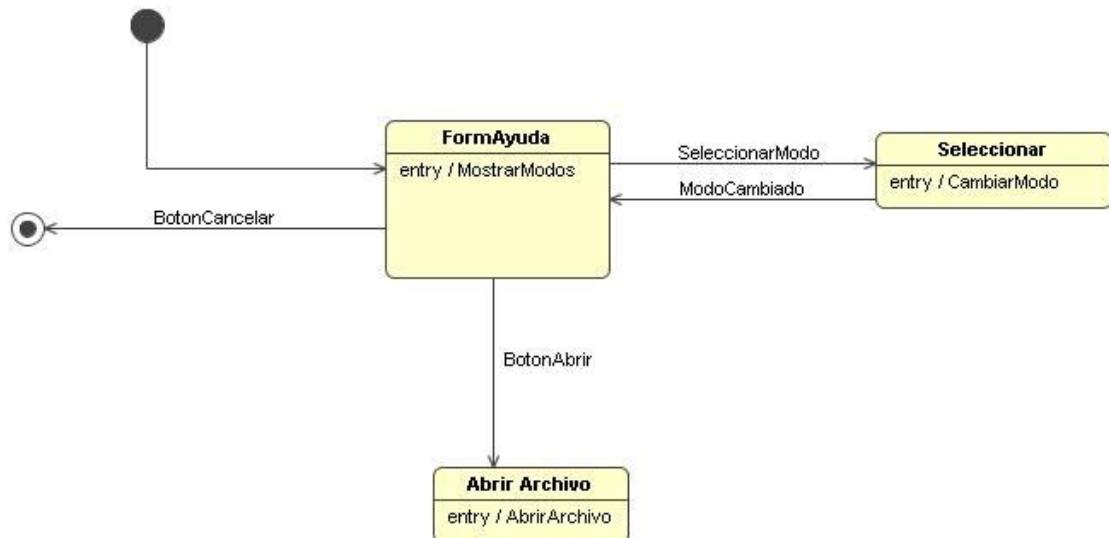


Fig53. Diagrama de transición de estados de Ayuda

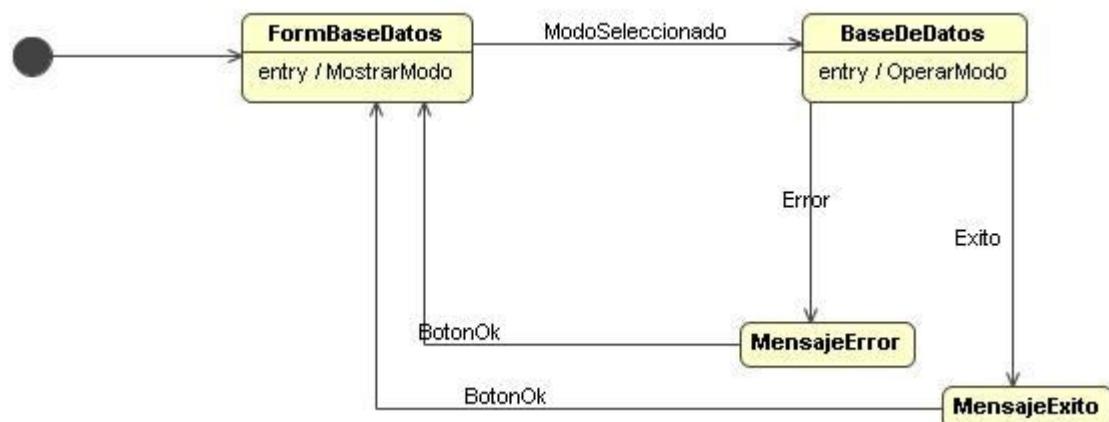


Fig54. Diagrama de transición de estados de Base De Datos

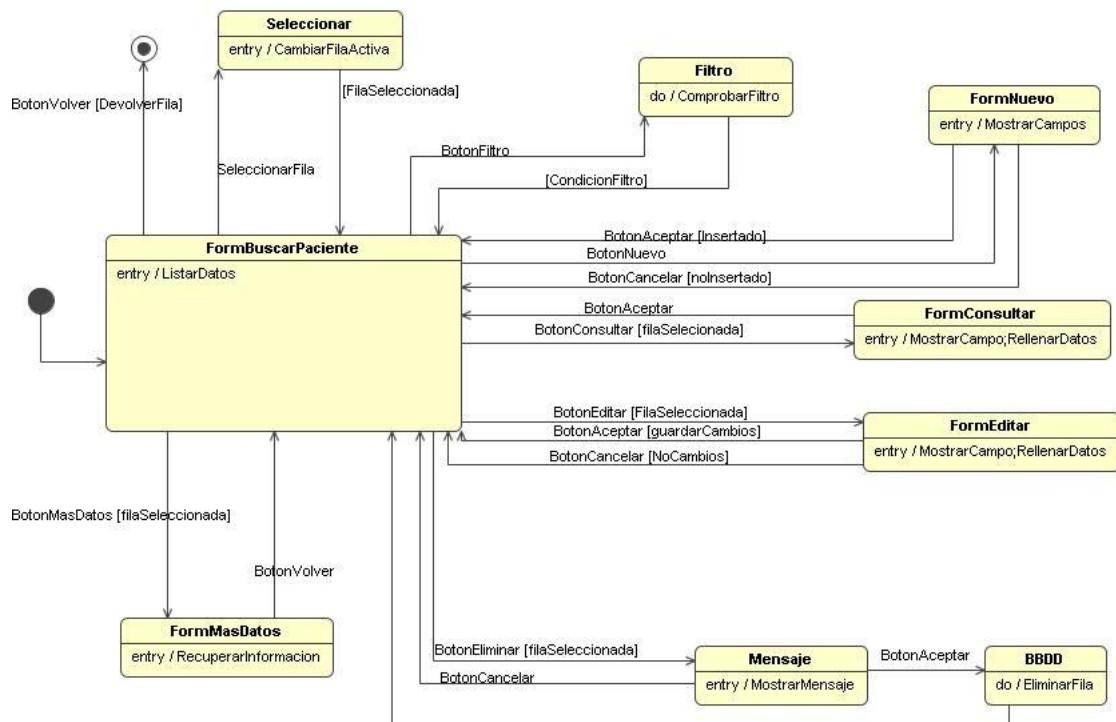


Fig55.Diagrama de transición de estados de Buscar Paciente

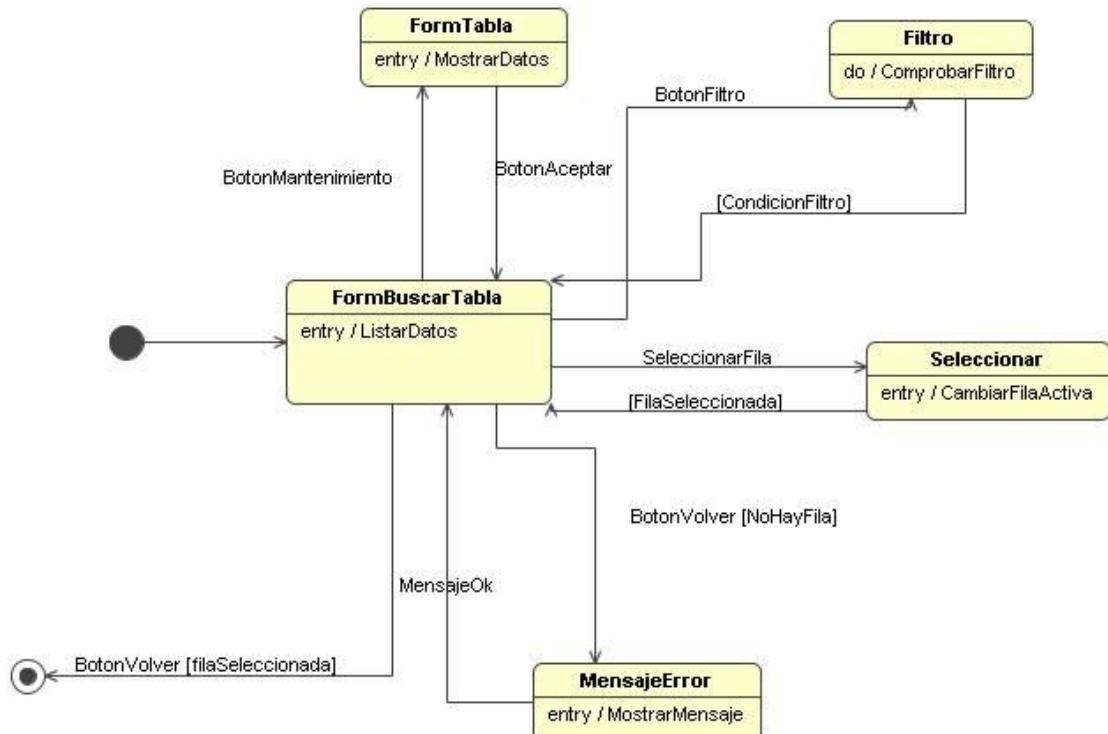


Fig56.Diagrama de transición de estados de Buscar Tabla

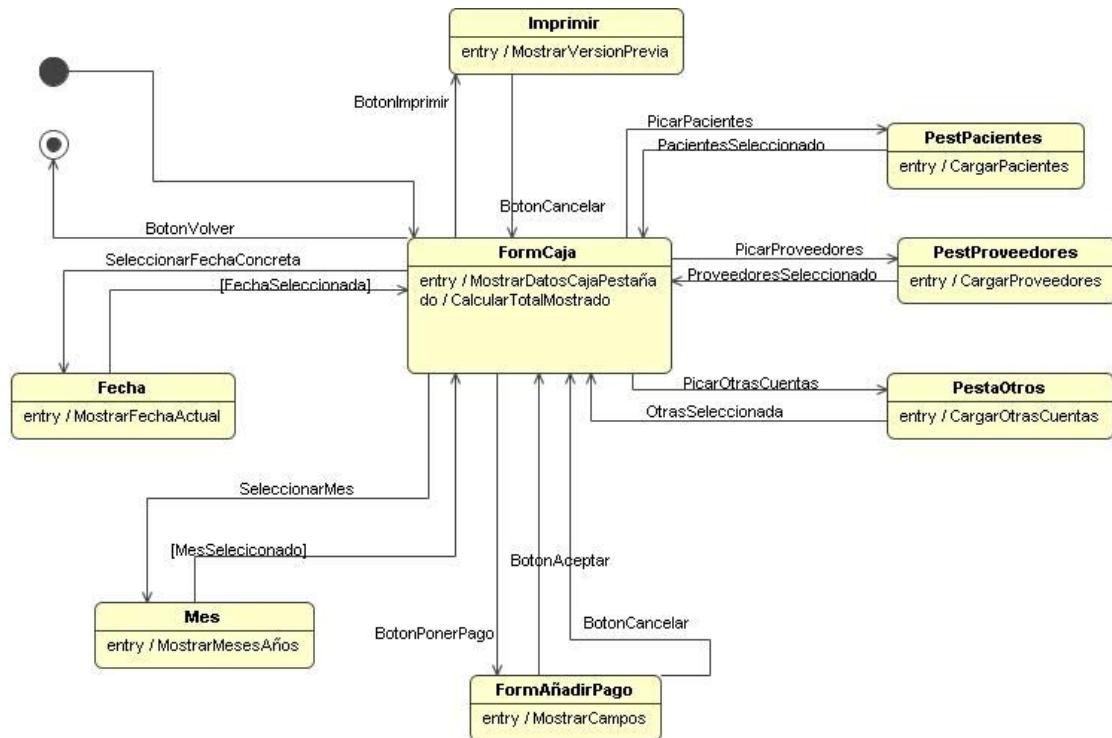


Fig57. Diagrama de transición de estados de Caja

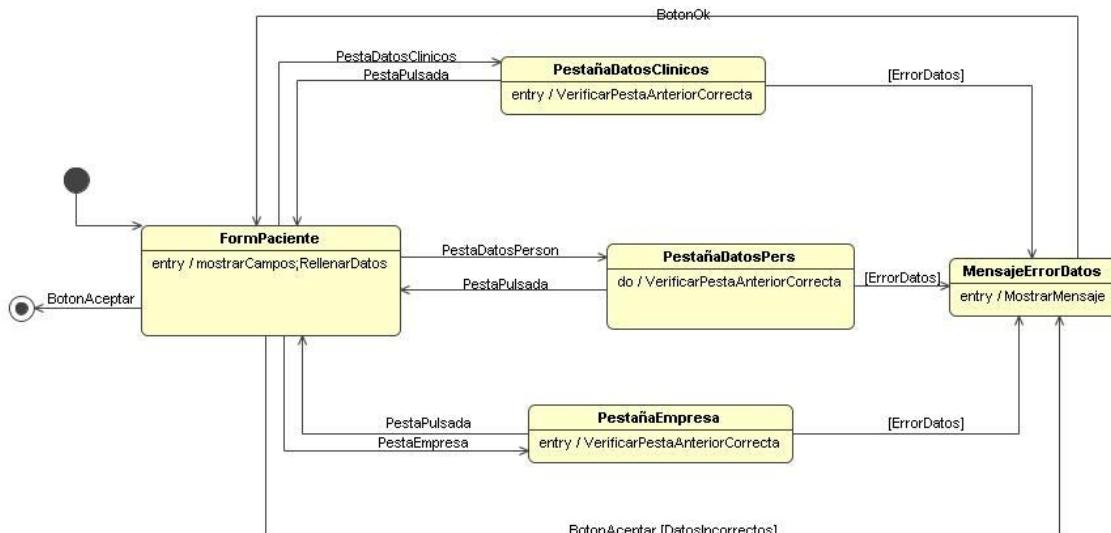


Fig58.Diagrama de transición de estados de Consultar Paciente

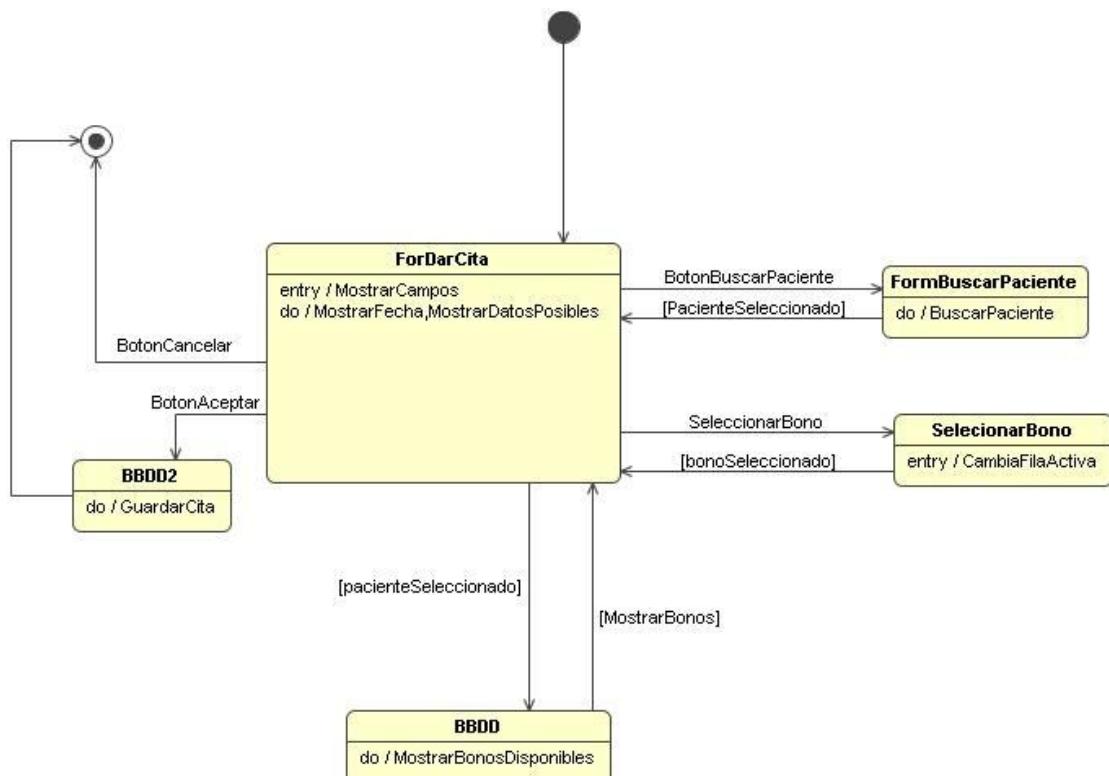


Fig59.Diagrama de transición de estados de Dar Cita

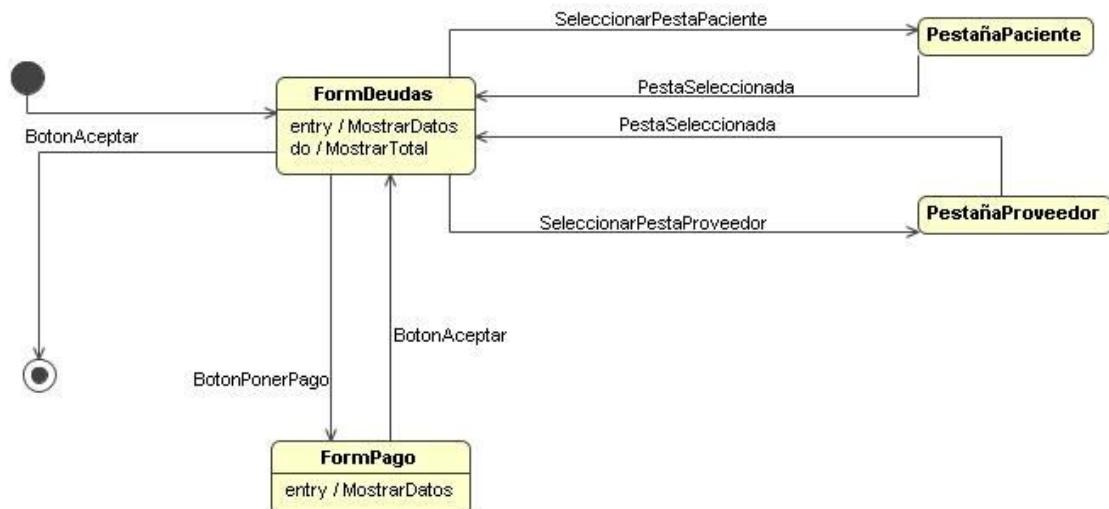


Fig60.Diagrama de transición de estados de Deudas

Software para la gestión de una consulta de masajes terapéuticos

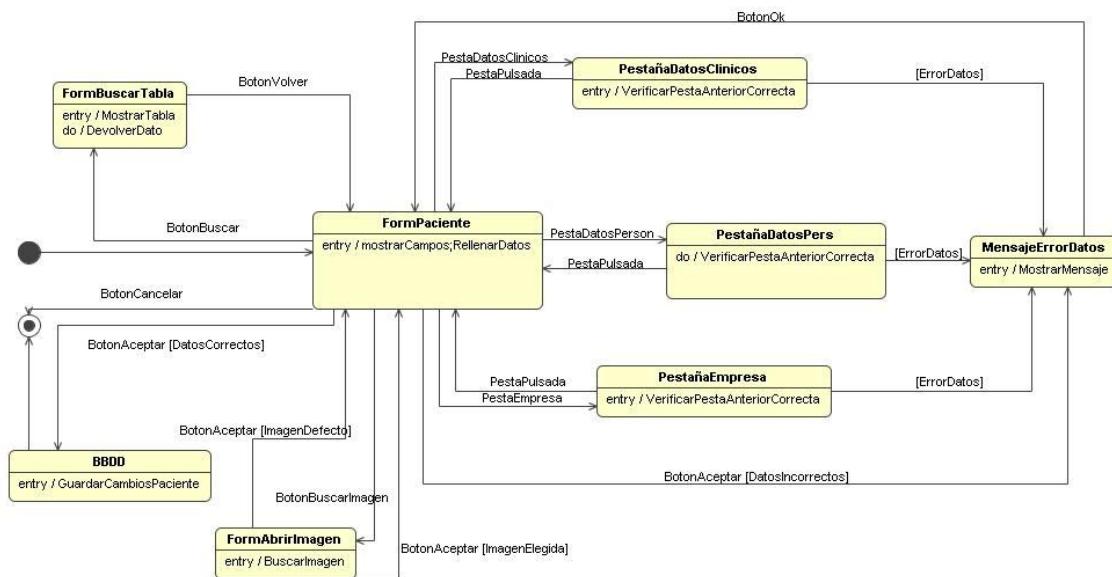


Fig61. Diagrama de transición de estados de Editar Paciente

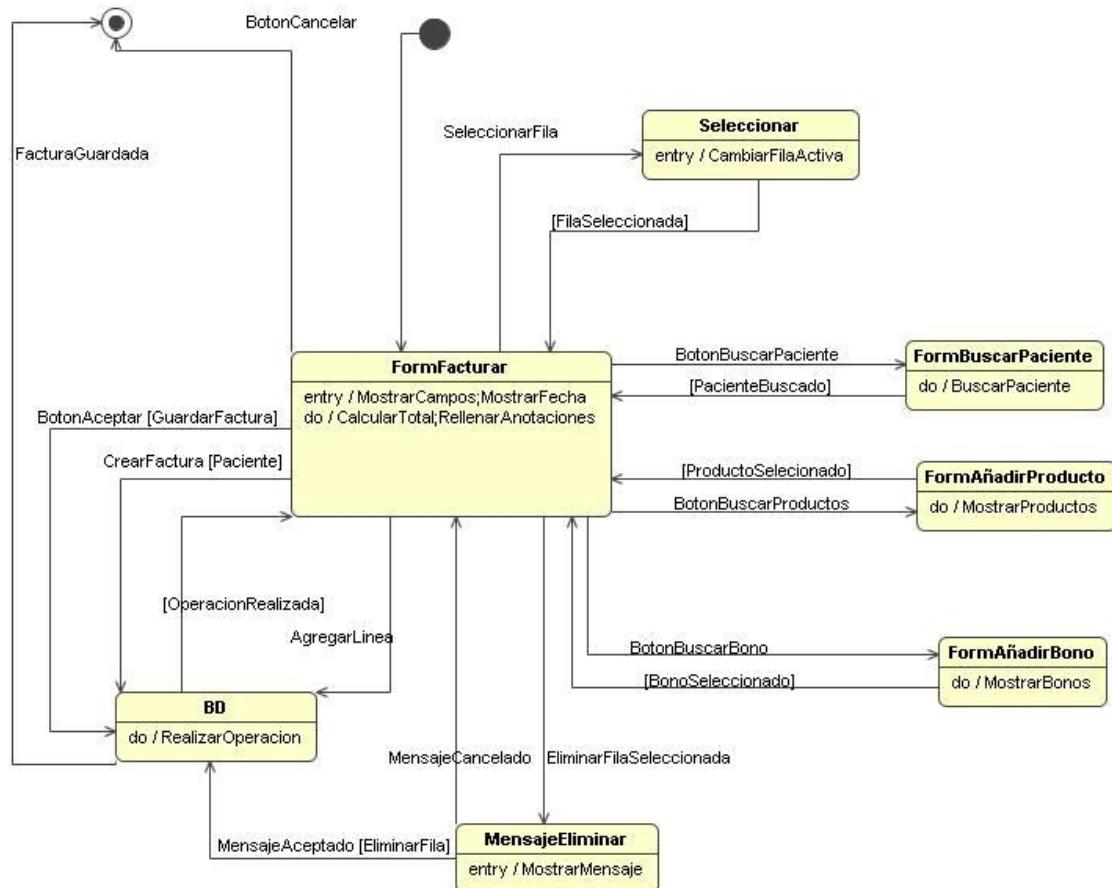


Fig62.Diagrama de transición de estados de Facturas

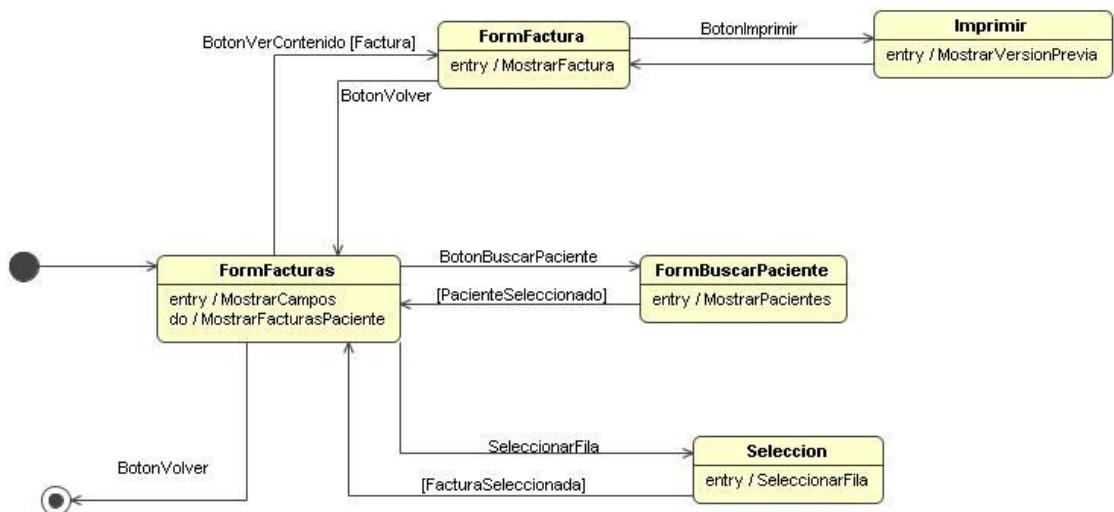


Fig63.Diagrama de transición de estados de Facturas Por Paciente

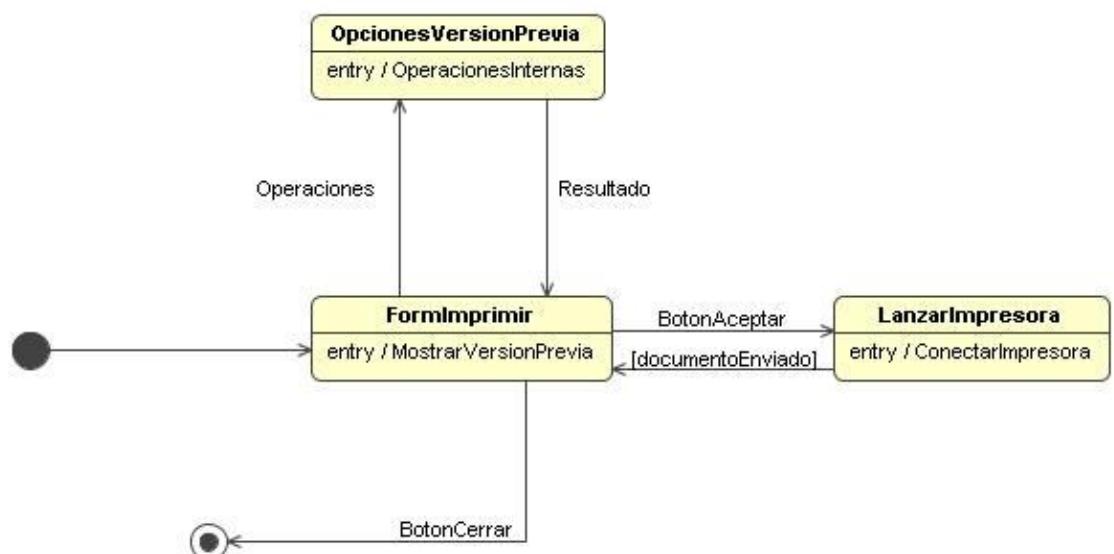


Fig64. Diagrama de transición de estados de Imprimir

Software para la gestión de una consulta de masajes terapéuticos

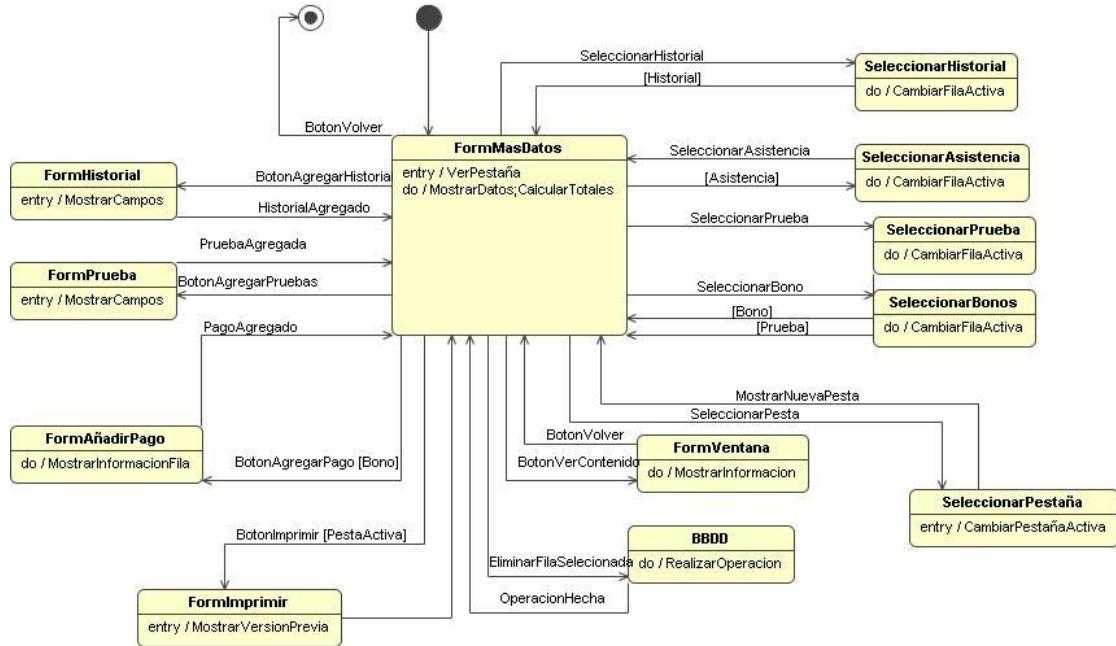


Fig65.Diagrama de transición de estados de Mas Datos

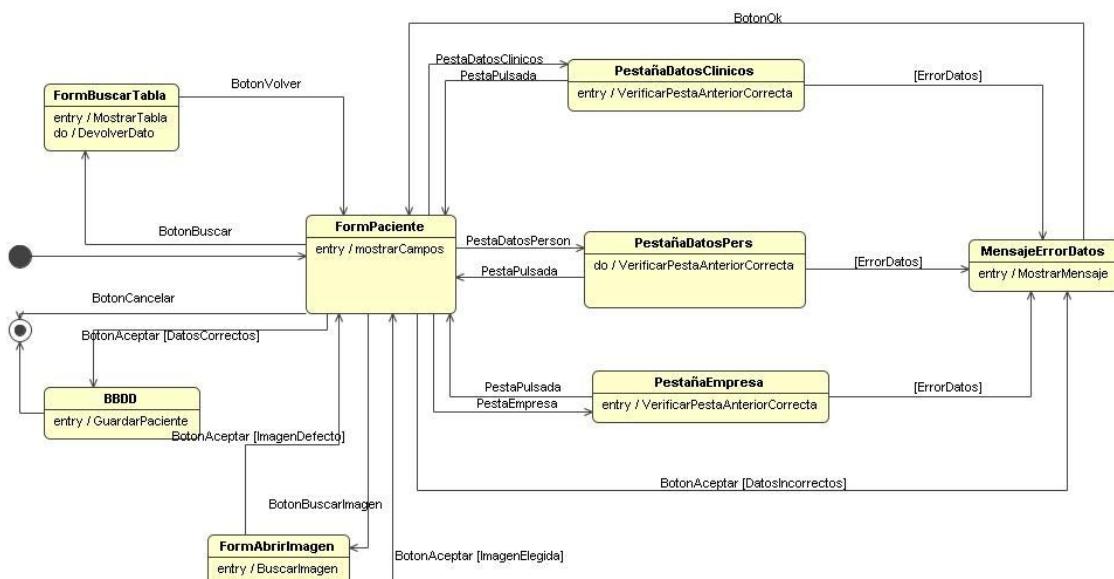


Fig66.Diagrama de transición de estados de Nuevo Paciente

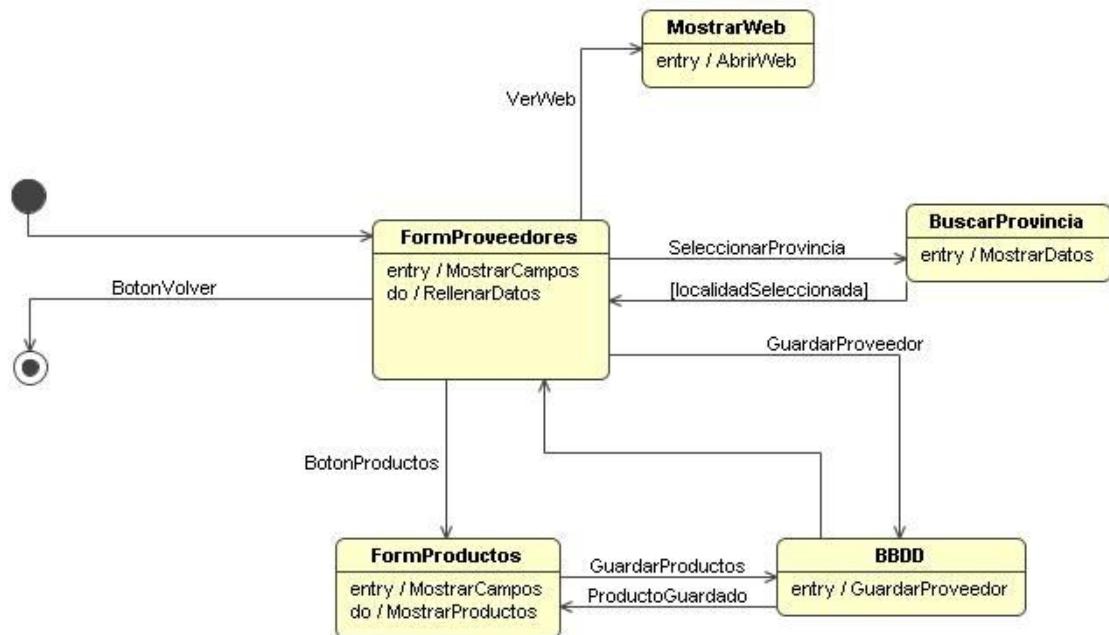


Fig67.Diagrama de transición de estados de Proveedores

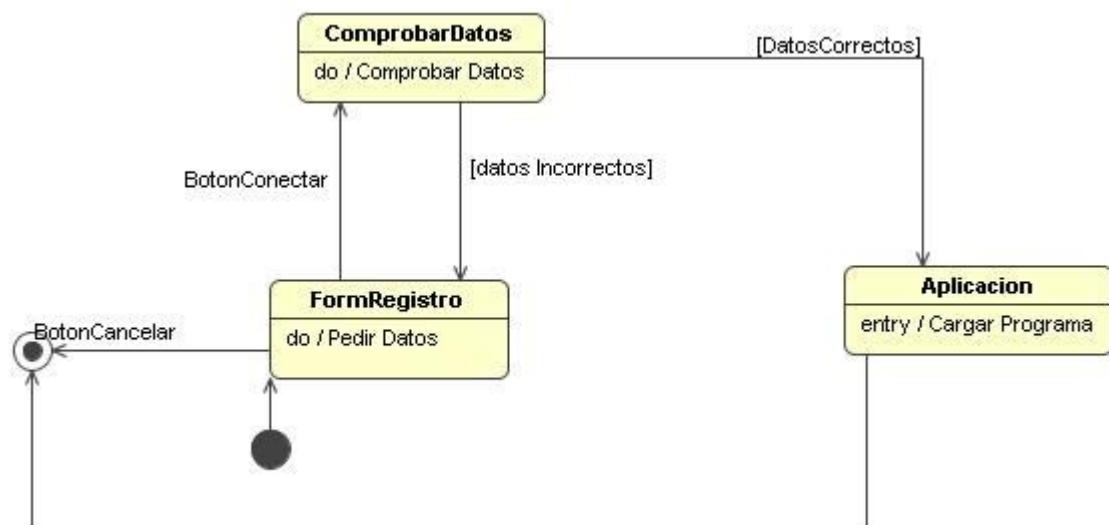


Fig68. Diagrama de transición de estados de Registro

Software para la gestión de una consulta de masajes terapéuticos

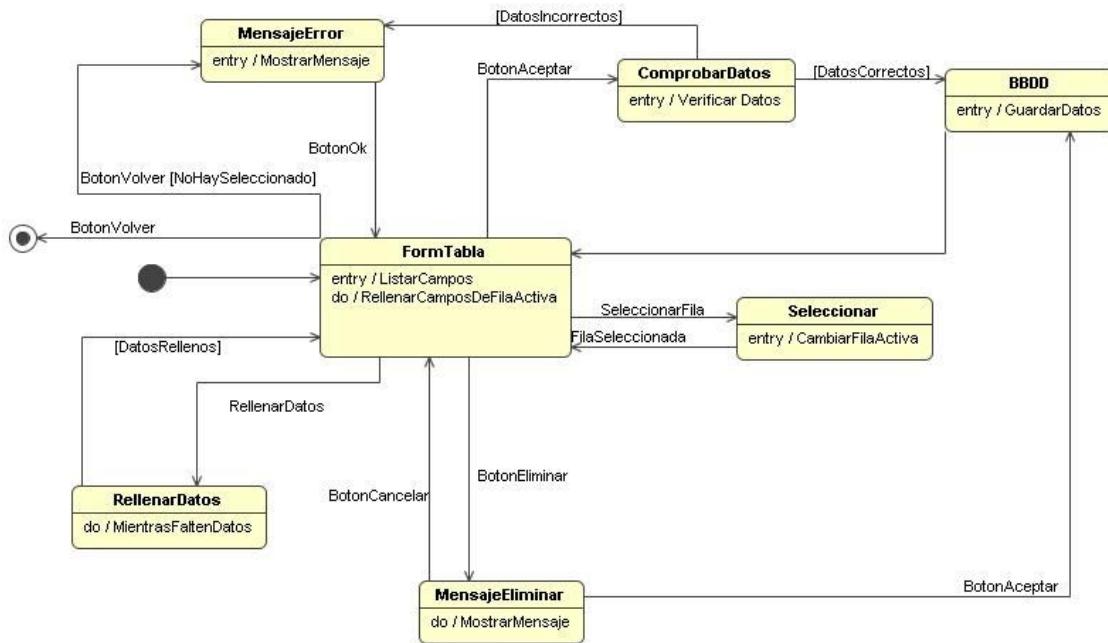


Fig69. Diagrama de transición de estados de Tabla

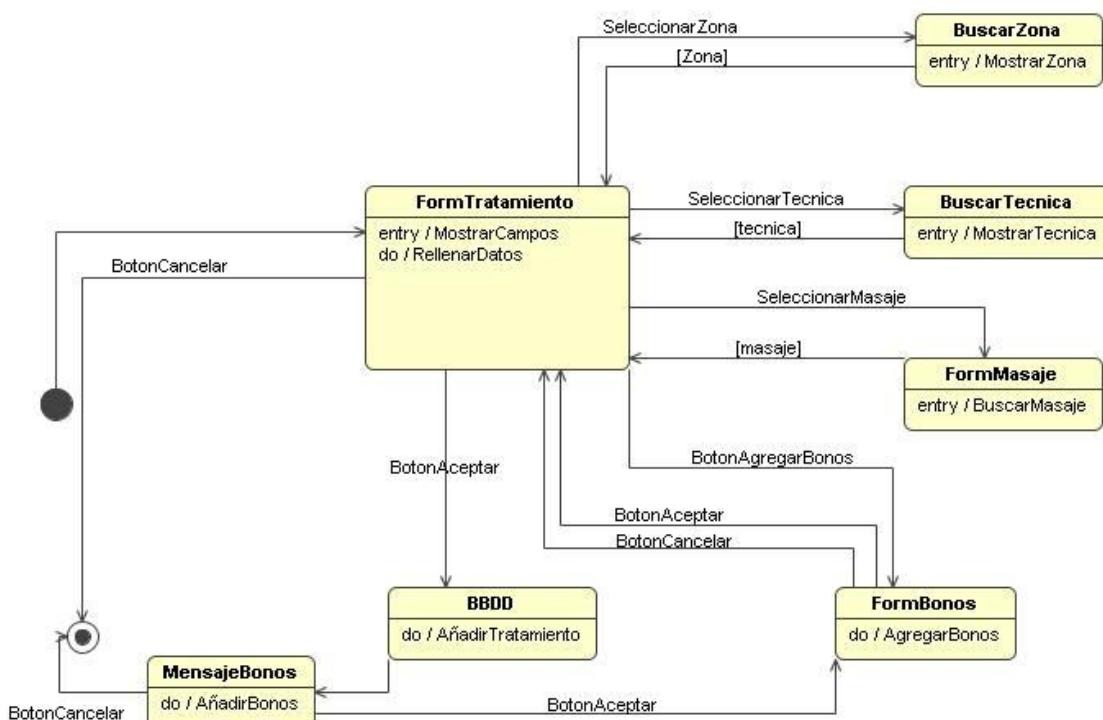


Fig70.Diagrama de transición de estados de Tratamiento

Capítulo 4: Herramientas Utilizadas

En este capítulo vamos a tratar las herramientas utilizadas durante el desarrollo del proyecto. Entre estas herramientas se encuentra MySql que es un gestor para la base de datos, C# para la implementación de la aplicación y Aqua Data Studio que nos a permitido interactuar con la base de datos mediante una interfaz más agradable que la propia consola de MySql.

4.1 MySQL

4.1.1: Introducción.

MySql es por excelencia uno de los gestores de bases de datos Open Source que más usuarios utilizan para sus aplicaciones.

Con MySql podemos almacenar los datos mediante una colección estructurada. En esta colección podemos almacenar desde una simple lista de datos como los nombres de personas o perfectamente una enorme cantidad información de una empresa.

En MySql disponemos de un gestor de base de datos relacional. Una base de datos relacional distribuye el almacenamiento de los datos en diferentes tipos de tablas. Esto permite velocidad y flexibilidad.

MySql es un lenguaje de consulta estructurado SQL. SQL es el lenguaje estándar más común para acceder a la base de datos.

En un principio MySql empezó de un modo trivial sobre las páginas Web, pero actualmente se ha convertido como una solución segura.

El motivo de que MySql sea utilizado por la inmensa mayoría de los usuarios se debe a que MySql es Open Source, es decir, que es posible para cualquier usuario usar y

modificar el software. Además MySQL es gratuito y admite todo tipo de ideas y sugerencias para expandir aún más su potencialidad.

MySQL proporciona todos los elementos para instalar la gestión de base de datos, preparar distintos niveles de acceso a usuario, proteger los datos y administrar el sistema.

4.1.2: Características

A continuación vamos a especificar ciertas características que más destacan sobre MySQL.

- **Características internas**

Está realizado mediante el lenguaje C y C++ pero está probado en diferentes tipos de compiladores diferentes. De hecho podemos trabajar con MySQL en C, C++, Eiffle, Java, Perl, PHP etc. Y para completar la potencia de utilización podemos aplicarlo en cualquier tipo de plataforma. Dispone de opciones de ejecución sobre hebras para su utilización en múltiple CPU.

- **Características técnicas**

Se proporciona sistemas de almacenamientos transaccionales y no transaccionales. Usa tablas MyISAM en las que recae la importancia sobre la consulta select. Estas tablas se utilizan más en aplicaciones donde apenas se hacen consultas de inserción, modificación o eliminación como las páginas Web, por lo que son muy rápidas con compresión de índice.

Fácil de añadir otro sistema de almacenamiento. Esto es útil si se desea añadir una interfaz SQL para una base de datos propia. Además se amplia esta característica pudiendo importar y exportar en distintas plataformas y realizar volcados de información online.

Posee ‘Joins’ muy rápidos usando multi-join, tablas-hash en memoria, que son usadas como tablas temporales.

Todas las funciones están preparadas para las exigencias mas altas de los programadores, están implementadas usando una librería altamente optimizada y normalmente no se realiza reserva de memoria tras la inicialización de consultas.

- Tipos de columnas. MySql dispone de tres tipos de columnas que son:
 - Numérica. En estas columnas se puede almacenar precios, edades o cantidades. MySql proporciona dos tipos de numéricos que son el entero y punto flotante.
 - Cadena. Los tipos de cadena se utilizan para almacenar tipos compuestos de caracteres.
 - Fecha. En este tipo de columna podremos almacenar

- Tipos de tablas

Existen dos tipos de tablas de transacción segura (Innodb y BDB). El resto (ISAM; MyISAM, MERGE y HEAP) no son de transacción segura. La elección del tipo de tabla adecuado puede afectar enormemente al rendimiento.

Innodb

Estas tablas disponen de commit y de rollback por ser de transacción segura. A diferencia con las tablas MyISAM, estas cuando se realiza una inserción quedan bloqueadas, debido a que en la duración del insert no se puede realizar ninguna operación. En cambio la tabla Innodb utiliza funciones de bloqueo a nivel de fila que permite que solo se bloquee la fila que se esté modificando y no toda la tabla y podríamos seguir ejecutando otras funciones.

ISAM

Estas tablas provienen de versiones antiguas de MySql. Estas fueron renovadas por las tablas MyISAM en la versión 3.23. Por lo que si nos encontramos con algunas de estas tablas, es posible que estemos trabajando con una base de datos antigua. La principal diferencia entre las dos, es que el índice de las tablas MyISAM es mucho más pequeño que en el de las tablas ISAM, de manera que un SELECT con un índice sobre

una tabla MyISAM además de ser mucho más rápido, utilizará mucho menos recursos del sistema.

MyISAM

Con este tipo de tablas tenemos tres versiones:

- Estáticas. Estas tablas son de longitud fija, siendo el tamaño de cada registro de 10 bytes. Estas tablas se caracterizan por ser muy rápidas, se almacenan en caché, se reconstruye fácilmente ante un fallo y requieren mas espacio en disco.
- Dinámicas. En estas tablas las columnas disponen de diferente tamaño. Aunque este tipo de datos ahorra espacio en disco es mucho más complejo. Las características de estas tablas son que todas las columnas de cadena son dinámicas, por regla general ocupan menos espacio que las estáticas, requieren un mantenimiento regular para evitar la fragmentación y son más difíciles de reconstruir tras un fallo.
- Comprimidas. Son as tablas de solo lectura que utilizan mucho menos espacio en disco. Son útiles para datos que no cambian y se dispone de poco espacio. Sus características es que son más pequeñas, la carga de acceso es menor, cada columna se puede comprimir con diferentes algoritmos.

Merge

Estas tablas son la fusión entre tablas MyISAM iguales. De forma general estas tablas se usan cuando las tablas mencionadas empiezan a ser demasiado grandes. Las ventajas de estas tablas es que son más rápidas en ciertas situaciones y el tamaño de la tabla suele ser pequeño. En cambio las desventajas son más lentas en las búsquedas y el comando Replace no funciona entre ellas.

Heap

Las tablas Heap son el tipo de tabla más rápido porque se almacena en memoria y utilizan un índice asignado. La contrapartida es que como se almacenan en memoria, en el caso de una falla del sistema, los datos se pierden.

- Niveles de transacción

Se puede modificar el comportamiento predeterminado al trabajar con transacciones mediante el establecimiento del nivel de transacción. Existen varios niveles de transacción en MySql. En concreto admite los siguientes niveles de aislamiento de transacción.

- READ UNCOMMITTED

Este nivel permite transacciones para leer datos sin confirmar desde otras transacciones (es lo que se conoce como lectura sucia).

- REPEATABLE READ

Este nivel no permite lecturas no susceptibles de repetición (que son las que se dan cuando otra transacción ha modificado los datos, incluso si se han confirmado).

- SERIALIZABLE

Este nivel no permite lecturas fantasma, que tienen lugar cuando otra transacción ha confirmado una nueva fila que coincide con los resultados de nuestra consulta. Los datos serán los mismos en cada ocasión.

- Propiedades de las sentencias

MySql proporciona soporte completo para operadores y funciones en las consultas dentro del Select y Where, también admite Group By y Order By para las funciones de agrupación. Se puede utilizar Left Outer Join y Right Outer Join cumpliendo los estándares de SQL. Se pueden crear alias en tablas y columnas como especifica el estándar de SQL. Las operaciones de Delete, Insert, Replace y Update siempre devuelven el número de filas que han sido afectadas.

- Propiedades de las funciones

El comando específico Show se puede usar para obtener la información acerca de la base de datos, tablas e índices. Con el comando Explain obtenemos como resuelve una consulta.

Los nombres de funciones no son problemáticos con los nombres de las tablas o sus columnas.

Se puede utilizar tablas de distintas bases de datos en la misma consulta.

- Seguridad

Dispone de un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite la verificación basada en el host. Las contraseñas son seguras ya que cuando se realiza algún tráfico de contraseñas estas están encriptadas al conectarse con un servidor.

- Límites

Está preparado para gestionar base de datos con grandes contenidos de registros en la base de datos (50 millones de registros). El máximo de índices por tabla es de 64, siendo cada índice de 1 hasta 16 columnas. El máximo ancho de límite es de 1000 bytes.

- Conectividad

Los usuarios se pueden conectar a través del servidor MySql usando TCP/IP en cualquier plataforma. Bajo la plataforma de Windows se puede utilizar pipes para la conexión, mientras que para Unix, los usuarios se pueden conectar usando ficheros.

Los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción –shared memory. De esta forma los usuarios se pueden conectar a través de memoria compartida.

- Localización

El servidor nos proporciona mensajes de error a los usuarios en distintos idiomas. Posee soporte completo para diferentes conjuntos de caracteres, estando disponible el soporte Unicote.

Todos los datos se almacenan en el conjunto de caracteres elegido. Las comparaciones para columnas normales de cadena de caracteres son insensibles a mayúsculas y minúsculas.

La ordenación por defecto es la Sueca a no ser que se haya especificado un conjunto de caracteres. Esta ordenación es posible cambiara cuando arranca el servidor.

- Herramientas

MySQL Server tiene disponible comandos SQL para chequear, optimizar, y reparar tablas. Todos estos comandos están disponibles mediante la línea de comandos y el cliente mysqlcheck. MySQL también incluye myisamchk, que es una utilidad en línea de comandos muy rápida para poder ejecutar estas operaciones sobre las tablas MyISAM.

4.2: Visual Studio. C#

4.2.1: Introducción

Visual C# suministra a los profesionales de la programación un lenguaje moderno basado en la orientación de componentes, para poder crear con gran rapidez soluciones muy interesantes controladas en los datos. Debido a la rapidez de programación, diseño e implementación de los servicios Web XML, estas soluciones controladas por los datos, Visual C# proporciona una funcionalidad superior para la optimización de los procesos empresariales.

Ofrece a los programadores generar soluciones mediante formularios y dispositivos cliente ligeros e inteligentes. Debido a que es un lenguaje de programación elegante, los programadores pueden mezclar sus conocimientos de C++ y Java para disfrutar de una experiencia satisfactoria con la nueva plataforma.

4.2.2: Características de C#

Algunas de las siguientes características de las que se comenta no son propias del lenguaje C# sino sobre la plataforma .NET, pero si tienen repercusión sobre el lenguaje. Las características son:

- **Sencillez:** En C# se elimina muchos elementos que otros lenguajes aportan y que no son necesarios en .NET. Por ejemplo:
 - El código escrito en C# es auto contenido, lo que quiere decir que no necesita de ficheros extras al propio fuente como ficheros de cabecera o ficheros IDL.
 - Para el tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), con esto se consigue facilitar la portabilidad del código.
 - Se elimina elementos que no son útiles de lenguajes C++ como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) .

- **Modernidad:** C# añade al lenguaje elementos que a lo largo de los años ha ido demostrándose que son muy útiles en el desarrollo de las aplicaciones y que en otros lenguajes como por ejemplo Java o C++ hay que simular, como un tipo básico **decimal** que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero). El añadir la instrucción **foreach** que permita recorrer colecciones más fáciles y ampliable a tipos definidos por el usuario, la inclusión de un tipo básico **string** para representar cadenas o la distinción de un tipo **bool** específico para representar valores lógicos.
- **Orientación a objetos:** En la actualidad cualquier lenguaje de programación para propósito general está orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia que podemos encontrar de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro porque no permite la definición de funciones o variables globales, sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos. Con esto conseguimos reducir problemas por conflictos de nombres y facilita en mayor medida la legibilidad del código.

Otras características de C# respecto a la programación de orientada a objetos.: **encapsulación, herencia y polimorfismo.**

Respecto a la característica de encapsulación hay que señalar que aparte de los típicos modificadores **public**, **private** y **protected**, C# define un cuarto modificador llamado **internal**. Este nuevo modificador que puede combinarse con **protected** e indica que al elemento al cual se está definiendo sólo puede accederse desde su mismo ensamblado.

Sobre a la herencia al igual que java y diferente que C++, C# sólo admite herencia simple de clases, debido a que la múltiple provoca más problemática que facilidades, y en la mayoría se puede sustituir por interfaz y herencia simple. De todos modos, esto vuelve a ser más bien una característica propia del CTS que de C#.

- **Orientación a componentes:** La propia sintaxis de C# añade elementos propios del diseño de componentes que otros lenguajes deben simular a través de construcciones más o menos complejas.
- **Gestión automática de memoria:** Todo lenguaje de .NET posee para su utilización el recolector de basura. El recolector tiene el efecto en el lenguaje de que no hace falta añadir instrucciones de destrucción de objetos. Por el contrario, debido a que la destrucción de los objetos a través del recolector de basura es indeterminista y sólo se realiza cuando éste se active, ya sea por falta de memoria, finalización de la aplicación o solicitud explícita en el fuente. Mediante la instrucción `using` se proporciona un mecanismo de liberación de recursos.
- **Seguridad de tipos:** C# añade mecanismos en los que se permite asegurar que los accesos a los datos siempre se realicen correctamente. Esto permite evitar que se produzcan errores más difíciles de detectar debido al acceso a memoria no perteneciente a ningún objeto y es especialmente imprescindible en un entorno gestionado por un recolector de basura. Para todo esto se toman precauciones del tipo:
 - Sólo se puede **convertir entre tipos compatibles**. Entre un tipo y el antecesor suyo, entre tipos para los que se haya definido explícitamente un operador de conversión, y entre un tipo y un tipo de un hijo suyo del que un objeto del primero almacenase alguna referencia del segundo (**downcasting**). Obviamente, la última posibilidad sólo la puede comprobar en tiempo de ejecución el recolector de basura y no el compilador, por lo que en realidad entre el recolector y el compilador aseguran la corrección de las conversiones.
 - No está permitido usar **variables no inicializadas**. El compilador asigna a los campos un valor por defecto, consistente en ponerlos a cero y controla a través del análisis el flujo de control del fuente que no se lea ninguna variable sin haberse inicializado.

- Se verifica que todo **acceso a los elementos de una tabla** se realice con índices que esté dentro del rango.
- Con diferencia de Java, C# incluye **delegados**. Son parecidos a los punteros a funciones de C++, pero mantienen un enfoque orientado a objetos, pueden guardar referencias a varios métodos a la vez, y se verifica que los métodos a los que apunten, posean parámetros y valores de retorno del tipo indicado a la definición.
- Se permite definir métodos que admitan un número no determinado de parámetros de un cierto tipo, pero a diferencia sobre C++, en C# siempre se verifica que los valores que se les pasen en cada llamada sean de los tipos definidos.
- **Instrucciones seguras:** Para no cometer errores comunes, en C# se han definido unas determinadas restricciones en el uso para las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición debe ser una expresión condicional y no aritmética, con esto se evitan errores por confusión del operador de igualdad (==) con el de asignación (=). También todo caso de un **switch** debe terminar con un **break** para indicar cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.
- **Sistema de tipos unificado:** Con diferencia sobre C++, en C# todos los tipos de datos que vayamos a definir, siempre derivarán, aunque sea de forma implícita, de una clase base común denominada **System.Object**, con lo que se dispondrán de todos los miembros definidos en ésta clase (es decir, serán “objetos”)

Pero respecto a Java, en C# esto también es aplicable a los tipos de datos básicos. Para conseguir que todo lo anterior no tenga una repercusión negativa en su nivel de rendimiento, se ha añadido otro mecanismo, transparente de **boxing** y **unboxing**, con el que se obtiene que sólo sean tratados como objetos cuando la situación sea necesaria, y por el contrario se les puede aplicar optimizaciones más específicas.

El hecho de que todos los tipos que se definan, vengan de una clase común, facilita en gran medida que el diseño de las colecciones genéricas que puedan guardar objetos de cualquier tipo.

- **Extensibilidad de tipos básicos:** C# permite definir, a través de **estructuras**, tipos de datos para los que se utilicen idénticas optimizaciones que para los tipos de datos básicos. Es decir, que sea posible guardar en una pila y se asignen por valor y no por referencia. Para poder lograr que lo último no posea efectos ineficientes al pasar estructuras por parámetros de métodos, se da la posibilidad de pasar referencias a pila por medio del modificador de parámetro **ref**.

Extensibilidad de operadores: Con la idea de facilitar la legibilidad del código y poder conseguir que los nuevos tipos de datos básicos que estén definidos, estén al mismo nivel que los básicos predefinidos en el lenguaje, a diferencia de Java y al igual que C++, C# permite redefinir el significado de la gran mayoría de los operadores.

- **Ampliación de Versiones:** C# añade una **política de versionado** que da posibilidad de crear nuevas versiones de tipos sin miedo a que la introducción de nuevos miembros vayan a crear errores difíciles de detectar en hijos previamente desarrollados y ya extendidos del mismo nombre que los recién introducidos.

Si una clase introduce un nuevo método cuyas redefiniciones deban seguir la regla de llamar a la versión de su padre en algún punto de su código, difícilmente seguirían esta regla miembros de su misma signatura definidos en clases hijas previamente a la definición del mismo en la clase padre; o si introduce un nuevo campo con el mismo nombre que algún método de una clase hija, la clase hija dejará de funcionar. Para evitar que esto ocurra, en C# se toman dos medidas:

- **Eficiente:** En un principio, en C#, todo el código añade numerosas restricciones para poder asegurar su seguridad y no evitar la posibilidad del uso de punteros. Sin embargo, y con diferencia de Java, en C# es posible no incluir dichas restricciones manipulando los objetos a través de punteros. Para hacer esto, basta con marcar regiones de código como inseguras y podrán usarse en todas ellas

punteros de forma parecida a cómo se hace en C++. Lo que puede resultar muy útil para situaciones donde se requiera una eficiencia y velocidad de procesamiento superior.

- **Compatible:** Para ayudar con la migración de programadores, C# no sólo mantiene una sintaxis muy parecida a C, C++ o Java que se puede añadir directamente en código escrito fragmentos de código escrito en estos lenguajes, sino que el recolector de basura también ofrece la posibilidad de acceder a código nativo escrito mediante funciones sueltas no orientadas a objetos como las DLLs de la API Win32.

4.2.3: Resumen

C# es un nuevo lenguaje orientado a objetos con el objetivo de que recoger en un único lenguaje las principales características de Java y de C++, y algunas influencias respecto Visual Basic.

Similitudes con Java:

- Declaración de clases.
- Interfaces.
- Operaciones booleanas.
- Control de errores.
- Recolector de basura.

Similitudes con C/C++:

- Modo de compilación.
- Estructuración.
- Pre-procesador.
- Sobre-carga de operadores.

Nuevas características:

- Declaración en el espacio de nombres.
- Existe una variedad más amplia de tipos de datos que los que están disponibles en C, C++ o Java.

- Para cada clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.
- Se puede declarar un número variable de argumentos como paso de parámetros en los métodos.
- Antes de que un método vaya a ser redefinido en una clase, se debe declarar el mismo como virtual. El método redefinido en la subclase debe ser declarado con la palabra override.
- Un objeto COM tiene unas propiedades, y visto que las clases en C# se pueden utilizar como objetos COM, C# posibilita la declaración de propiedades dentro de cualquier clase.
- Se permite que a los desarrolladores mantengan múltiples versiones de clases en forma binaria, almacenándose en distintos espacios de nombres. Esto da a lugar que nuevas versiones y anteriores de software puedan ejecutarse de forma simultánea.

4.3: Aqua Data Studio.

4.3.1: Introducción

Aqua Data Studio es más que una herramienta para realizar consultas en el lenguaje de SQL. Aqua Data Studio ofrece a los programadores de base de datos una interfaz agradable para poder crear, editar y ejecutar consultas SQL.

Aqua Data Studio posee cobertura sobre cualquier servidor de base de datos que este disponible en la máquina.

Con esta herramienta se puede ver y comprender de forma visual, la estructura de la base de datos que tengamos preseleccionada. Además incluye la posibilidad de salvar los resultados de cualquier consulta en distintos formatos incluyendo HTML o XML.

4.3.2: Características

- Se puede utilizar Windows, Linux o sobre cualquier plataforma en la que se pueda ejecutar Java.
- Ofrece soporte para distintos servidores de base de datos. Entre ellos podemos encontrar MySql, Oracle , SyBase, DB2, ASE 12.5, PostgreSQL.
- Suministra un potente explorador de base de datos. Para cada base de datos se puede consultar las tablas, las vistas, los triggers, los procedimientos y las funciones. Por ejemplo, para cada tabla se puede consultar sus atributos, índices, restricciones y triggers.
- Se puede crear cualquier secuencia y ejecutarla. Posee una ayuda para iniciar una consulta, así como saber la consulta necesaria para crear una tabla con todos sus atributos y restricciones.
- Permite añadir colores a las sintaxis según especifiquemos la base de datos en la que nos encontremos.
- Se puede almacenar los resultados obtenidos tras ejecutar una consulta mediante CSV, HMTL y XML. Además de almacenar los datos, estos son visualizados en el momento de ejecutar la consulta.

- Ofrece una ayuda para completar las sentencias SQL en el momento de escritura. Por ejemplo con esta ayuda no hace falta saber los atributos que posee una tabla.
- Suministra las operaciones de importación y exportación para los datos que queramos, o en su defecto todos los atributos, los de una tabla de la base de datos.
- Añade otra utilidad para visualizar como están relacionadas las tablas o las vistas entre sí. Con este visualizador se puede averiguar las conexiones entre las tablas y los nombres de las restricciones que poseen.

Capítulo 5: Manual De Usuario

5.1 Instalación

Para poder utilizar antes la aplicación, debemos ejecutar antes el instalador para poder tener el archivo ejecutable de la aplicación.

Este paso es bien sencillo, tan solo debemos de hacer doble clic sobre el archivo ejecutable. A continuación nos saldrá una ventana de aviso, de si queremos ejecutar el archivo como la siguiente:

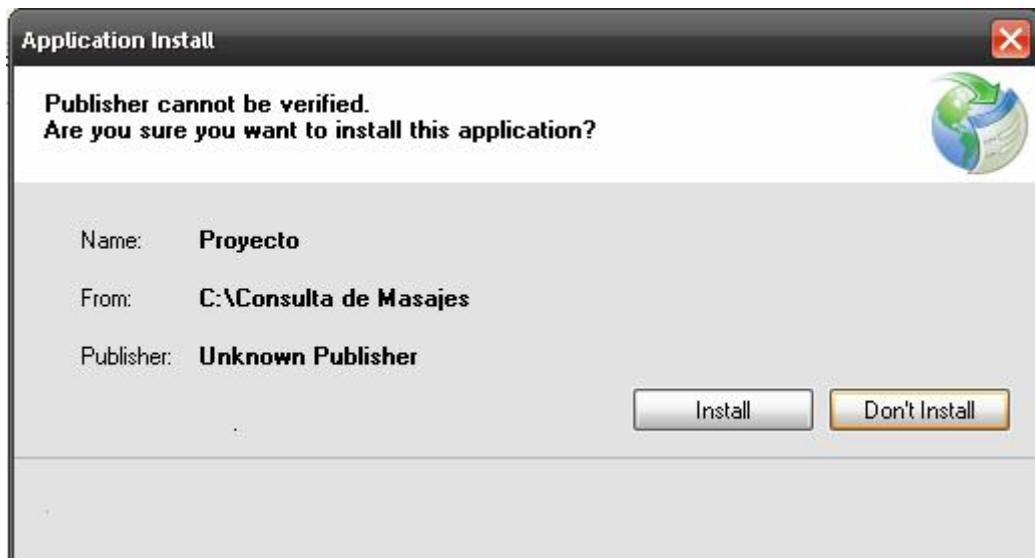


Fig71.Instalación de la aplicación.

Después de pulsar sobre el botón de Install no aparece una ventana con el progreso de la instalación como la siguiente:



Fig72. Progreso de la instalación.

Y automáticamente a continuación se muestra el inicio de la aplicación:



Fig73 Acceso al programa.

El nombre del usuario y de la contraseña, son los mismos que los que posee la base de datos MySql, ya que nos conectamos directamente a la base de datos.

5.2Introducción

Cada vez que entremos en la aplicación vamos a tener siempre a la vista la siguiente ventana:

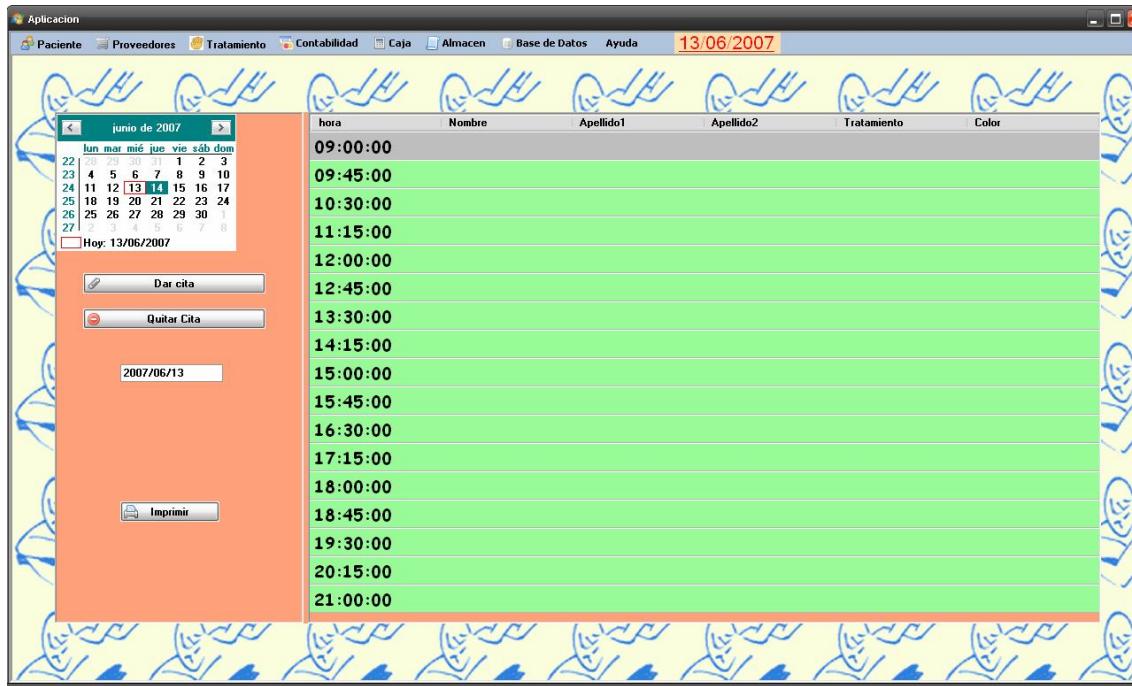


Fig74 Ventana Principal

Como se puede ver, siempre vamos a tener a primera mano la agenda, para poder saber cuales son los tratamientos y pacientes del día actual. En la parte superior tenemos el menú con toda la funcionalidad de la aplicación.

De izquierda a derecha vamos a comentar las funciones que posee la aplicación. En primer lugar tenemos a Paciente, en ella podemos realizar todas las operaciones con los paciente, inserción, modificación, consultar sus datos etc. Después nos siguen los proveedores que es ídem de lo mismo que los pacientes. A continuación tenemos tratamiento que además de lo mismo que con los pacientes también aquí es donde vamos a abonar a los pacientes con los bonos de los tratamientos. Luego tenemos la contabilidad, aquí podremos realizar facturas, consultar las deudas que hay con los pacientes y proveedores y ver las facturas de los pacientes. Siguiendo con la contabilidad al lado está la caja, en ella podremos ver la entrada y salida de dinero de cada día. Seguidamente aparece el almacén que nos servirá para llevar un control sobre los productos que tenemos. Como penúltima opción aparece la base de datos, cuya

utilidad es para poder exportar e importar la base de datos. Para terminar aparece la ayuda, en la que se puede consultar la ayuda en diferentes formatos.

A continuación se detalla y esboza con pantallas toda la funcionalidad de la aplicación.

5.3 Manual de Usuario

5.3.1 Elementos comunes

La aplicación consta de una serie de comportamientos comunes en las diversas ventanas que posee. En este apartado hablaremos de todos los posibles iconos, botones, textos o acciones que poseen en común, de forma que mas adelante cuando acaparemos el desarrollo de la ventana nos centremos exclusivamente en el objetivo de esta.



Fig75. Botón Nuevo

Con este botón, damos la posibilidad al usuario de crear una nueva entidad dependiendo de la ventana en la que nos encontremos.

Al hacer clic sobre el se nos abrirá una ventana para que podamos llenar los datos necesarios para que exista la entidad.



Fig76. Botón Consultar

Cuando tenemos seleccionado una entidad en la tabla y pulsamos sobre este botón, a continuación se nos abre una ventana para consultar los datos que posee la entidad.



Fig77. Botón Editar

Este botón posee la misma funcionalidad que el botón anterior, con la diferencia de que en la ventana que se abre nos permite la opción de editar los datos.

El objetivo de este botón es modificar los datos almacenados sobre la entidad seleccionada.



Fig77.Botón Eliminar

Si pulsamos sobre este botón, además de mostrarnos un mensaje de confirmación, nos da la opción de eliminar la entidad que tenemos seleccionada



Fig78.Botón Volver

Este botón es uno de los más importantes. Además de cerrar la pantalla en la que se encuentra, manda la información necesaria a la ventana que lo llamó. De esta forma devolveremos la información a la ventana anterior.



Fig79.Botón Aceptar

Al pulsar sobre este botón provocamos que se guarden los cambios producidos desde que se inicio la ventana en la que posee el botón hasta que decidimos pulsarlo.



Fig80.Botón Cancelar

A diferencia con el botón anterior, cuando pulsamos sobre 'Cancelar' deshacemos los cambios desde que abrimos la ventana.



Fig81. Botón Ver

Este botón es parecido al botón consultar. En lo que se diferencia es que este botón hace referencia cuando queremos ver el contenido de una fila de una tabla de la base de datos.



Fig82. Botón Imprimir

Al pulsar sobre este botón se nos abre una visión previa para mostrarnos el resultado de imprimir la ventana en la que estamos.

Si en la ventana que nos encontramos existe una tabla, esta tabla será impresa en otra hoja distinta.

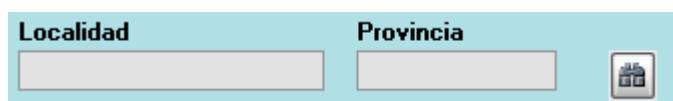


Fig83. Buscar Datos

Cuando veamos una caja de texto y al lado un botón con los prismáticos, significa que el contenido de la caja puede ya estar almacenado. Para llenar la caja pulsamos el botón y se nos abre la siguiente ventana:



Fig84. Ventana Buscar

Ahora si seleccionamos una fila de la tabla y le damos a volver conseguimos rellenar la caja de texto con la fila seleccionada. Cuando estemos en esta ventana no podremos hacer clic sobre el botón volver sin seleccionar una fila. En cambio si la tabla no contiene la información que necesitamos porque todavía no está almacenada, procederemos a ello. Primero pulsamos sobre el botón que tiene el icono de un disket (de mantenimiento). Ahora se nos aparece la siguiente ventana:



Fig85. Ventana Mantenimiento Datos Simples.

El funcionamiento de esta ventana es el siguiente. Si escribimos la información en las cajas de texto que existan y le damos al botón de 'Añadir', agregamos la información y se actualiza la tabla. Si pulsamos sobre el botón de 'Nuevo' borramos el contenido de las cajas de texto. Si pulsamos sobre el botón de eliminar nos aparece un dialogo de advertencia. Si lo confirmamos y la fila no está relacionada con ninguna fila de otra entidad se procede a eliminarlo, pero si existe alguna relación nos informa con un mensaje del motivo por cual no se puede eliminar. A continuación se muestra la ventana de dialogo de advertencia y la de información:

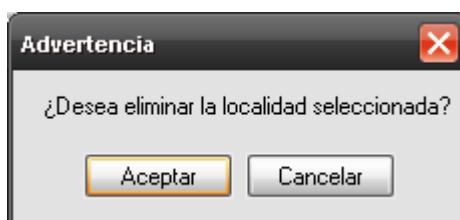


Fig86. Ventana Advertencia

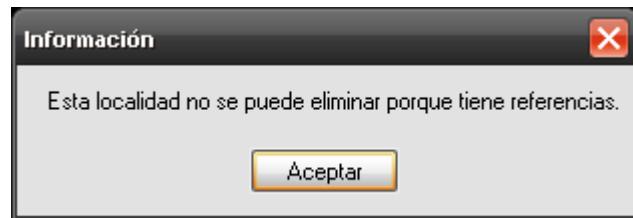


Fig87. Ventana Información



Fig88. Filtrar

Cuando veamos en las ventanas de buscar este filtro, es para poder filtrar el contenido de la tabla. Si por ejemplo estamos en la ventana de buscar paciente y podemos filtrar por el nombre podemos poner por ejemplo 'Antonio'. En la tabla aparecerán todas las personas que tengan de nombre Antonio, como Antonio, Antonio Jesús o Francisco Antonio. También el filtro de búsqueda es dinámico, es decir, de forma que vamos escribiendo la tabla se va filtrando con las condiciones anteriores y con el texto escrito. Un ejemplo de esto es cuando escribimos 'An' de 'Antonio', pues aparece cualquier fila que en su nombre aparezca 'an', como por ejemplo Francisco.

Para terminar la introducción se aclara que todas las ventanas aparecerán por defecto en el centro de la pantalla. La ventana inicial de la aplicación es la agenda, esta ventana nunca podrá ser cerrada. Si estamos en una ventana y pulsamos un botón que nos abre una segunda ventana, no tendremos acceso a la primera ventana hasta cerrar la segunda. El colorido de las ventanas depende del tema propio de la ventana. A continuación se muestra una tabla con el color y el tema que están relacionados.



Fig89. Color1

Este color se refiere a cualquier ventana relacionada con el almacén (naranja claro).



Fig91. Color2

Este color hace referencia a ventanas relacionadas con la economía. Algunas de las ventanas son como Caja, Facturas etc (azul claro).



Fig92. Color3

Este color se verá en todas las ventanas de paciente (celeste).



Fig93. Color4

Este color es para el proveedor y sus productos (rosa claro).



Fig94. Color5

Este color se verá en las últimas ventanas accesibles de la aplicación. Son ventanas de mantenimiento de la base de datos como profesión, unidad etc (salmon).



Fig95. Color6

Este último color se verá en las ventanas de los tratamientos (verde).

5.3.2 Acceso al sistema

Mediante la siguiente ventana restringimos el acceso a la aplicación. Si no se posee la información necesaria para acceder a la aplicación se cancela la operación de acceso.



Fig96. Ventana Acceso

Si pulsamos el botón conectar y los datos son correctos nos adentramos en la aplicación y se inicia con la ventana inicial. Por el contrario si los datos son incorrectos se muestra un mensaje de error y se cierra la ventana de acceso.

Cuando el usuario introduce los datos correctos se le muestra la siguiente ventana.



Fig97. Ventana Principal

Esta ventana es la ventana principal. Tiene como característica que no se podrá cerrar nunca, a no ser que se cierre la aplicación por completo.

Como se puede observar, tenemos siempre visible la agenda de citas del día que tenemos seleccionado. Cuando se inicia por primera vez la ventana aparece por defecto el día actual.

A la izquierda podemos observar que existe un calendario. Gracias a este calendario se puede navegar por los días para visualizar en la tabla como está el día de citas.

Debajo del calendario tenemos tres botones. Una vez hayamos seleccionado una fila de la tabla y pulsamos:

Si el botón pulsado es "Dar de cita", se nos abre la siguiente ventana:

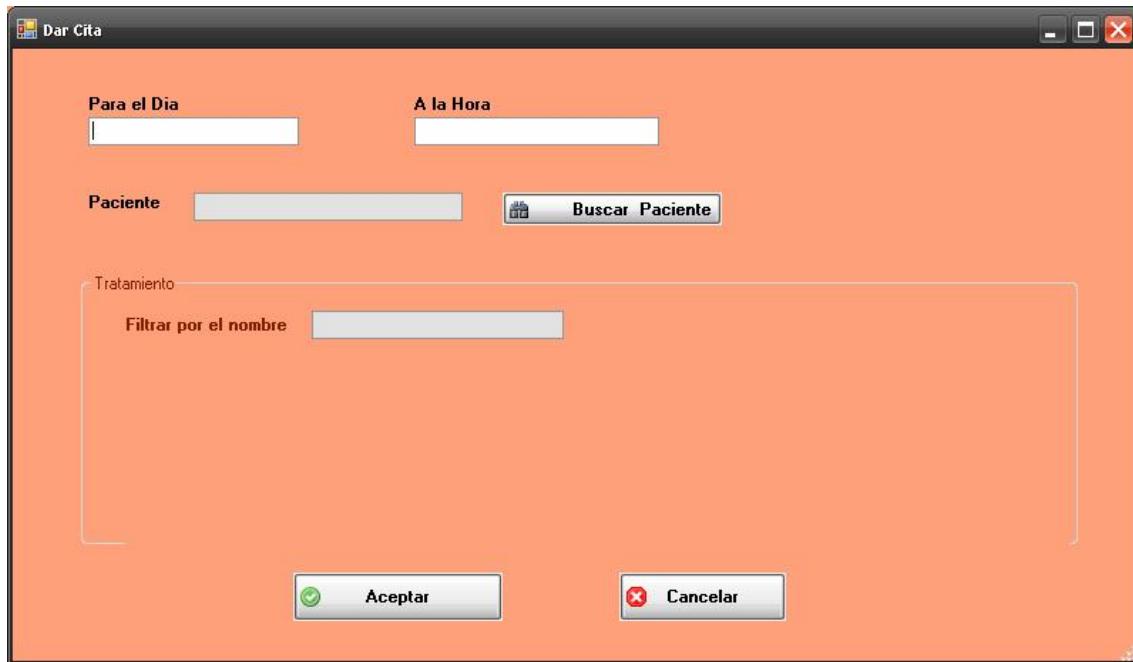


Fig98. Ventana Dar Cita

En esta ventana si pulsamos el botón buscar paciente nos aparece una ventana que la comentaremos mas adelante. Nos basta con saber que hay seleccionaremos el paciente al que le daremos la cita. En los textos de fecha y hora aparecen por defecto la fila que estaba seleccionada en la ventana principal. Cuando hemos seleccionado el paciente que deseábamos nos aparece en la tabla de tratamiento, los tratamientos que puede recibir según los bonos que tenga activo. Si pulsamos sobre el botón 'Aceptar' guardamos la cita que hemos creado (si todos los datos son correctos). Si pulsamos sobre el botón 'Cancelar', cancelamos la operación de dar cita

Por el contrario si el botón pulsado es "Quitar Cita", borramos la cita que estaba seleccionada tras confirmar el mensaje del que se nos advierte de la operación.

El último botón, es el de imprimir, con el que conseguiremos imprimir el listado del día que esté seleccionado en el calendario. Con este listado podremos tener en un formato físico la agenda de un día.

La tabla que muestra los datos de la cita se compone de los campos Hora, Nombre, Apellido1, Apellido2, (datos del paciente), Tratamiento (tratamiento al que es sometido) y color del tratamiento. El color representa el grado de perspectiva que le

quiera dar el usuario. Con esto conseguimos que a primera vista se pueda ver como está organizado un día en concreto.

5.3.3 Pacientes



Después de pulsar sobre este enlace desde la ventana principal nos aparece la siguiente ventana.



Fig99. Ventana Pacientes

Esta ventana aparecerá por toda la aplicación (y también otras muy parecida a esta). En esta ventana se puede dividir en dos partes.

La primera parte son los botones que aparecen a la izquierda en modo columna. Estos botones están comentados en el apartado de introducción.

Si pulsamos sobre el botón de 'Nuevo' (la diferencia con los demás ya está comentada) nos aparece la siguiente ventana:

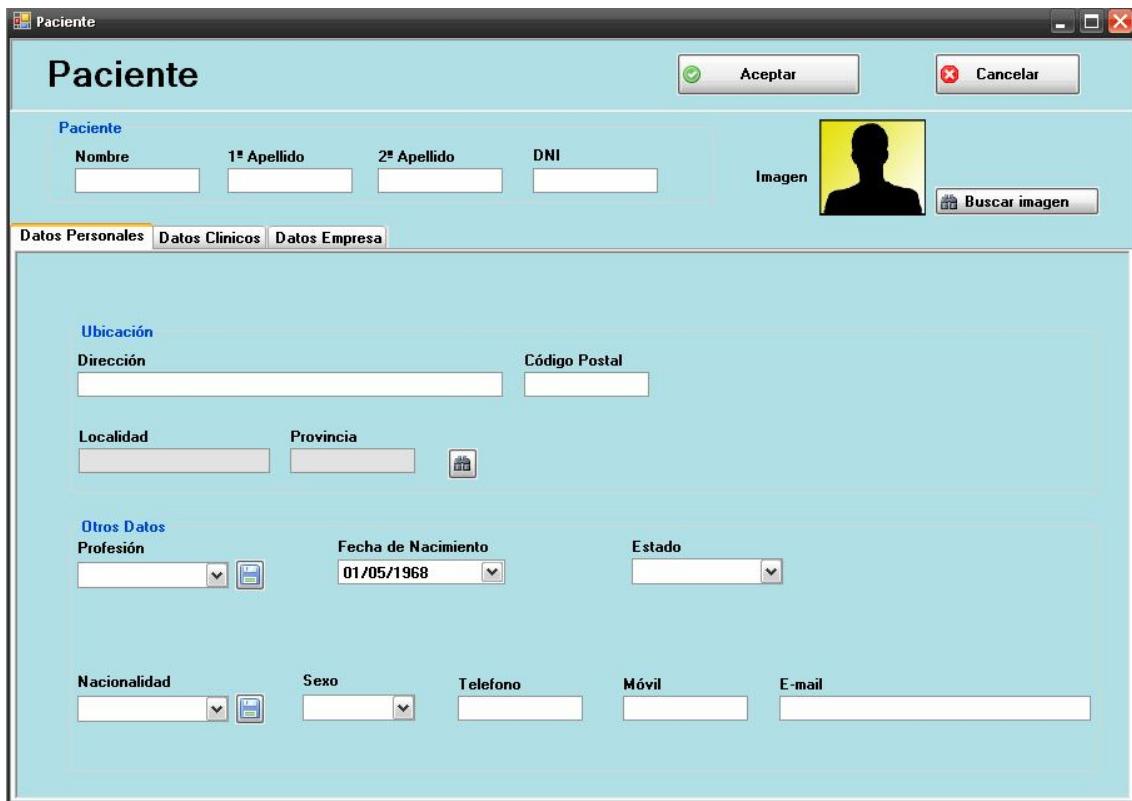


Fig100. Ventana Paciente Nuevo

Esta ventana está partida en tres rectángulos horizontales:

El primer rectángulo lo forma los dos botones de Aceptar y Cancelar que mantiene el significado de Introducción.

El segundo rectángulo lo forma la información general de un paciente. Esta información consta de Nombre, Apellidos, DNI y una imagen.

El tercer rectángulo adquiere toda la información necesaria de un paciente. Como la información es demasiada y heterogénea para una misma ventana, aparece para una visión más clara con tres pestañas. La primera pestaña tiene los datos personales del paciente. Aquí aparece el primer ejemplo del botón buscar en la caja de texto de la provincia. La segunda pestaña es los datos clínicos. En esta pestaña se almacenan los datos sobre la estatura, peso, presión arterial, pulso condiciones del cuerpo y operaciones que haya recibido. La tercera pestaña refiere a la empresa. En esta pestaña por defecto aparece deshabilitada la opción de que nuestro paciente venga de una empresa. En el caso del que el paciente venga de una empresa solo hay que seleccionar la casilla y darle al botón de buscar empresa.

A continuación nos aparece la ventana que nos muestra las empresas que están guardadas:

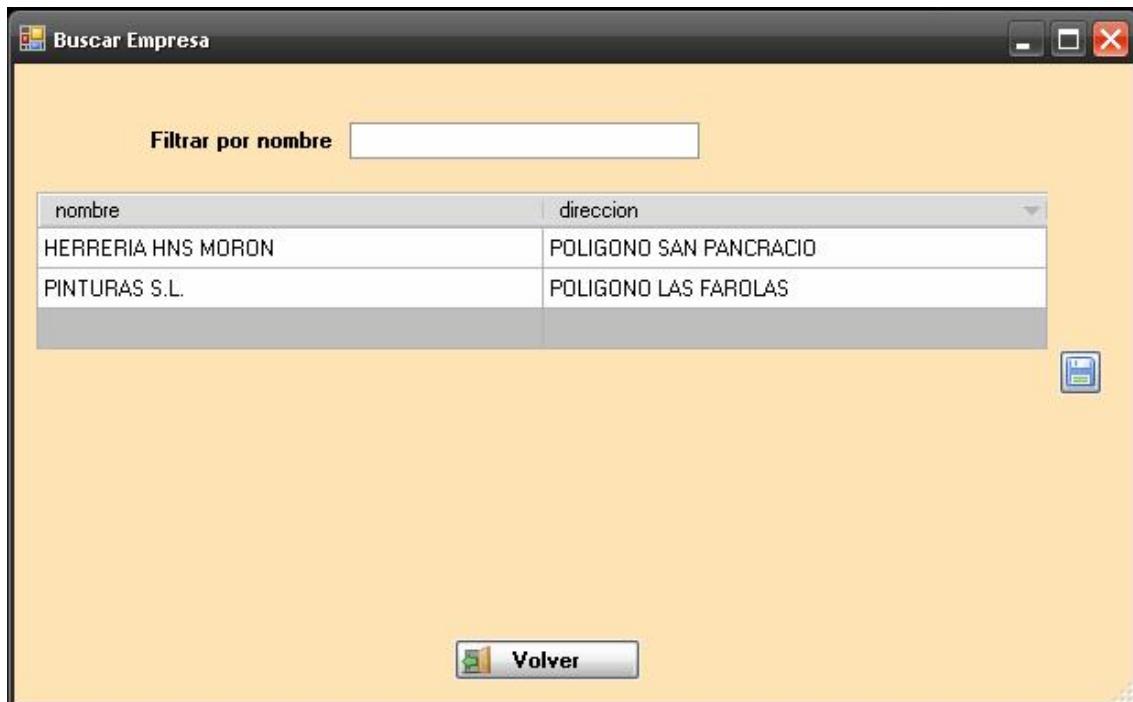


Fig101. Ventana Buscar Empresa

y luego la ventana que aparece es si pulsáramos el botón de mantenimiento de la empresa:



Fig102. Ventana Mantenimiento Empresa

El funcionamiento de esta última ventana es común para todas las ventanas de mantenimiento. A la izquierda están los mismos botones, al centro tenemos la tabla con los datos almacenados actualmente y en la parte superior de la tabla están las cajas de texto para insertar nuevos o modificar ya algunos.

Si seguimos con la ventana que empezamos, que era la búsqueda de pacientes nos falta por detallar la segunda parte.

La segunda parte está compuesta por el centro de la pantalla. En el centro tenemos una tabla que muestra el nombre y los apellidos de todos los pacientes que tenemos almacenados. En la parte superior de la tabla tenemos una caja de texto para filtrar la tabla. Para terminar tenemos otro botón que es el de más datos.

Si pulsamos sobre este botón se nos abre una ventana con información complementaria del paciente. A continuación mostramos esta ventana.



Fig103. Ventana Asistencia Paciente

Por defecto al abrir la ventana se muestra la primera pestaña que es la de asistencia. En esta pestaña se puede apreciar un listado con todas las asistencias que tiene el paciente.



Fig104. Ventana Historiales Paciente

Al pulsar sobre la pestaña de historiales se muestra una tabla con todos los historiales que tiene el paciente. Un historial solo puede estar asociado a una asistencia y viceversa. A la derecha de la tabla podemos ver los botones de nuevo, ver, imprimir y eliminar. Para el botón de nuevo es necesario tener seleccionado una asistencia de en la pestaña de asistencia. Para el resto de botones es necesario tener seleccionado un historial. Si pulsamos sobre el botón de Nuevo nos aparece una ventana como la siguiente para llenar los datos necesarios de un historial.

Paciente: [Text input field]

Principal

Motivo de la consulta: [Text input field]

Fecha de inicio del dolor: [Text input field] **Tipo de dolor:** [Text input field]

Possible causa: [Text input field]

Antecedentes del motivo

¿Ha visitado a un médico o terapeuta? [Text input field] **Diagnóstico:** [Text input field] **Tratamiento actual:** [Text input field]

Evolución: [Text input field] **Ha ocurrido antes:** [Text input field] **¿Cuando?** [Text input field] **¿Cuánto tiempo dura?** [Text input field]

¿Qué lo provocó entonces? [Text input field] **Tratamientos anteriores:** [Text input field]

Evolución: [Text input field] **Informes Médicos:** [Text input field]

Otros

¿Ha habido algún cambio importante en su vida? [Text input field]

Observaciones: [Text input field]

Aceptar **Cancelar**

Fig105. Ventana Historial

La funcionalidad de la ventana es sencilla y no es necesario documentarla.



Fig106. Ventana Pruebas Paciente

Al pulsar sobre la pestaña de pruebas nos aparece una tabla con todas las pruebas que se ha sometido el paciente. Cada prueba se refiere a una única asistencia. A la derecha de la tabla tenemos los mismos botones que historiales. En esta pestaña para crear una nueva prueba pulsando el botón nuevo es necesario tener seleccionado una asistencia y el historial que le corresponde. Para los demás botones el comportamiento es el mismo que en el historial. Si pulsamos el botón Nuevo la ventana que aparece es la siguiente:

Software para la gestión de una consulta de masajes terapéuticos

Fig107. Ventana Pruebas

Que tampoco es necesario detallarla.

Fecha	Nombre	Definicion	PrecioTotal	Pagado	Disponibles
16/04/2007	HIDRATANTE CHOCOLATE	DECIMO	215	45	5
01/05/2007	Terapeutico espalda	Decimo	180	0	8
01/05/2007	Drenaje Linfatico de piernas	Quintuple	110	0	4

Fig108. Ventana Añadir Pago

Al pulsar la pestaña de bonos vemos que el contenido de la pestaña es una tabla con todos los bonos que tiene el paciente. En la tabla se puede ver el estado del bono. En la parte inferior derecha hay una caja de texto con el contenido del saldo que tiene el cliente. En la parte inferior izquierda tenemos un botón para añadir un pago del bono que tengamos seleccionado. La ventana que se muestra tras pulsar el botón es la siguiente:



Fig109. Ventana Pago Bono

Con esta ventana podemos añadir una cuantía con la información que se muestra.

Al cerrar la ventana de añadir el pago se vuelve a la pestaña de bonos y se actualiza la información.



Fig110. Ventana Pagos Paciente

Si pulsamos la pestaña de pagos, se muestra una tabla con la información de la fecha, la hora, el concepto y la cuantía de todos los pagos recibidos por el paciente.

5.3.4 Proveedores



Si pulsamos sobre proveedores en la ventana principal, nos aparece la siguiente ventana:



Fig111. Ventana Proveedores

Esta ventana es muy parecida a la ventana anterior de buscar paciente. La

funcionalidad de los botones es la misma y ahora el filtro es para filtrar por el nombre de los proveedores.

Si pulsamos sobre el botón de nuevo y rellenamos los campos obtenemos la siguiente ventana:

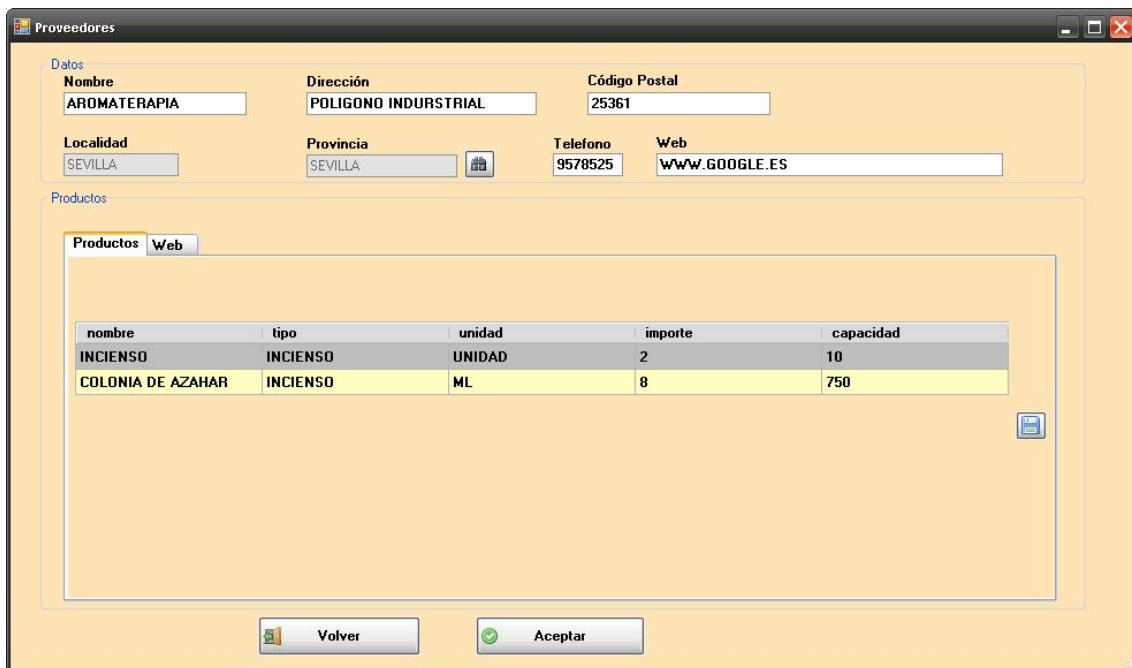


Fig112. Ventana Nuevo Proveedor

Al principio el botón de productos con icono de disket en la pestaña de productos se encuentra deshabilitado. Cuando rellenemos los campos de la parte superior y pulsemos sobre el botón de Aceptar nos aparecerá un mensaje de confirmación para añadir productos. Cada vez que le demos al botón de añadir producto nos aparecerá esta ventana:

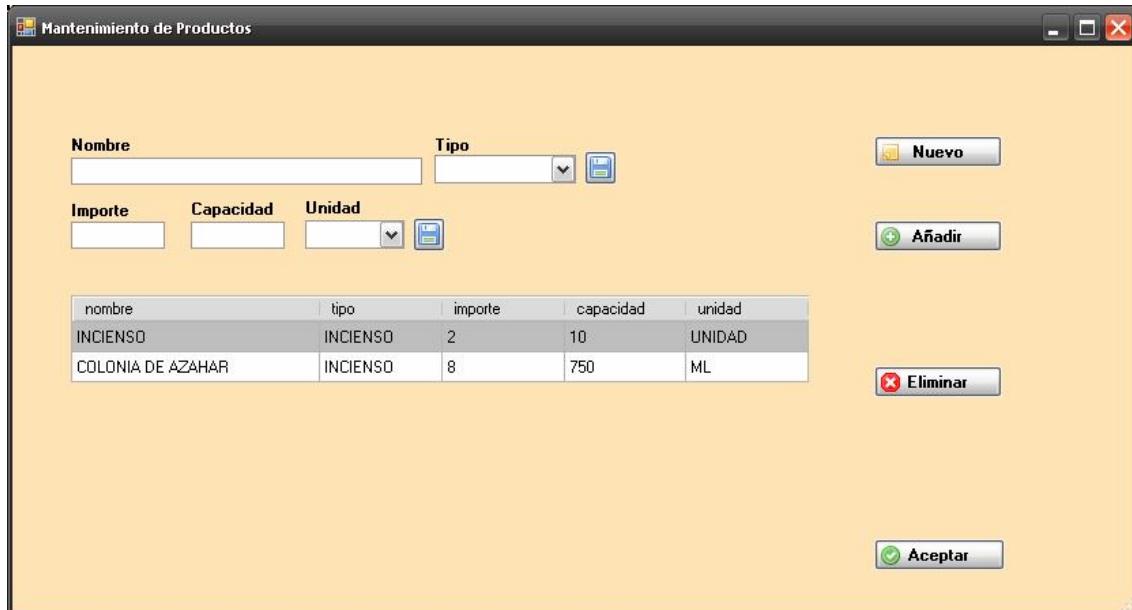


Fig113. Ventana Producto

Esta ventana servirá para añadir un producto al proveedor. Rellenamos los campos y le damos aceptar. Si la unidad o el tipo deseado no se encontraran en el combo, siempre podemos añadir unos nuevos como siempre en el botón de disket. Las ventanas correspondientes que nos saldrían serían:

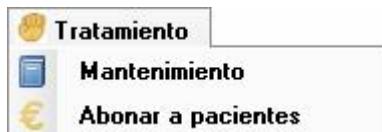




Fig114. Ventana Mantenimiento Unidad y Tipo

Para terminar con los proveedores, la ventana dispone de otra pestaña llamada Web. En esta ventana se muestra un navegador con la página que contenga la caja de texto Web. Esto servirá de ayuda para disponer en la propia aplicación la página Web del proveedor.

5.3.5 Tratamiento



Si pulsamos sobre esta pestaña, se nos abre dos opciones. La primera es el mantenimiento de los tratamientos mientras que la segunda es el método para asociar el bono de un tratamiento a un paciente. Veamos la primera, la ventana que se abre es la siguiente:



Fig115. Ventana Tratamiento

Esta ventana sigue la misma funcionalidad que la de buscar paciente o la de buscar proveedor. Pasamos directamente a la ventana que surge de pulsar el botón Nuevo.

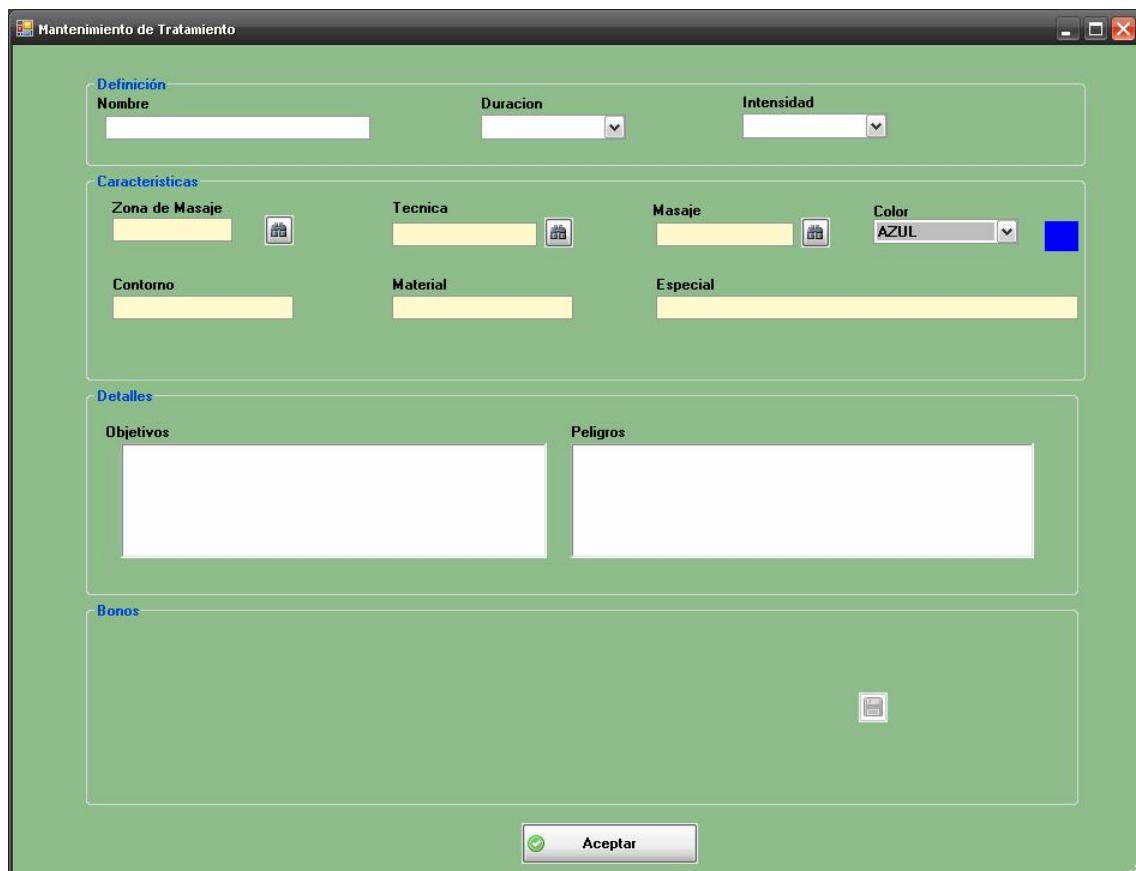


Fig116. Ventana Nuevo Tratamiento

Como podemos ver la ventana se compone de cuatro fases.

En la primera fase tenemos la definición del tratamiento. Para ello rellenamos la caja de texto y elegimos los valores para los combobox.

En la segunda fase tenemos las características. Para insertar las características pulsaremos los botones y elegiremos los valores que nos interesen pulsando sobre el botón de volver. Si los valores no estuviesen, pulsamos sobre el botón del disket y añadimos la información. Mostramos las seis ventanas que pueden suceder, dos por cada combo, una ventana de búsqueda y otra de mantenimiento.

Ventana de búsqueda para la zona:



Fig117. Ventana Buscar Zona.

Ventana de mantenimiento para la zona:



Fig118. Ventana Mantenimiento Zona.

Ventana de búsqueda para la técnica:



Fig119. Ventana Buscar Técnica

Ventana de mantenimiento para la técnica:



Fig120. Ventana Mantenimiento Técnica

Ventana de búsqueda para el masaje:



Fig121. Ventana Buscar Masaje

Ventana de mantenimiento para el masaje:

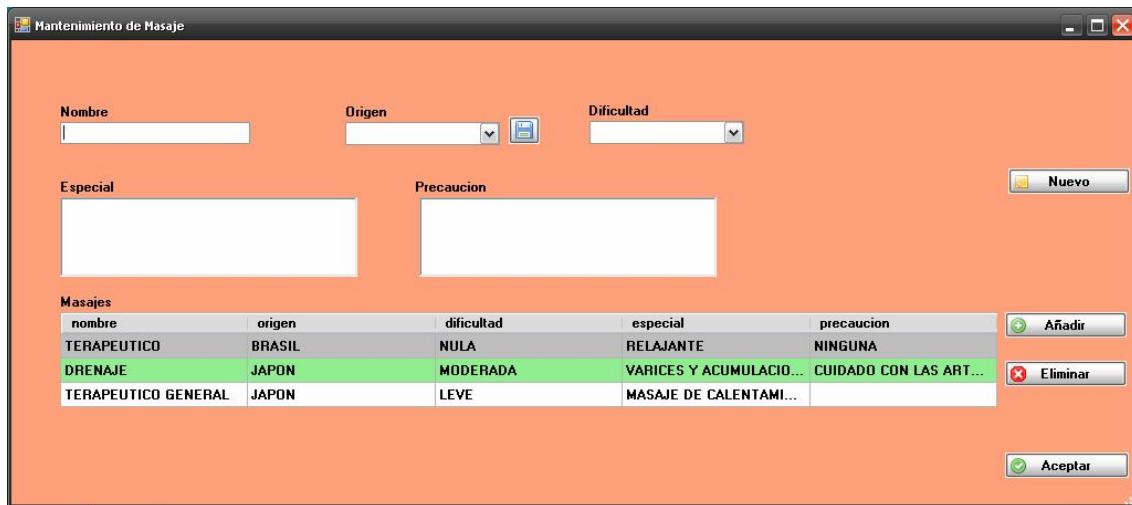


Fig122. Ventana Mantenimiento Masaje

En la tercera fase tenemos la opción de comentar algunos de los detalles del tratamiento.

En principio la cuarta fase esta desactivada, ya que aún no existe el tratamiento. Si pulsamos el botón de Aceptar, se nos mostrará un mensaje para añadir bonos:

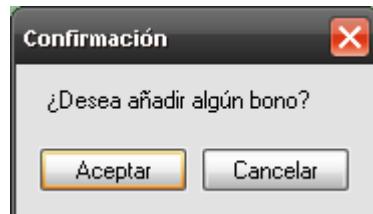


Fig123. Ventana Confirmación

Si pulsamos el botón para añadir bonos, se nos abrirá una ventana como la siguiente. La funcionalidad de esta ventana es idéntica a las ventanas de mantenimiento.



Fig124. Ventana Bonos

En cambio, si decidimos pulsar la pestaña de Abonar pacientes se mostrará la siguiente ventana:

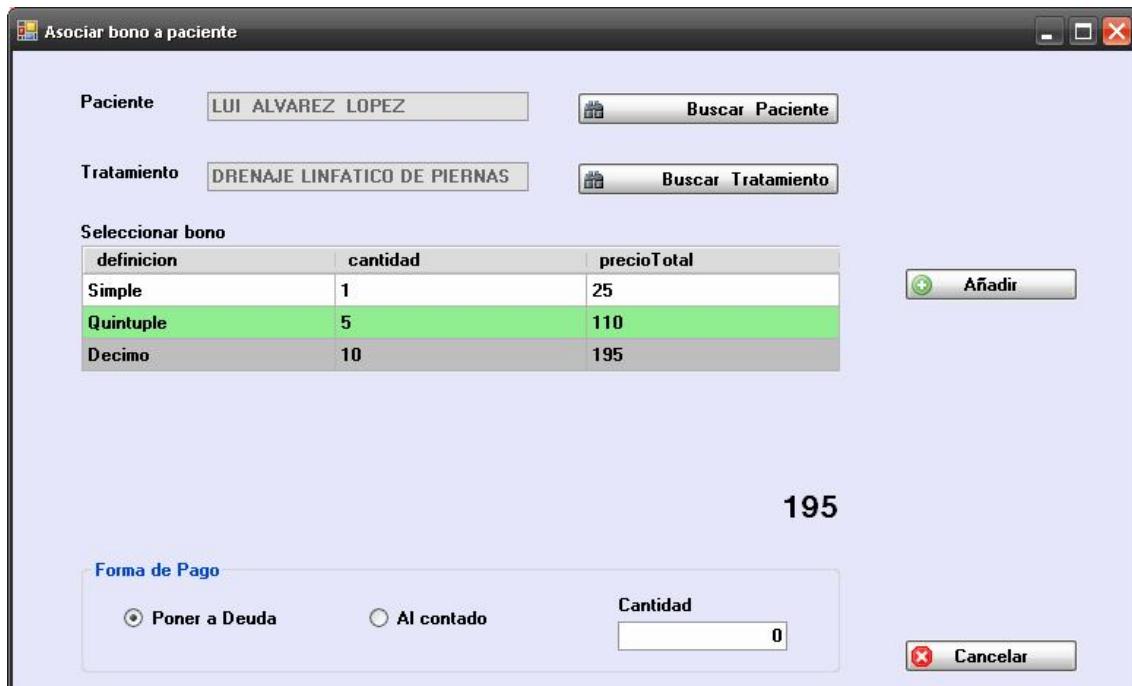
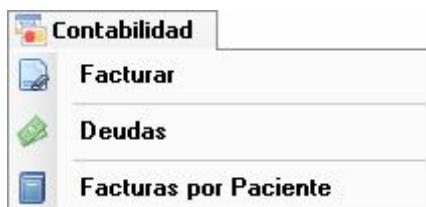


Fig125. Ventana Asociar Bono

En esta ventana hemos rellenado los campos de paciente y tratamiento para

explicar mejor el funcionamiento. Si pulsamos sobre el botón de buscar paciente se nos abre la ventana de buscar paciente comentada anteriormente. Elegimos el paciente deseado y al pulsar sobre el botón de volver se muestra la información en la caja de texto. Si pulsamos sobre el botón de buscar tratamiento ocurre lo mismo, pero con tratamiento. Al volver de buscar el tratamiento se muestran los bonos que dispone el tratamiento. Hasta que no pulsemos sobre un bono y busquemos un paciente, es decir, hasta que no tengamos los datos del paciente y los del bono, el botón de Añadir estará deshabilitado. En la parte inferior de la ventana tenemos dos botones de selección en forma de radio. Estos botones son para seleccionar si la cuantía la abonará el paciente o se queda por pagar. Por defecto aparecerá pendiente de pago. Por el contrario si el paciente va a abonar parte o total de la cuantía, seleccionamos el pago y ponemos la cuantía a entregar. Tras darle a añadir se realizan las operaciones de asociar el bono al paciente y añadimos en caja la entrada de dinero si la hubiera. De forma indirecta tendremos guardados los pagos del paciente y sus deudas.

5.3.6 Contabilidad



En la contabilidad repartiremos las funciones en Facturar para crear facturas a pacientes, las deudas que existen entre los pacientes y la consulta y también las deudas que existen entre la consulta y los proveedores y por último podremos ver las facturas por pacientes. Veamos cada apartado.

Si pulsamos sobre Facturar se muestra la siguiente ventana.

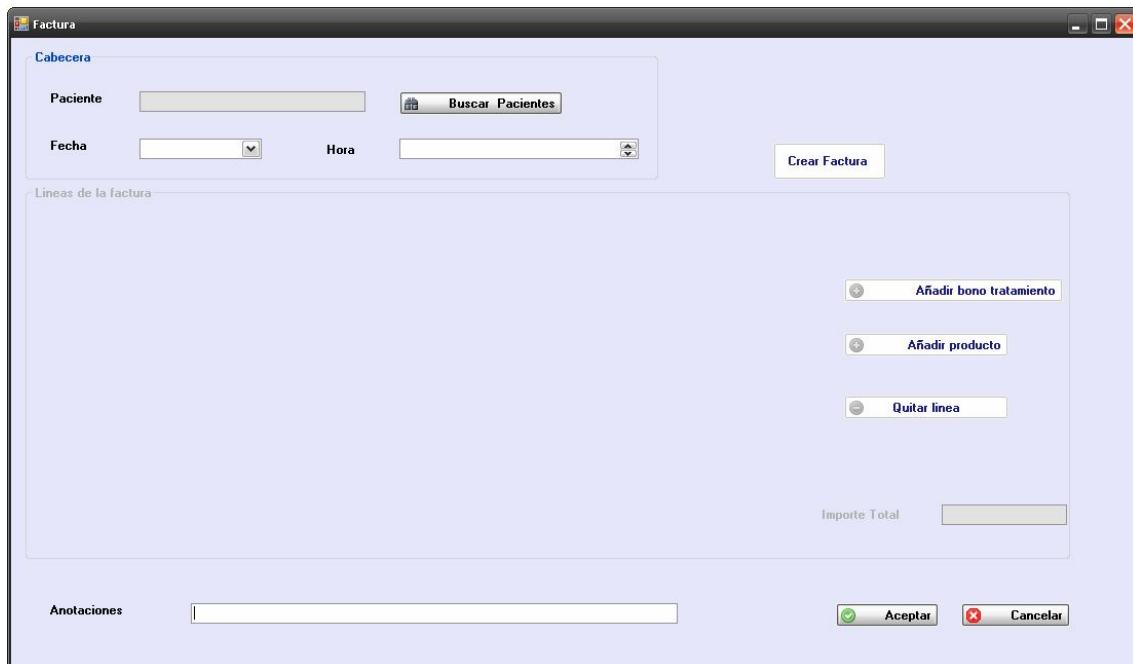


Fig126. Ventana Factura

Para realizar una factura hay que seguir los siguientes pasos.

En primer lugar tenemos que buscar al paciente. Para ello pulsamos el botón de buscar paciente, seleccionamos el paciente en la ventana de búsqueda y al pulsar volver se mostrará la información. Cuando tengamos el paciente seleccionado, se habilitará el botón de crear factura. Al pulsar sobre este botón tendremos acceso para añadir líneas a la caja. Estas líneas se basan en insertar productos o bonos o eliminar las líneas. A continuación se detalla como insertar las líneas de caja.

Para insertar un bono pulsaremos sobre el botón de insertar bono. La ventana que se muestra es la siguiente:

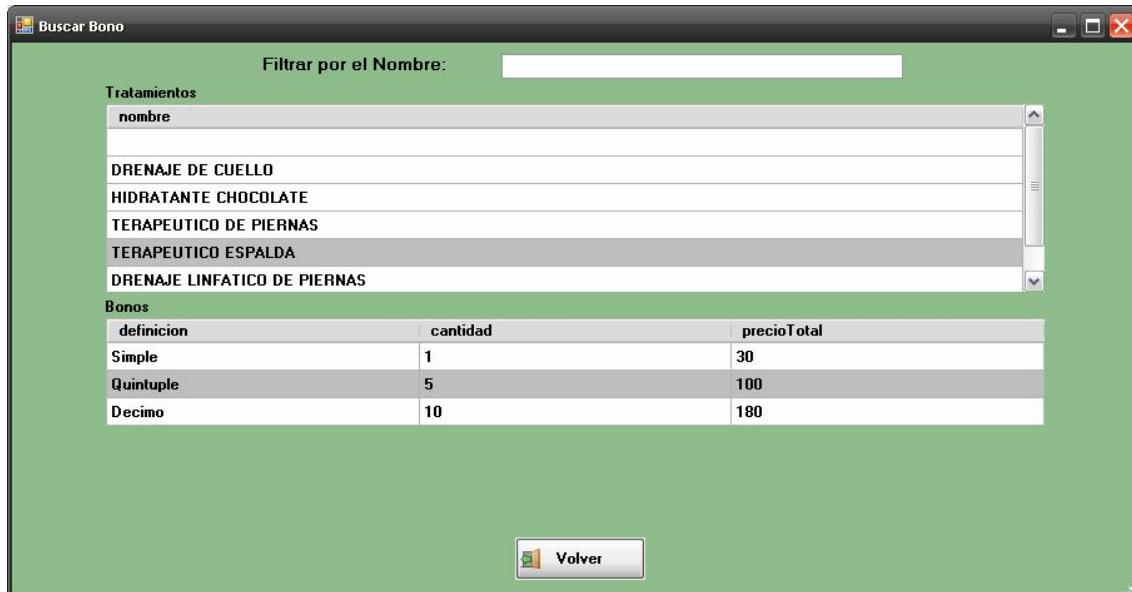


Fig127. Ventana Buscar Bono

En esta ventana tenemos dos tablas y el botón volver que ya sabemos su funcionamiento. En la tabla superior se muestra los tratamientos que tenemos almacenados. En la tabla inferior se mostrarán los bonos del tratamiento que tengamos seleccionado. Para seleccionar un bono, primero seleccionaremos el tratamiento y luego el bono, pulsando volver añadimos el bono a la caja. La ventana tiene también un filtro para la tabla de tratamiento.

Para insertar un producto pulsaremos sobre el botón de insertar producto. La ventana que se muestra es la siguiente:

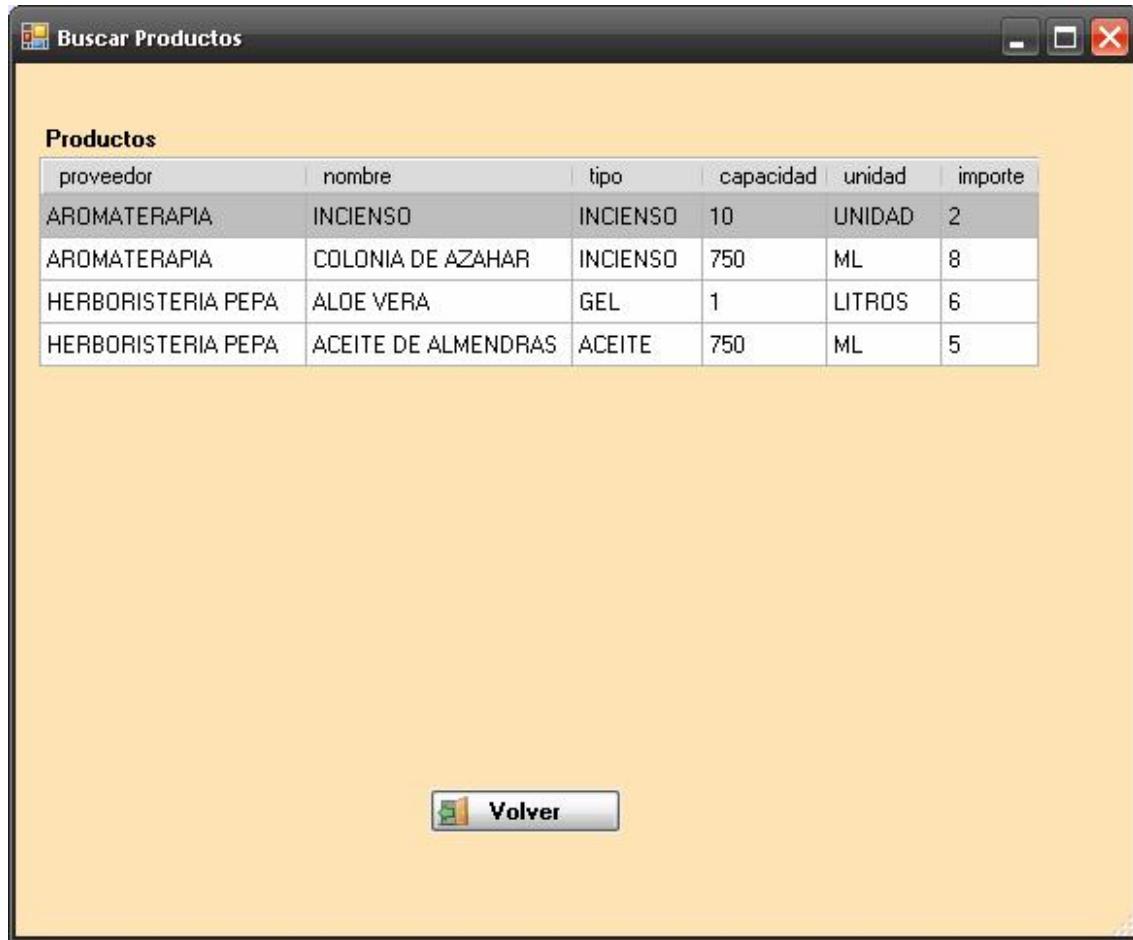


Fig128. Ventana Buscar Producto

En esta ventana se muestran todos los productos que hay en el almacén. Seleccionaremos el producto y al pulsar sobre el botón volver añadimos el producto a la caja.

Si por cualquier de error, equivocación u otros motivos, deseamos eliminar una línea de caja, seleccionaremos la línea a eliminar y pulsaremos el botón de eliminar línea de caja.

La factura tiene un campo de anotaciones, este se encuentra ubicado en la parte inferior en forma de caja de texto.

Para terminar con la factura, pulsaremos el botón de aceptar para guardar los cambios o cancelar para anular la factura.

Si pulsamos sobre Deudas se muestra la siguiente ventana.



Fig129. Ventana Deudas

En esta ventana se pueden ver y actuar sobre las deudas de los pacientes y sobre los proveedores.

La ventana está compuesta por dos pestañas llamadas pacientes y proveedores, las dos pestañas son iguales pero hemos separado el concepto, debido a que los pacientes son los que deben el dinero y los proveedores son a los que le debemos el dinero.

En la pestaña tenemos una tabla con la información de la deuda, una caja de texto en la parte inferior derecha que nos indica el total de deudas de la tabla, y un botón, situado al inferior de la tabla parte izquierda, llamado añadir pago. Este botón es para anular la deuda o parte de ella. Si pulsamos sobre este botón se nos abre una ventana con la información de la deuda y una caja de texto para indicar la cantidad que se desea añadir. Al pulsar aceptar volvemos a la ventana de deudas con las tablas actualizadas.

Por última opción podemos pulsar sobre la pestaña de facturas por paciente. La ventana que se muestra es la siguiente:

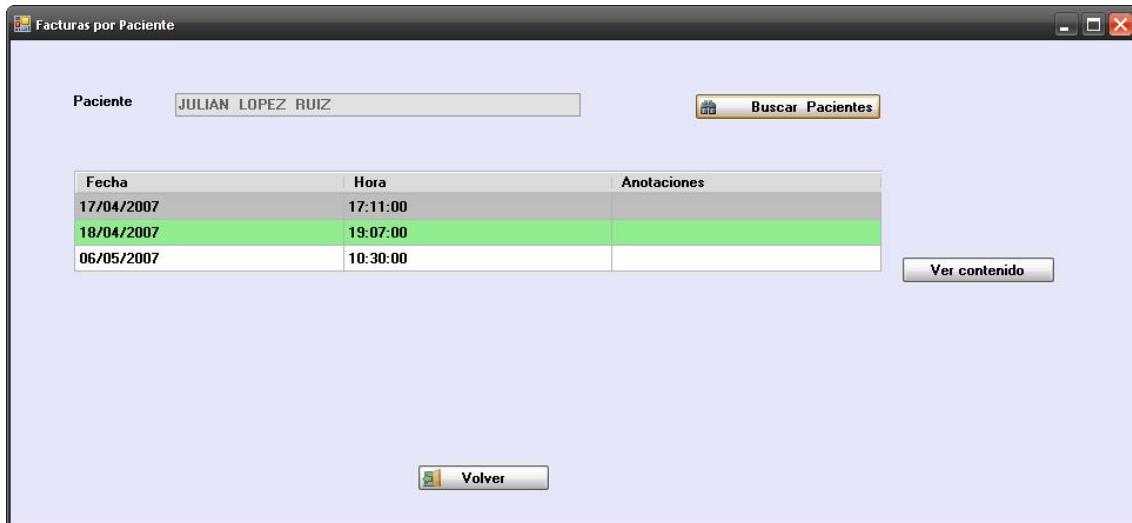


Fig130. Ventana Facturar Paciente.

Lo primero que hacemos en la ventana es buscar el paciente de la misma forma que se viene explicando en el manual. Después de seleccionar el paciente, se mostrará en la tabla todas las facturas que posee el paciente. Si seleccionamos una factura de la tabla podemos pulsar el botón de ver contenido para consultar los datos que contiene. Además de ver el contenido aparece un botón para imprimir la factura.

5.3.7 Caja



Para acceder a la caja de la empresa, solo tenemos que pulsar sobre este ícono. La ventana que resulta de pulsar se muestra a continuación.

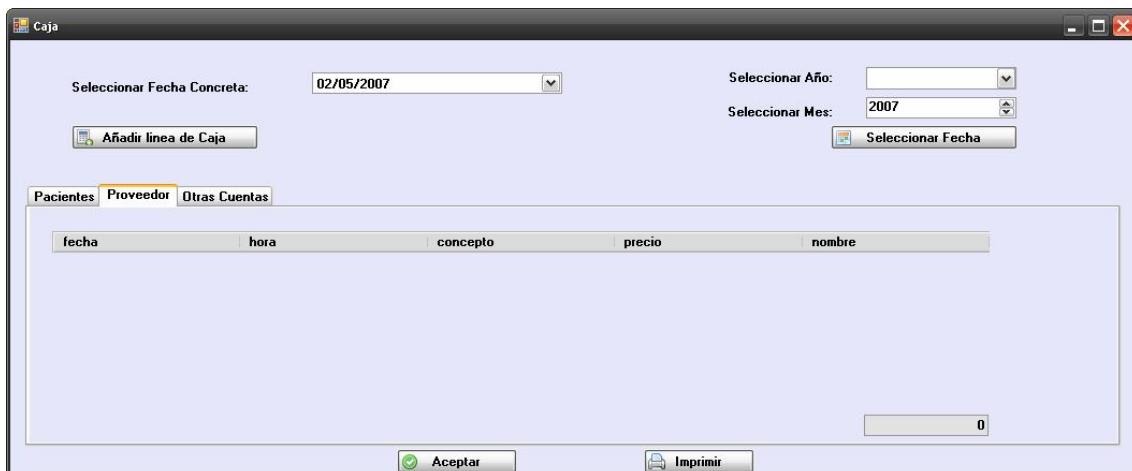


Fig131. Ventana Caja.

Cuando se abre la ventana por primera vez, en las tablas que se comentaran en los siguientes párrafos, estará establecida por la fecha actual.

La ventana se divide en dos fases horizontales.

La primera fase horizontal, parte superior, es para seleccionar la fecha. En la parte izquierda podemos establecer las fechas seleccionadas por días. En la parte derecha podemos seleccionar un mes en concreto.

Cada vez que seleccionemos la fecha que deseemos los resultados se mostrarán en la parte inferior de la ventana.

La segunda fase horizontal tiene una característica muy especial. Los resultados están separados en tres apartados, que son paciente, proveedores y otros. Para ver los datos que queramos, solo tenemos que picar sobre la pestaña deseada. En la parte inferior de la tabla aparece en una caja de texto el total de las filas que se muestran.

Además de lo comentado, podemos localizar en el centro a la izquierda un botón llamado añadir línea de caja. Al pulsar sobre este botón se muestra la siguiente ventana que pasamos a detallar su funcionamiento.

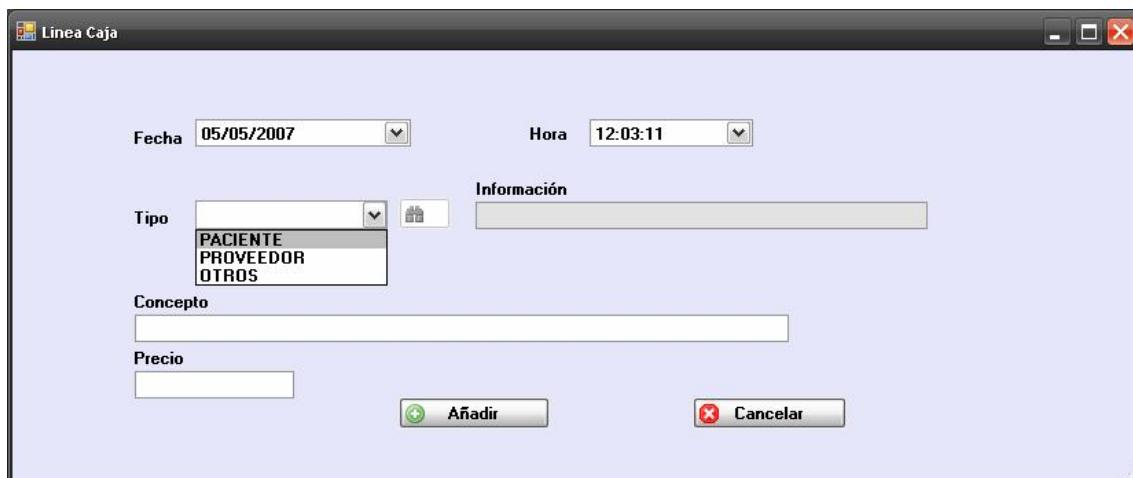


Fig131. Ventana Línea Caja

Al principio de la ventana se muestran las fechas actuales, pero estas se pueden

modificar por si el concepto de añadir a la caja fuese de otra ocasión.

En el combo de Tipo, se podrá elegir el origen del concepto. Este origen podrá ser un paciente, un proveedor u otros conceptos. En la derecha del combo Tipo, aparece un botón de búsqueda. Este botón estará inactivo hasta que no este seleccionado un origen del combo.

Cuando pulsamos el botón, en el caso de que el origen sea paciente o proveedor, se abrirá la ventana de búsqueda para seleccionar el origen. En la caja de texto que aparece a la derecha del botón, se llenará automáticamente con los datos del origen. En cambio si el origen es otros datos, no tendremos que realizar ninguna búsqueda. Terminada la selección del origen solo tenemos que llenar el concepto de la línea de caja y la cantidad.

Nota1: Si la cantidad es negativa tendremos que indicarlo.

Nota2: Estas cantidades que se añaden, en el caso de pacientes o proveedores, no descuentan las deudas pendientes. Para anular las deudas del paciente, tendremos que seleccionar el paciente y en bonos añadir pago. Para el caso de los proveedores en el apartado de deudas de contabilidad. En cambio estas líneas de caja tiene la utilidad para indicar datos económicos extras como propinas, gastos internos, venta de productos a pacientes etc.

5.3.8Base de Datos



En este apartado podremos realizar la gestión de seguridad de la base de datos.

El primer caso es Exportar. Aquí se realiza la creación de los ficheros necesarios para guardar la información de la base de datos. En el caso de que se repita esta operación de forma consecutiva, se reemplazarán los ficheros existentes por los actuales. El sentido de la información viaja desde la aplicación hacia los ficheros.

El segundo caso es Importar. En este caso se aborda la recuperación de la base de datos. Para ello se recogerá la información existente de los ficheros y se reemplazará por el contenido actual de la base de datos. Ahora el sentido de la información viaja desde los ficheros hacia la aplicación.

Antes de realizar las operaciones de la exportación o importación sobre la base de datos o la aplicación se mostrará un mensaje de confirmación. Si la respuesta es de la confirmación es afirmativa se procederá con la operación.

!! ¡ADVERTENCIA!! !

A pesar de los mensajes de confirmación, si se realiza por primera vez la importación sin haber hecho antes la exportación, como no existe información almacenada en los ficheros, el resultado será el borrado total de la aplicación.

5.3.9 Almacén



Pulsaremos sobre esta pestaña del menú principal para acceder al almacén. La ventana que se muestra es la siguiente:

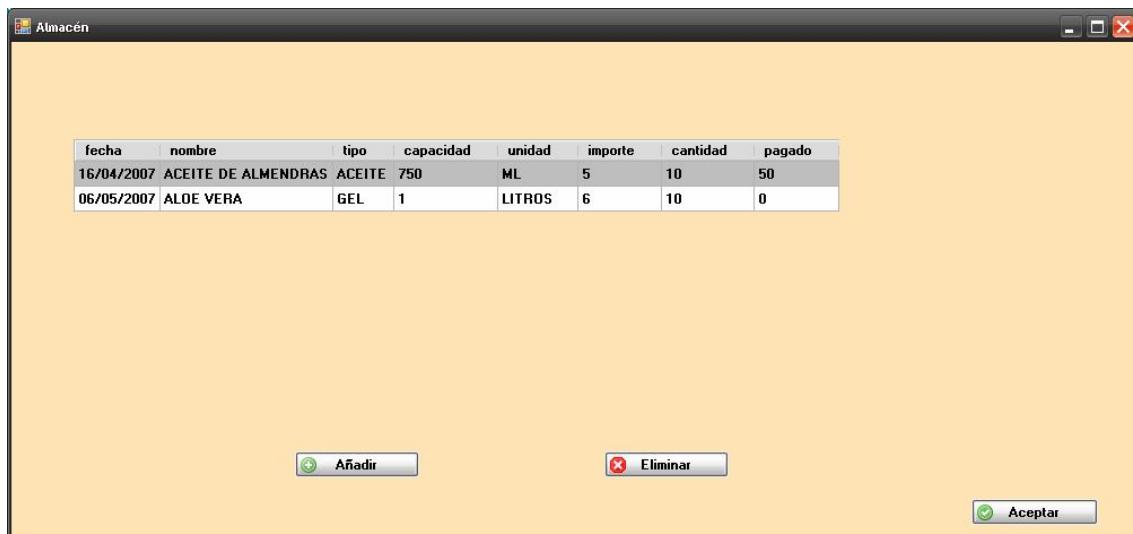


Fig132. Ventana Almacén

En la ventana se muestra una tabla con la información del producto y la fecha de entrada. Además se muestra la cantidad existente de cada producto.

Debajo de la tabla hay dos botones, uno para añadir producto y otro para eliminar.

Si pulsamos el botón de añadir producto, se muestra la ventana de todos los productos que existen. La ventana es la siguiente:

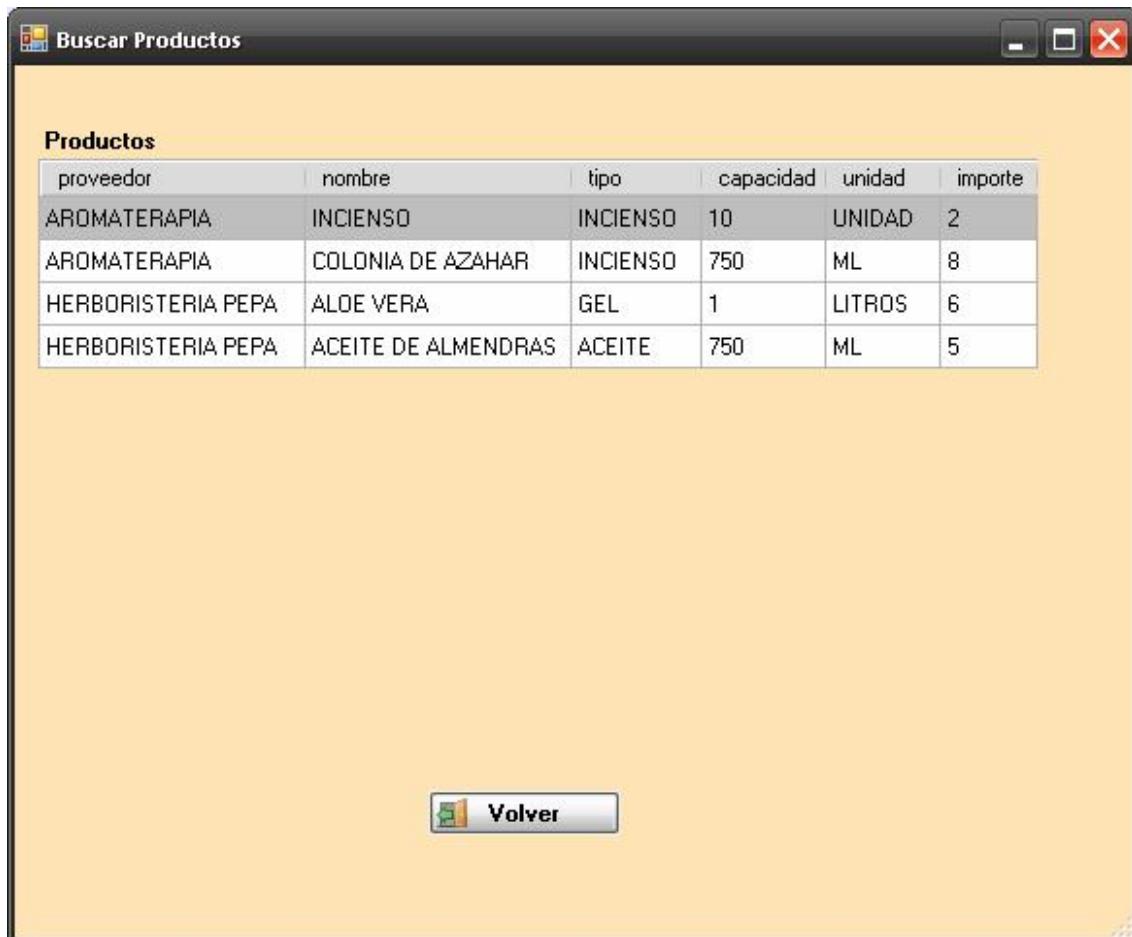


Fig133. Ventana Buscar Producto

Seleccionamos el producto y pulsamos sobre el botón volver. Ahora se muestra una ventana simple que pregunta por la cantidad que deseamos añadir al almacén y si lo vamos a pagar o poner a deuda. Rellenamos las condiciones y pulsamos aceptar. La ventana de la que hablamos es esta:

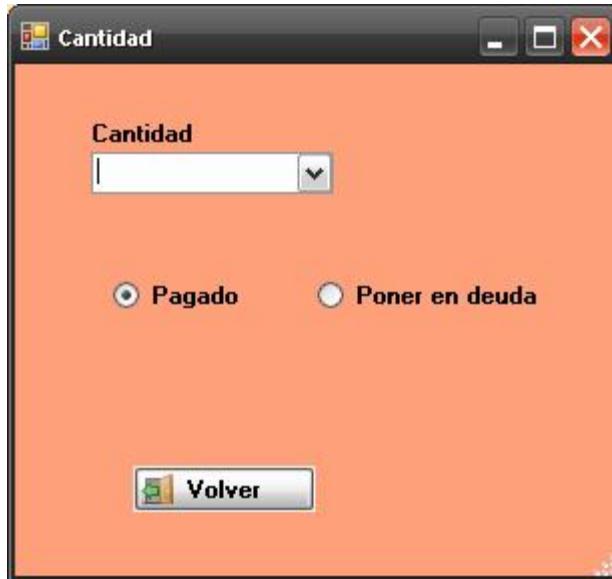
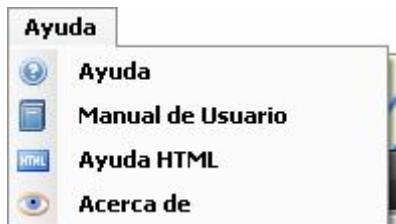


Fig134. Ventana Cantidad

Con el botón de eliminar podremos eliminar un producto que se haya incluido por error u otras circunstancias.

5.3.10 Ayuda



Si pulsamos sobre ayuda podremos ver cuatro contenidos.

El primero abrirá el archivo de ayuda. En este se podrá consultar la ayuda y navegarnos a través del índice.

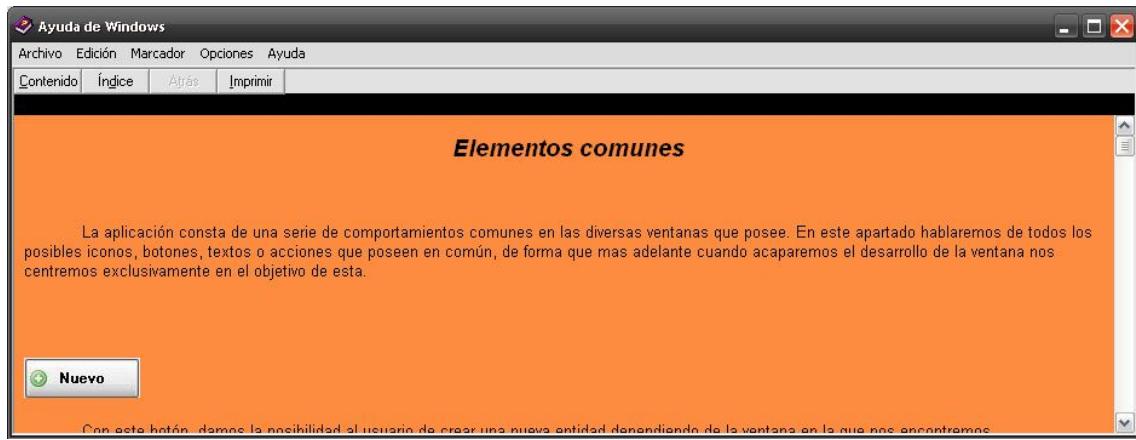


Fig135. Ventana Ayuda

El segundo abrirá el manual de usuario en formato pdf.

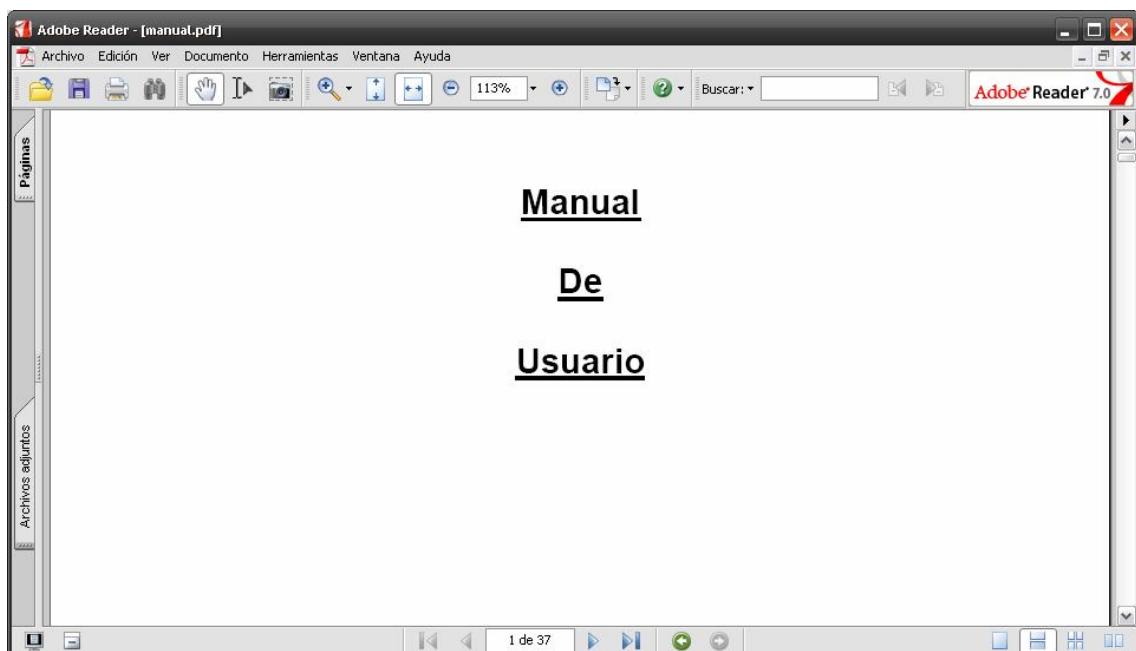


Fig136. Ventana Ayuda en PDF.

En el tercero podremos consultar la ayuda por medio de la página Web.

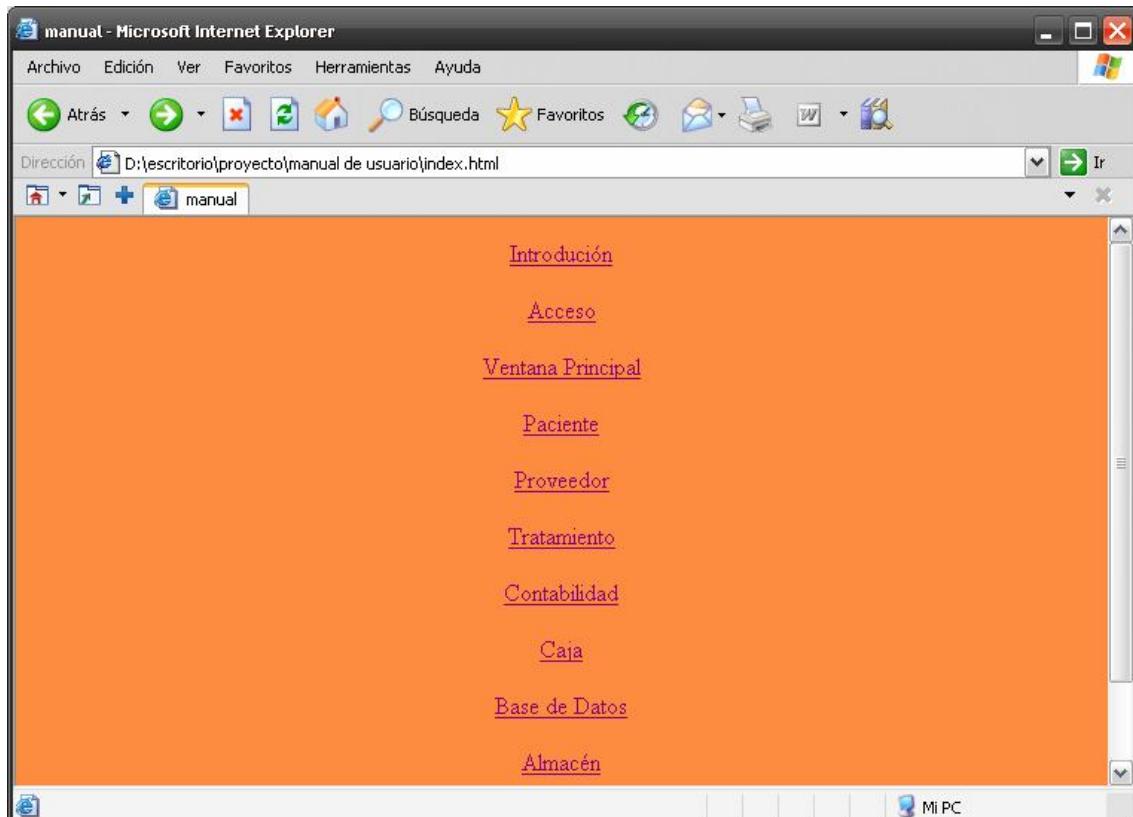


Fig137. Ventana Ayuda En Web

En el Acerca de se podrá ver información del desarrollo de la aplicación.



Fig138. Ventana Acerca de.

Capítulo 6: Conclusiones.

6.1 Problemas encontrados durante el desarrollo del proyecto.

El mayor problema que tenemos al realizar una aplicación desarrollada bajo el compilador de C# y utilizando Mysql como gestor de base de datos es que entre ellos no existe en principio ninguna conexión. Desde la web oficial de mysql, nos podemos descargar una librería que nos comunica ambas cosas. Una vez agregada la librería al proyecto solo debemos de crear una clase utilizando toda la potencia de la librería para crear métodos como insertar valores en una tabla, rescatar desde un ‘select’ y así con todos los métodos necesarios. A modo de ejemplo hemos desarrollado un método que dándole un ‘textbox’ y un ‘select’ nos pinta la caja de texto con el resultado obtenido.

Otro problema que tuvimos fue en la impresión de documentos. La forma más fácil de realizar esta tarea es mediante una utilidad que posee Visual Studio llamada Crystal Reports. Esta opción es desechada por ser de pago. La solución que se dio fue crear una clase que imprimiera el formulario en el que nos encontrábamos, esta forma es de una dificultad elevada, pero Visual Studio carece de esta opción.

Un problema que en principio no contábamos es que la poca experiencia de nuestro usuario con los programas informáticos, provocó un conjunto de dudas al no saber claro como quería que fuera el producto final. El realizar el proyecto incremental y enseñarle varios prototipos de la posible aplicación nos sirvió, por un lado para llenar de capacidades nuevas a nuestro cliente y por otro lado para concretar el objetivo final.

6.2 Conclusiones finales.

Terminada la construcción de la aplicación, hay que ver las posibilidades que se abren para la utilización continua de la aplicación.

La aplicación se ha realizado mediante diálogos. Este objetivo ha permitido que el resultado final de la aplicación sea una interfaz muy sencilla, las ventanas poseen toda la información necesaria y los botones están colocados de una forma que se adapta al clic más corto, es decir, que los botones más usados son los primeros en hacer clic. La navegación entre ventanas es trivial y se van abriendo las ventanas según los datos que necesitemos para las operaciones.

Otro objetivo que se buscaba, y era uno de los motivos por lo que surge el proyecto, era asesorar de forma eficiente la consulta, ya sea la agenda, control de pacientes, caja o las distintas funciones. Este objetivo se ha cumplido con creces y ha permitido agilizar las tareas cotidianas de la consulta. Ahora la consulta posee una base de datos eficiente con una aplicación para gestionar la base de datos.

6.3 Posibles mejoras del sistema de información.

Realizar en la primera versión un software sin ningún fallo y con la mayor completitud y el mayor número de funciones posibles es difícil. De hecho, en el transcurso del desarrollo del proyecto, han ido saliendo nuevas ideas que aportarían a la aplicación un grado superior de ayuda al usuario. Estas nuevas ideas son las siguientes.

- Una nueva pestaña en las observaciones de los pacientes que nos indicará de forma gráfica el estado físico. Esta pestaña contendría una imagen del cuerpo humano y mediante colores se podría observar que zona está dañada y cual está sana.
- Otra utilidad para la publicidad, seria una función que permitiera una campaña de mailing. Como conocemos los correos electrónicos de los pacientes, se podría mandar publicidad, ofertas o estado de deudas con el paciente.

- Otra utilidad sería enlazar la aplicación a una página Web y permitir que los usuarios pudieran pedir cita vía Internet, y poder administrar ciertas funciones de la aplicación.
- Sería de gran utilidad poder elegir si imprimir las facturas de la forma actual o bien unificadas en una misma hoja. Actualmente está pensado para almacenar en carpetas las facturas, y puede ser que al paciente le interese más la otra versión.

Capítulo 7: Bibliografía

Visual C#:

Francisco Charte .Visual C#.Net. Anaya. 2002

Jeff Ferguson. La Biblia de C#. Anaya. 2002

José Antonio González Seco. El lenguaje de programación C#. 2001

Manuales recopilados de Internet para el manejo y configuración de MySQL con Visual C#.

Magic Draw:

www.magicdraw.com/

Métrica 3:

Página oficial de Métrica:

<http://www.csi.map.es/csi/metrica3/index.html>.

Foro oficial de Métrica:

<http://foro.map.es/>

Manual y Técnicas de utilización de Métrica3:

<http://www.desarrollos-mecame.com/formacion/Ingenieria-Software/Ingenieria%20software%20-temario.htm>

Marcela Genero, Mario Piattini, Coral Calero. Metrics for software conceptual models, Imperial College Press, cop. 2005.

Ingeniería del Software:

Cabrera Sánchez, Gregorio. Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. Ra-Ma.2003.

G. Booch, J. Rumbaugh, I. Jacobson El lenguaje Unificado de Modelado. Addison Wesley, 1999.

Antonio de Amescua Seco. Análisis y diseño estructurado y orientado a objetos de sistemas informáticos. McGraw-Hill, D.L. 2003.

Mario G.Piattini Velthuis. Análisis y diseño de aplicaciones informáticas de gestión: una perspectiva de ingeniería del software. Madrid: RA-MA, cop. 2004

Sara Mª García Blanco, Enrique Morales Ramos. Thomson: Paraninfo Análisis y diseño detallado de aplicaciones informáticas de gestión: desarrollo de aplicaciones informáticas. , D.L. 2003

Base de datos:

H. Korth, A. Silverschartz, S. Sudarshan. Fundamentos de bases de datos. 4ª Edición McGraw-Hill. 2002.

Ramez A. Elmasri. Fundamentos de Sistemas de Bases de Datos: Addison-Wesley 3ª Edición. 2004

Links interesantes:

Programación en Castellano: <http://programacion.com>

La Web del programador: <http://www.lawebdelprogramador.com/>

C# Help: <http://www.csharpHelp.com/>

Clases de ayuda: <http://www.codeproject.com/>

Sitio oficial de Aqua Data Studio: <http://www.aquafold.com/>

Curso MySql: <http://mysql.conclase.net/curso/index.php>

Foro y clases para C#: <http://www.csharpfr.com/Default.aspx>

<http://foros.solocodigo.com/index.php>

API para C#: [http://msdn2.microsoft.com/es-es/library/ms306608\(vs.80\).aspx](http://msdn2.microsoft.com/es-es/library/ms306608(vs.80).aspx)