

# **DermaSystem**

## **Gestión Informática en Dermatología**

**Sara Llorente Navarro**

**Carlos Manuel Muñoz Díaz**

**José Carlos Olivar Conde**

**Proyecto de la Facultad de Informática**

**Departamento de Ingeniería del Software e Inteligencia Artificial**

**Universidad Complutense de Madrid**



**Trabajo Fin de Grado en Ingeniería del Software**

**Curso 2012-2013**

Director: Gonzalo Pajares Martinsanz

Colaboradora: María Guijarro Mata-García

Colaboradora externa: Isabel Longo Imedio

## Índice

<b>AUTORIZACIÓN</b>	<b>I</b>
<b>RESUMEN</b>	<b>II</b>
<b>PALABRAS CLAVE</b>	<b>II</b>
<b>ABSTRACT</b>	<b>III</b>
<b>KEYWORDS</b>	<b>III</b>
<b>1 INTRODUCCIÓN</b>	<b>4</b>
<b>2 PLANTEAMIENTO</b>	<b>6</b>
<b>3 ORGANIZACIÓN DEL TRABAJO</b>	<b>7</b>
<b>4 ANÁLISIS DE LA SITUACIÓN ACTUAL</b>	<b>8</b>
<b>5 OBJETIVOS</b>	<b>11</b>
<b>6 ESPECIFICACIÓN DE REQUISITOS</b>	<b>12</b>
6.1 SISTEMA DE MARCADO	13
6.2 CONTROL DEL SISTEMA DE MARCADO	13
6.3 BASE DE DATOS	13
6.4 AUTENTICACIÓN DE USUARIOS	14
<b>7 DIAGRAMA DE CASOS DE USO</b>	<b>15</b>
<b>8 ESPECIFICACIÓN DE CASOS DE USO</b>	<b>17</b>
8.1 SUBSISTEMA USUARIOS	18
8.2 SUBSISTEMA PACIENTES	21
<b>9 DISEÑO</b>	<b>26</b>
9.1 PATRÓN MODELO VISTA CONTROLADOR	26
9.2 PATRÓN DAO	29
9.3 PATRÓN FACTORÍA ABSTRACTA	29
9.4 PATRÓN SINGLETON	30
9.5 PATRÓN OBSERVER	31
9.6 VARIOS SISTEMAS OPERATIVOS	32
9.7 BASE DE DATOS	33
9.7.1 CONSIDERACIONES PREVIAS	33
9.7.2 ELECCIÓN DE LA BASE DE DATOS	33
9.7.3 DISEÑO DE LA BASE DE DATOS	34
9.7.4 HOSPEDAR LA BASE DE DATOS	35

<b>10</b>	<b>DIAGRAMAS DE SECUENCIA</b>	<b>36</b>
10.1	DIAGRAMA AÑADIR MARCA	36
10.2	DIAGRAMA EDITAR PACIENTE	36
<b>11</b>	<b>DIAGRAMA DE CLASES</b>	<b>39</b>
<b>12</b>	<b>TECNOLOGÍAS UTILIZADAS</b>	<b>40</b>
12.1	JAVA	40
12.2	JAVA SWING	40
12.3	JCALENDAR	40
12.4	HIBERNATE	41
12.5	ASSEMBLA	42
12.6	IDES	42
12.7	SUBCLIPSE	42
12.8	ITEXT	43
<b>13</b>	<b>IMPLEMENTACIÓN</b>	<b>43</b>
13.1	INTERNACIONALIZACIÓN (I18N)	43
13.2	CONFIGURACIÓN DE LA APLICACIÓN	45
13.3	IMPRESIÓN DE MARCAS PDF	46
13.4	INTERFACES	46
13.4.1	APPLICATIONVIEW	46
13.4.2	USERVIEW	47
13.4.3	PATIENTVIEW	47
<b>14</b>	<b>CONCLUSIONES Y FUTURO</b>	<b>50</b>
14.1	CONCLUSIONES	50
14.2	FUTURO DE LA APLICACIÓN	50
<b>15</b>	<b>GLOSARIO</b>	<b>52</b>
<b>16</b>	<b>BIBLIOGRAFÍA</b>	<b>53</b>
	<b>APÉNDICE</b>	<b>54</b>
A1.	ACCESO A LA APLICACIÓN	54
A2.	CAMBIO DE IDIOMA	54
A3.	INFORMACIÓN Y AYUDA	55
A4.	MI CUENTA	55
A5.	EDITAR MI CUENTA	55
A6.	MANUAL DE USUARIO: GESTIÓN DE USUARIOS	56
1)	GESTOR DE USUARIOS	56
2)	BÚSQUEDA DE USUARIOS	57
3)	AÑADIR UN NUEVO USUARIO	57
4)	EDITAR UN USUARIO	58
5)	ELIMINAR UN USUARIO	58
A7.	MANUAL DE USUARIO: GESTIÓN DE PACIENTES	60

1) GESTOR DE PACIENTES	60
2) BÚSQUEDA DE PACIENTES	61
3) EDITAR UN PACIENTE	61
4) ELIMINAR UN PACIENTE	62
<b>A8. MANUAL DE USUARIO: MARCAS</b>	<b>63</b>
1) SELECCIONAR UN ÁREA	63
2) REGISTRAR UNA MARCA	64
3) BORRAR MARCAS	68
4) IMPRIMIR EL ÁREA DE UN PACIENTE	70

## Índice de figuras de la memoria

Figura 1: Arquitectura general del sistema.....	10
Figura 2: Diagrama del módulo usuarios (administrador y usuario normal) .....	15
Figura 3: Diagrama del módulo usuarios en relación a pacientes.....	16
Figura 4: Esquema general del Modelo Vista Controlador .....	27
Figura 5: MVC en DermaSystem .....	28
Figura 6: Esquema general del patrón DAO dentro del modelo .....	29
Figura 7: Patrón Singleton .....	31
Figura 8: Declaración de la clase Model .....	31
Figura 9: Comportamiento de las Vistas.....	32
Figura 10: "Suscripción" de las vistas. ....	32
Figura 11: Base de Datos: diseño y esquema de las tablas .....	35
Figura 12: Diagrama de secuencia añadir marca.....	37
Figura 13: Diagrama de secuencia editar paciente .....	38
Figura 14: Diagrama de clases .....	39
Figura 15: Clase AppLanguage .....	44
Figura 16: Contenido del archivo de idioma.....	45
Figura 17: Diagrama de vistas.....	49

## Índice de figuras del apéndice

Figura A1: Ventana principal de DermaSystem .....	54
Figura A2: Ventana principal de gestión de usuarios .....	56
Figura A3: Añadir nuevo usuario .....	57
Figura A4: Edición de usuario .....	58
Figura A5: Eliminar usuario.....	59
Figura A6: Ventana principal de gestión de pacientes .....	60
Figura A7: Información del paciente.....	61
Figura A8: Información del paciente.....	62
Figura A9: Confirmar eliminar paciente .....	62
Figura A10: Seleccionar área .....	63
Figura A11: Ventana de marcas detallada .....	64
Figura A12: Botón de selección de marca .....	64
Figura A13: Ventana selector de marcas .....	65
Figura A14: Indicador de marca seleccionada.....	65
Figura A15: Ventana de marcas .....	66
Figura A16: Zona de inserción de marcas.....	66
Figura A17: Detalle de marca.....	67
Figura A18: Confirmación de guardado de marca.....	67
Figura A19: Resultado de registro de marca.....	68
Figura A20: Historial de marcas .....	68
Figura A21: Selección de marca en el historial .....	69
Figura A22: Detalle de marca.....	69
Figura A23: Confirmación eliminación de marca.....	70
Figura A24: Impresión de marcas .....	71

## Autorización

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Firmado:

Sara Llorente Navarro

Carlos Manuel Muñoz Díaz

José Carlos Olivar Conde

## Resumen

DermaSystem ha sido un trabajo de fin de grado que aportó a sus autores la oportunidad de reflejar los conocimientos que adquirieron a lo largo de sus años de estudio en la Facultad de Informática en la Universidad Complutense de Madrid.

No sólo eso, sino que permitió desarrollar una aplicación informática para un cliente real, lo que confiere al trabajo una componente de utilidad. En concreto, el trabajo surgió gracias a la iniciativa de la Doctora en medicina D<sup>a</sup> Isabel Longo Imedio, especialista en Dermatología Médico-Quirúrgica y Venereología en el Hospital Gregorio Marañón de Madrid quien planteó la necesidad de desarrollar una herramienta informática en el área de dermatología facilitar el seguimiento de pacientes con múltiples tumores cutáneos.

Pese a que el principal objetivo fue cubrir el mayor número posible de necesidades y requisitos para tal aplicación, como fue un sistema de marcado de imágenes, este proyecto persiguió además otras metas, que se sintetizan en lo siguiente, lograr que DermaSystem fuese escalable para conseguir que en el futuro otros técnicos en el área informática pudiesen continuar su desarrollo aportándole nuevas funcionalidades.

El resultado final, fruto de la confluencia de intereses y objetivos, ha sido el desarrollo de la aplicación DermaSystem.

## Palabras clave

Aplicación informática, sistema de marcado, dermatología, Java, Modelo Vista Controlador, programación orientada por eventos, i18n, Hibernate.

## Abstract

DermaSystem has been a final degree project, which allowed the authors the authors to reflect the knowledge that they acquired during their years of study at the School of Computer Science in the University Complutense of Madrid.

Additionally, the authors were able to develop an application for a real client, giving the work a practical use. Specifically, the work came about through the initiative of the medical doctor Imedio Ms. Isabel Longo, Specialist Medical-Surgical Dermatology and Venereology at the Hospital Gregorio Marañón de Madrid who raised the need to develop a software tool in the area of Dermatology to monitor the follow-up of patients with multiple cutaneous tumors.

Although the main objective was to cover the widest possible range of needs and requirements for such an application, as a marking system of images, this project also pursued other goals: that basically comprise the design of DermaSystem as a scalable tool to ensure that other engineers in computer science could continue its development by incorporating new features.

The outcome, as a result of the confluence of interests and goals, has been the development of the DermaSystem application.

## Keywords

Computer-based application, marking system, dermatology, Java, Model View Controller, event-driven programming, i18n, Hibernate.

# 1 Introducción

El proyecto que se presenta surge como una iniciativa de la Doctora en medicina D<sup>a</sup> Isabel Longo Imedio, especialista en Dermatología Médico-Quirúrgica y Venereología en el Hospital Gregorio Marañón de Madrid quien plantea la necesidad de desarrollar una herramienta informática en el área de dermatología como ayuda al seguimiento y diagnóstico de pacientes en dicha área.

En esta primera fase del proyecto se pretende desarrollar la base para una aplicación de futuro en la que se puedan incluir los avances tecnológicos a los que hubiere lugar, incluyendo el procesamiento de imágenes con fines de tratamientos selectivos y automáticos como ayuda al diagnóstico de pacientes.

Tanto los avances en las consultas dermatológicas como la investigación relacionada en los laboratorios demandan actualmente una tecnología que ayude a realizar las tareas cotidianas de una manera más sencilla y eficaz. De aquí surge una de las principales ideas de la aplicación DesmaSystem plasmada en la presente memoria. Gracias a ella, doctores e investigadores pueden desarrollar su actividad rutinaria sin prestar más atención de la necesaria a tareas consideradas como secundarias en el ámbito de la medicina, tal es el caso del almacenamiento de datos, registro de pruebas, comprobación de resultados de almacenamiento, copias de seguridad de la información u otras tareas consideradas accesorias.

Hoy día existe un creciente interés por el desarrollo de aplicaciones informáticas en todos los ámbitos de la vida. En este sentido, no podía ser menos la medicina y más concretamente la dermatología. Existen diversas iniciativas en el área de la dermatología cuya finalidad consiste en aplicar las tecnologías informáticas como ayuda a las tareas de consulta diarias, a la vez que se desarrollan iniciativas de investigación. Tal es el caso de algunos sistemas basados en razonamiento como ayuda al diagnóstico (Aldridge y col., 2011), que se utilizan como soporte a la vez que sirve de herramienta de evaluación en los procesos de enseñanza-aprendizaje en dicha área.

Sin embargo y como paso previo, no se pretende un sistema sofisticado de diagnosis, sino más bien comenzar por la parte más sencilla de organización y visualización efectiva de la información. En definitiva, se trata de dar respuesta a la siguiente pregunta que se plantea de forma sencilla e intuitiva: ¿por qué no extender estos avances tecnológicos a procesos más comunes en la vida diaria de una consulta dermatológica?

Sabemos que un dermatólogo pasa consulta día a día, lleva un registro de las afecciones de sus pacientes y sus diagnósticos mediante un historial clínico. Es decir, realiza un seguimiento de sus pacientes, estas tareas, absolutamente rutinarias, implican un consumo de tiempo, que puede utilizarse en otros menesteres más productivos y eficientes desde el punto de vista de las actividades de los especialistas en dermatología.

Si bien es cierto que gran parte de este proceso está actualmente siendo informatizado y gracias a ello el personal médico tiene acceso a todos estos datos, no cabe la menor duda de que siguen necesitandose herramientas informáticas específicas para poder realizar los trabajos cotidianos de una manera más eficaz durante sus consultas. Tal es el caso que nos ocupa, que origina el planteamiento de la aplicación que se presenta

Con el objetivo de su ampliación futura, hacia metas más ambiciosas, se presenta un diseño amplio que contempla tal posibilidad, sin descartar su proyección hacia la telemedicina como disciplina en pleno desarrollo y a tener en cuenta dentro de los desarrollos tecnológicos de futuro (Martínez-Alcalá y col., 2012).

Dentro de la Telemedicina, los avances se centran sobre la base de acceso a la información almacenada en determinados servidores (computadores de altas prestaciones) mediante tecnologías web. Sin duda dicho almacenamiento se ha de gestionar eficientemente mediante el acceso a bases de datos específicas, algo que de alguna manera contempla la aplicación que se presenta, en tanto en cuanto el diseño se plantea abierto a dichas tecnologías, aunque en este caso la aplicación se desarrolle de forma local.

Las técnicas basadas en aplicaciones informáticas se ven favorecidas por las posibilidades ofrecidas por el cada vez más amplio uso de las tecnologías a través de la World Wide Web (www). Esto es así por la facilidad de intercambio de información a todos los niveles y de imágenes en particular. Más específicamente, se habla de lo que hoy en día conforma una entidad cada vez más relevante, el cual es el caso de lo que se conoce como “e-Health” o salud electrónica, cuya finalidad estriba en mejorar los servicios de salud (diagnóstico, seguimiento y gestión) gracias a la aplicación de las tecnologías de información y comunicación.

No cabe la menor duda que en todos los ámbitos profesionales y en el médico-dermatológico en particular, existe un creciente uso de tecnologías informáticas donde se incluyen tareas relacionadas con la comunicación mediante el envío e intercambio de información, así como la consulta de historiales clínicos (Kurki y col., 2012; Melas y col., 2012). Todas las actividades encaminadas a facilitar y potenciar estas tecnologías ha de redundar con claridad en una mejora en los servicios de salud, tal y como resulta ser la pretensión de los mismos.

La posibilidad de disponer de las aplicaciones informáticas en los cada vez más difundidos dispositivos electrónicos con conexiones fiables de red (debido fundamentalmente al avance de las tecnologías asociadas) hace que los desarrollos informáticos en el ámbito de la medicina encuentren un caldo de cultivo apropiado para favorecer su difusión y uso.

Dentro de los planteamientos y aplicaciones informáticas propuestas, las interfaces y, más concretamente, las gráficas, constituyen un elemento esencial que permiten visualizar la máxima información posible con el mínimo tiempo y esfuerzo. Con tal propósito, las aplicaciones informáticas requieren, sin duda, de un diseño robusto a la vez que intuitivo y amigable con el objeto de conseguir que el sistema sea eficiente, robusto y atractivo. En este sentido, en el trabajo que se presenta se plantea igualmente tal posibilidad, de forma que, tanto el diseño actual como la pretendida ampliación del mismo implique el menor esfuerzo posible, facilitando su uso a la vez que se consiguen las metas mencionadas en el desarrollo actual.

Por las razones y motivos anteriormente expuestos, se presenta el diseño de la aplicación informática en dermatología (DermaSystems) cuyo planteamiento y detalles se exponen a lo largo de la presente memoria. Conviene tener en cuenta que el diseño se basa en los requisitos planteados desde el área de dermatología, reseñada previamente, bajo las consideraciones de los desarrollos basados en las ciencias computacionales.

## 2 Planteamiento

La principal ventaja de un proyecto de esta índole es su naturaleza real, dentro de un marco específico, el cual es la dermatología en el ámbito de un Hospital.

La doctora especialista en dermatología planteó esta necesidad con el fin de su uso diario, cuya pretensión era apoyarse en esta aplicación durante sus consultas guardando datos, diagnósticos e historiales clínicos de pacientes. En este sentido, y desde el punto de vista práctico, el proyecto atiende a los requisitos de la doctora especialista en dermatología, que ejerce el rol de cliente y usuario final a la vez, tal y como se entiende en los desarrollos de los proyectos informáticos.

Con tal propósito se ha creado DermaSystem. Una aplicación informática orientada a servir como herramienta de ayuda en el área de dermatología; la cual proporcionará apoyo y ayuda durante las consultas dermatológicas diarias, facilitando las labores rutinarias que no implican dedicación directa al paciente y en el futuro al diagnóstico.

DermaSystem se ha desarrollado para un cliente en concreto, sin embargo se ha querido dotar a la aplicación de una cierta escalabilidad. De tal manera que en un futuro esta herramienta pueda ser usada por más doctores dentro de la misma

especialidad dermatológica, a la vez que en otras especialidades médicas a las que puede servir de inspiración.

Se pretende como algo primordial que su funcionalidad pueda ser ampliada para adaptarse a las necesidades y preferencias de todos los usuarios que la han de utilizar en el futuro.

Como pre-requisito se ha tenido en cuenta el hecho de tratar de cubrir las principales necesidades del cliente, que se sintetizan de forma concisa en lo siguiente

- Aplicación sencilla y amigable.
- Fácil de usar.
- Cumplir al menos las siguientes funcionalidades: disponer de un sistema de marcado de imágenes prediseñadas del cuerpo humano, una lista de las afecciones dermatológicas más comunes (aportadas por la doctora) y, por supuesto, una gestión de historial de pacientes.

### 3 Organización del trabajo

A lo largo de la presente memoria se puede encontrar información suficiente sobre las etapas de desarrollo de este proyecto, así como su arquitectura o las tecnologías que se usaron. Se detallan también los objetivos que se perseguían para hacer de DermaSystem la aplicación ideal para el propósito que se pretende.

Este trabajo se organiza en cinco partes bien diferenciadas, cada una de las cuales aborda una parte esencial de lo que se considera el ciclo de vida de un proyecto de desarrollo software:

- Especificación de Requisitos: en este apartado se detallan los requerimientos indispensables que DermaSystem debía cumplir para su aceptación por parte del usuario final.
- Especificación de casos de uso: en esta sección se detallan las acciones que se pueden realizar en la aplicación como consecuencia de los procesos de trabajo habituales. Se determina el camino que se debe seguir para su ejecución así como los usuarios que podrán realizarlas.
- Diseño: esta parte detalla la arquitectura y el diseño de DermaSystem. Se puntualiza sobre los patrones usados y se explica cómo han sido usados, planteados y diseñados.

- Tecnologías utilizadas: al tratarse de un proyecto de cierta envergadura, máxime teniendo en cuenta las ampliaciones de futuro previstas, se han utilizado distintas técnicas y tecnologías para hacer que DermaSystem cumpliera todos los requisitos que se propusieron como parte esencial del proyecto. Aquí se detallan por tanto las tecnologías usadas y qué función cumplen dentro del proyecto.
- Implementación: en esta sección se proporciona una explicación detallada de cómo han sido diseñados distintos componentes de DermaSystem, aportando parte del código desarrollado en forma de imágenes y esquemas para su clarificación.

Todas estas cuestiones se detallan en los diferentes apartados que constituyen los bloques esenciales y las secciones correspondientes del trabajo que se presenta como síntesis del proyecto desarrollado.

## 4 Análisis de la situación actual

Para poder realizar una correcta especificación de requisitos respecto a la aplicación DermaSystem ha sido necesario mantener una serie de entrevistas con la clienta o usuaria final de la aplicación, que de alguna forma simboliza un usuario genérico en el área considerada, esto es, en dermatología.

Las entrevistas constituyen una herramienta de relevancia a la hora de clarificar y explicitar el conocimiento subyacente en el usuario, de forma que desde el punto de vista de la ingeniería informática queden recogidos todos los aspectos relevantes sin omisiones. Este aspecto resulta bien conocido en el ámbito de la ingeniería del software y más específicamente cuando el sistema informático involucra la necesidad de extraer información relevante con implicación de técnicas de conocimiento, tal es el caso de lo que se conoce específicamente como ingeniería del conocimiento (Pajares y Santos, 2005).

Con tal propósito se han considerado de forma especial las dificultades con las que se enfrenta un dermatólogo a la hora de pasar consulta, las necesidades que surgen a la hora de revisar los datos del paciente al que trata, así como los problemas que detecta a la hora de documentar tanto su historial como las afecciones actuales y pasadas del paciente.

Tal y como queda patente desde el inicio de las primeras entrevistas, es práctica habitual que el sistema de tratamiento de la información médica de un paciente se lleve a cabo manualmente, es decir, el doctor toma notas en papel sobre la información que le proporciona el paciente, posteriormente lo evalúa e incorpora su diagnóstico al historial. Es cierto que las aplicaciones ofimáticas e informáticas están

sustituyendo paulatinamente estas tareas, si bien no desde el punto de vista que se plantea en la presente aplicación informática. Queda pues patente, que se necesita un desarrollo que supere las meras tecnologías informáticas a la vez que facilite la labor del dermatólogo.

El principal problema que se detecta en una primera aproximación estriba en el difícil acceso a la información médica, especialmente relevante para la dermatóloga. Esto es así, ya que, pese a tener un historial clínico sobre el paciente, debe leerlo íntegramente para identificar los distintos problemas y lesiones de piel que el paciente ha tenido así como valorar lo reflejado en el informe con lo que se observa en el momento de la exploración presente. Esto, en la mayoría de las ocasiones, puede consumir un tiempo importante alargando la duración de la consulta a la vez que reduce el tiempo de dedicación a otras labores de diagnóstico o investigación más relevantes desde el punto de vista de la eficiencia médica. Estos consumos de tiempo afectan de manera importante a los procesos de enseñanza-aprendizaje involucrados en las consultas médicas en general, y en dermatología en particular. Gracias a la aplicación desarrollada, estos tiempos pueden verse reducidos considerablemente, aspecto que redundará en la efectividad de la misma.

En ocasiones, en el historial clínico del paciente el propio médico guarda bocetos del cuerpo humano, dibujando y señalando sobre el dibujo las lesiones observadas en la piel. A veces estas señales desaparecen del paciente, bien como consecuencia del tratamiento o por otras razones. Este es el momento en el que se les hace desaparecer del expediente de forma manual, a veces incluso sin anotación de fecha, ello conlleva posteriormente una cierta dificultad en el seguimiento de tales lesiones en el paciente.

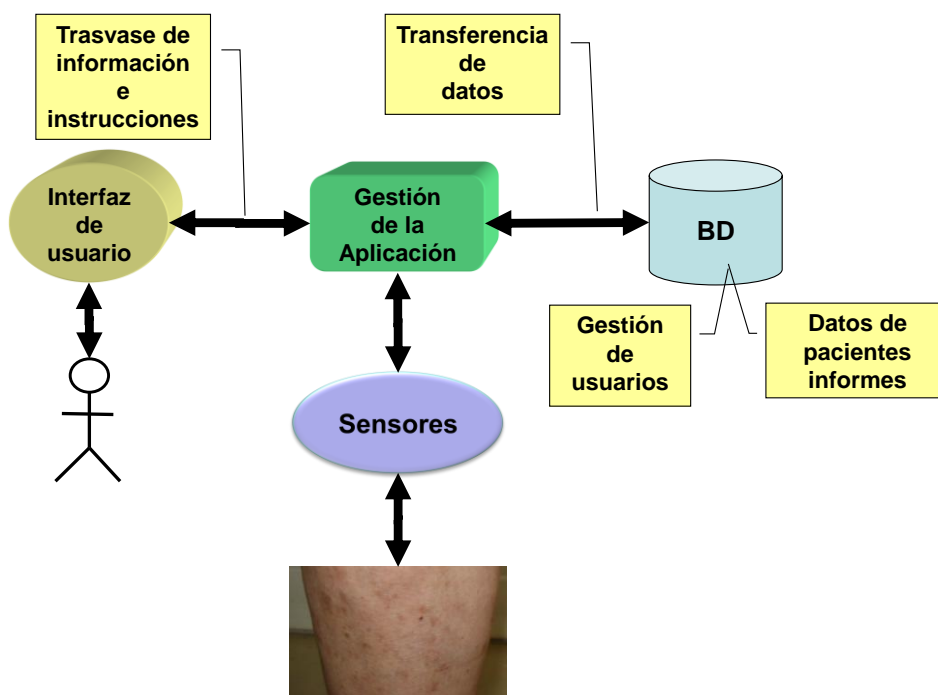
No solamente la desaparición de las lesiones constituye un problema sino la aparición de nuevas lesiones o señales en la piel. En este último caso el proceso consiste en la anotación o dibujado, describiendo sus características esenciales y secuencia de fechas. Independientemente de que estos procesos se realicen a través de herramientas ofimáticas, estas tareas pueden resultar tediosas, con tendencia a evitar determinadas anotaciones, que no quedarán reflejadas eficientemente en el expediente

Como se ha mencionado previamente, la herramienta cuyo desarrollo se presenta trata de solventar estas dificultades, facilitando las tareas más pesadas al especialista que ha de realizar las anotaciones o informes, a la vez que le proporciona una valiosa ayuda en sentido amplio.

En la Figura 1, se muestra el diagrama de bloques de alto nivel relativo al diseño inicial, surgido como consecuencia de la síntesis de los procesos de extracción de la información según los pasos especificados previamente.

En dicho diagrama se muestran cuatro bloques básicos que constituyen el conjunto del diseño:

- Interfaz de usuario
- Gestión de la Aplicación
- Base de Datos
- Sensores



**Figura 1: Arquitectura general del sistema**

El usuario accede a la aplicación a través de la interfaz de usuario, que envía información hacia el módulo Gestión de la Aplicación procedente de aquel. Cuando se han realizado los procesos necesarios, el usuario recibe la información pertinente como resultado del proceso realizado. Existe un trasvase de información e instrucciones en el doble sentido especificado por la doble flecha.

El módulo Gestión de la Aplicación se encarga de la organización de los procesos, así como de la adquisición de la información procedente bien de la Base de Datos o directamente de los sensores.

La Base de Datos contiene información relativa tanto a pacientes como a la propia gestión del sistema. En el primer caso se trata de datos de pacientes con su

correspondiente identificación e informes asociados. En el segundo caso, se hace referencia a la gestión de usuarios por parte del administrador del sistema y el acceso de estos mismos usuarios, tal y como se describe posteriormente.

El módulo identificado como Sensores es el encargado de la captura de información procedente directamente del paciente, está previsto dotar a la aplicación de una unidad de captura de imágenes mediante cámaras de visión, que proporcionarán información directa de las lesiones del paciente. Cabe la posibilidad de conectar otro tipo de sensores de captura de datos, que en su momento se consideren de interés relevante. Esta es la justificación para identificar el módulo Sensores de forma genérica. No obstante, esta parte no constituye específicamente el objetivo de la aplicación que se presenta, dejándose para desarrollos futuros.

Según los planteamientos anteriores, a continuación se exponen los objetivos principales que conforman el presente proyecto.

## 5 Objetivos

En vista de los problemas que le surgen a un especialista en dermatología durante su consulta habitual a la hora de tratar el historial de enfermedades o afecciones de los pacientes que atiende, así como su correcto seguimiento se pretende:

- Crear una aplicación que facilite el tratamiento de la información de tal manera que el médico pueda manejar la información que le transmite el paciente sobre su afección, así como poder realizar las anotaciones pertinentes de una manera rápida y eficaz, a la vez que lo refleja en el correspondiente informe, que se almacenará debidamente en la base de datos de la aplicación.

Al ser una aplicación destinada al servicio de dermatología donde se tratan problemas relacionados con lesiones de la piel, es muy importante que “mediante un simple vistazo” se puedan localizar las principales señales o síntomas que presenta el paciente en su piel, identificando las zonas más significativas o relevantes a la vez que se sitúan sobre la base del cuerpo humano. De aquí surge otro de los objetivos principales de esta aplicación, que consiste en:

- Desarrollar un sistema de marcado de imágenes prediseñadas del cuerpo humano.

Este sistema de marcado debe estar dotado de un cierto control, en concreto, debe registrar el tipo de marca, su nombre, descripción, la fecha en la que se introdujo y las anotaciones o diagnóstico del médico, de nuevo, a partir de esta consideración surge un nuevo objetivo:

- Control del sistema de marcado.

Al estar tratando con datos personales de un paciente y su historial clínico, se debe disponer de:

- Una Base de Datos robusta. Dicha base de datos debe ser sobre todo segura, ya que, entre otras cosas, se está tratando con información confidencial. Debe permitir el acceso a los datos a través de la aplicación de una manera fácil y rápida, si bien garantizando la integridad de la información.

El manejo de datos confidenciales hace necesario el uso de un sistema de autenticación. Ello permitirá controlar los accesos al sistema y además separará la administración de la aplicación de las funciones propias que utilizará el dermatólogo; sin olvidar que aportará escalabilidad a la aplicación permitiendo su uso a múltiples usuarios en un futuro. En este sentido se plantea su ampliación a plataformas multiusuario:

- Sistema de autenticación de usuarios mediante nombre de usuario y contraseña.

Por tanto, DermaSystem pretende dotar al servicio de dermatología en general y al dermatólogo en particular de una herramienta de trabajo útil que le permita registrar todos los datos del paciente que considere necesarios así como una manera gráfica de identificar y marcar las lesiones y señales que el paciente tenga en la piel. Todo ello bajo las características propias de un entorno amigable.

De una manera rápida, ágil y fácil podrá registrar estas señales, comprobar las que el propio paciente sufría anteriormente, como resultado de visitas anteriores a la consulta, así como documentarlas con comentarios específicos de utilidad para su posterior seguimiento y utilización en los informes que a tal fin puedan generarse.

## **6 Especificación de requisitos**

Como se ha mencionado previamente, DermaSystem debe ser una aplicación diseñada con el objetivo de ayudar a los especialistas en dermatología en su quehacer diario. De forma especial, en el manejo de los historiales médicos, pero más concretamente en la identificación y anotación de las lesiones, señales y marcas de la piel que presentan sus

pacientes. Todo ello debe diseñarse respetando la confidencialidad de los datos gestionados, así como preservando su integridad. Tal y como se ha mencionado previamente, a esto hay que añadir su carácter abierto para la incorporación de otras funcionalidades más avanzadas e incluso a plataformas multiusuario.

## 6.1 Sistema de marcado

Los dermatólogos, usuarios del sistema, deben poder añadir y eliminar marcas así como crear comentarios específicos para cada una de ellas. Para anotar estas marcas dispondrán de imágenes prediseñadas o plantillas que representan total o parcialmente el cuerpo humano. Cabe distinguir los dos tipos de plantillas genéricas que se mencionan a continuación:

- *Frontales*: tres plantillas para la cabeza (lateral izquierdo, frontal y lateral derecho) y una para el cuerpo completo.
- *De espalda*: dos plantillas para la cabeza (trasera y superior) y una para el cuerpo.

El usuario pulsará con los botones del ratón asociado al computador, sobre la zona afectada, se le mostrará un panel con todas las marcas disponibles, seleccionará la deseada y se incorporará a la imagen en la zona señalada por el médico. Asimismo, el doctor también podrá añadir una descripción detallada propia si lo desea sin limitación específica.

## 6.2 Control del sistema de marcado

Cada marca quedará registrada según la fecha y la hora de creación. En el caso de eliminación de marcas, éstas quedarán suprimidas de la imagen prediseñada, sin embargo no se eliminarán del historial del paciente con el fin de disponer de un histórico y evolución de sus lesiones.

## 6.3 Base de Datos

El historial clínico del paciente se guardará en una Base de Datos que almacenará sus datos personales, todas las marcas y anotaciones que haya realizado el doctor así como su diagnóstico.

Se ha hecho especial hincapié en el diseño de esta Base de Datos para poder en un futuro exportar los datos a un sistema de mayor capacidad y prestaciones desde el punto de vista informático, como pueda ser la base de datos del Hospital Gregorio Marañón de Madrid.

Su diseño contempla necesariamente el hecho de preservar la integridad de la información que contiene.

## **6.4 Autenticación de usuarios**

Se dotará a la aplicación de un sistema de verificación, que controlará el acceso a los datos y además delimitará las funciones de la aplicación en función del tipo de usuario que inicie sesión.

El sistema se diseñará de forma que existan dos tipos de usuarios: administrador y usuario habitual:

- El administrador llevará a cabo tareas relacionadas con el mantenimiento de la aplicación gestionando el acceso de los usuarios y comprobando la no violación de los accesos al sistema.
- El dermatólogo limitará sus acciones al uso de la aplicación, por tanto su rol será el conocido como usuario habitual desde el punto de vista de las aplicaciones informáticas.

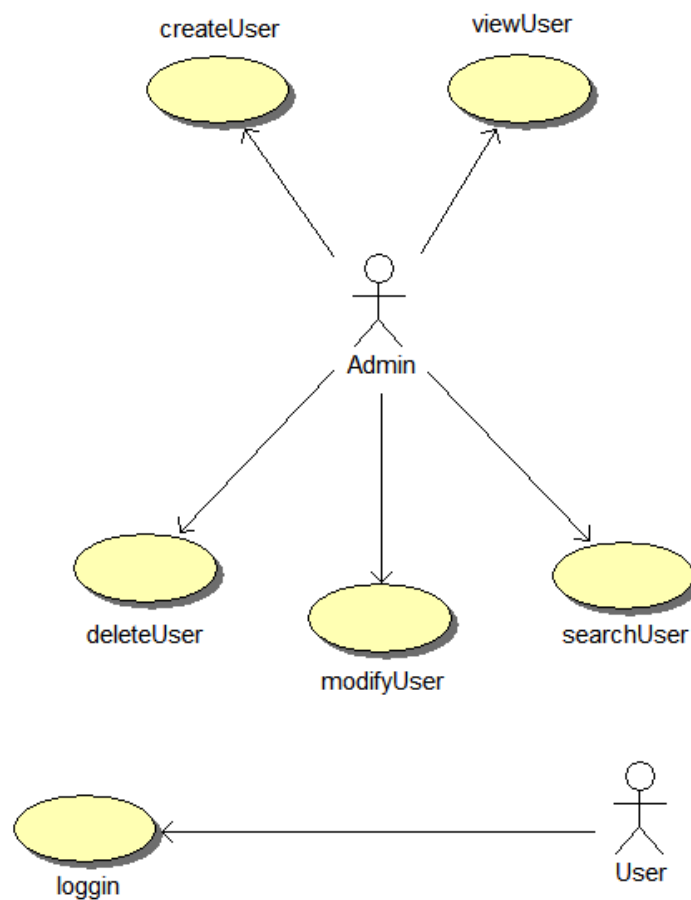
El objetivo que se persigue es dotar a la aplicación de un mayor nivel de seguridad y facilitar la posible implantación de DermaSystem en un sistema de más altas prestaciones.

En los sistemas multiplataforma debe contemplarse la compatibilidad y accesos simultáneos de varios usuarios, por lo que se han de habilitar los mecanismos pertinentes con tal propósito. Esto, evidentemente requiere de un control exhaustivo en lo que a los accesos a la aplicación se refiere.

## 7 Diagrama de casos de uso

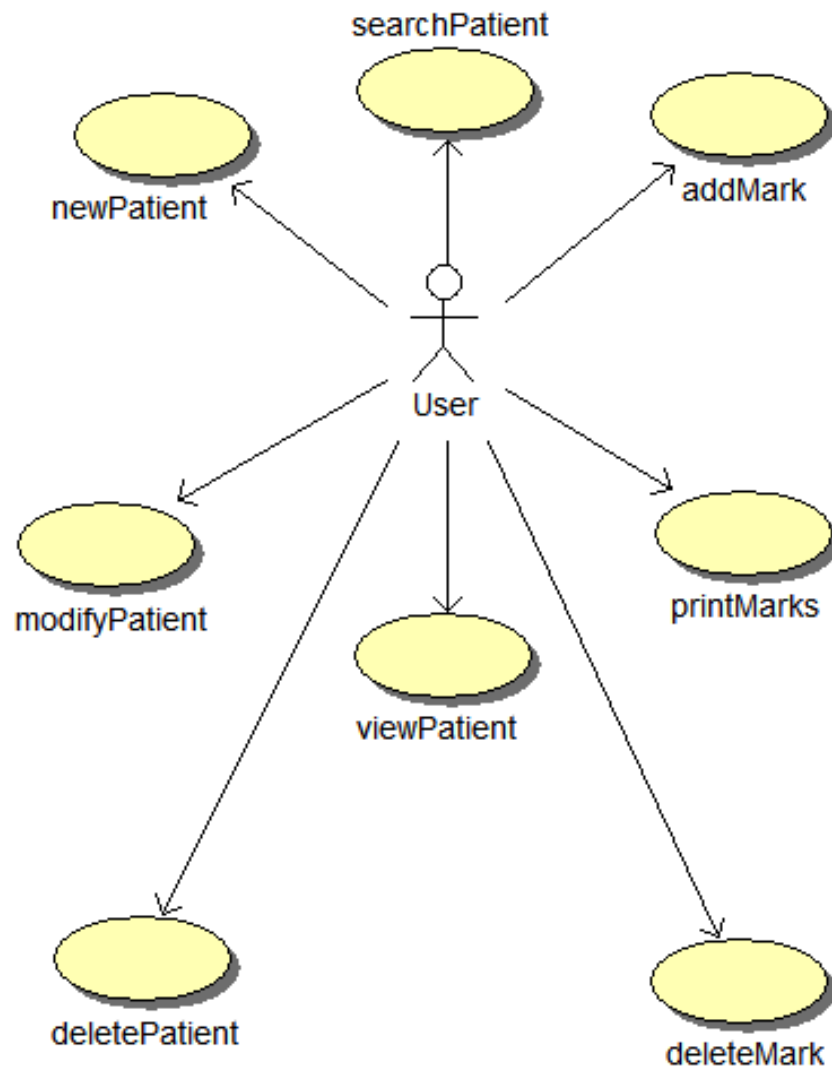
En la Figura 2 y Figura 3 se muestran sendos esquemas de casos de uso desde el punto de vista de los roles de administrador y usuario que debe identificarse para el acceso a la aplicación.

Sus funciones, tal y como se especifica más adelante están relacionadas con la gestión de usuarios en el primer caso y con el acceso a la aplicación en el segundo.



**Figura 2: Diagrama del módulo usuarios (administrador y usuario normal)**

Como puede observarse en la gestión de usuarios, el administrador constituye el centro del esquema que en su rol de principal tiene control sobre las funciones asociadas, en este caso cinco con posibilidad de realizar operaciones de control sobre los distintos usuarios en relación a la aplicación. Por parte del usuario, éste aparece relacionado con la operación de acceso a la aplicación, a través de la operación de entrada mediante su identificación previa a través del “login”



**Figura 3: Diagrama del módulo usuarios en relación a pacientes**

El usuario que ejerce su rol principal en la aplicación una vez identificado, tiene la posibilidad de ejercer diversas funciones y operaciones en la aplicación orientadas a la gestión de pacientes. Dichas operaciones están relacionadas con la creación de un espacio para la inserción de un nuevo paciente o la búsqueda de uno ya existente, así como su modificación e incluso eliminación. En esta línea también realiza operaciones de visualización del paciente creado o existente. El usuario igualmente tiene la capacidad de realizar operaciones sobre el marcado de los pacientes (añadir, modificar o imprimir).

## 8 Especificación de casos de uso

Como podemos observar en los diagramas mostrados en la Figura 2 y en la Figura 3, se identifican dos roles:

1. El administrador de la aplicación: es el encargado del mantenimiento de la aplicación y, como tal, gestiona los usuarios que la puedan utilizar. Puede dar de alta nuevos usuarios, a la vez que modificar o consultar sus datos o eliminarlos del sistema.
2. Usuario estándar/dermatólogo: es la persona que maneja la aplicación. Tendrá acceso a toda la funcionalidad que la aplicación ofrece sobre los pacientes; sin embargo no podrá gestionar usuarios.

La aplicación está dividida en dos subsistemas bien diferenciados:

- Usuarios
- Pacientes

El subsistema *Usuarios* es el encargado de la gestión de accesos a la aplicación.

Mientras que el subsistema *Pacientes* gestionará la información y datos de los pacientes registrados por el médico.

Los casos de usos mostrados en los diagramas anteriores se pueden agrupar y resumir brevemente:

Para el subsistema de usuarios:

- **CreateUser**: crea un nuevo usuario que podrá acceder al sistema. Dependiendo del rol que se le proporcione tendrá acceso a distintas funciones en la aplicación.
- **ModifyUser**: modifica los datos de un usuario existente. Asimismo permite cambiar el rol que dicho usuario desempeña en la aplicación.
- **DeleteUser**: desactiva un usuario existente en el sistema, de tal manera que éste ya no podrá acceder a ninguna función de la aplicación. Es decir impide su acceso a la misma.
- **SearchUser**: permite buscar un usuario en la base de datos por nombre, apellidos y DNI.
- **ViewUser**: muestra los datos de un usuario registrado en el sistema.

- **LogIn:** Permite el inicio de sesión en la aplicación introduciendo el nombre de usuario y la contraseña. En función del rol que tenga el usuario, podrá acceder a unas funciones u otras.

Para el subsistema de pacientes:

- **NewPatient:** crea un paciente nuevo registrando sus datos personales.
- **ModifyPatient:** modifica los datos personales del paciente.
- **DeletePatient:** desactiva un paciente en la base de datos.
- **SearchPatient:** permite buscar un paciente en la base de datos por nombre, apellidos y DNI.
- **ViewPatient:** muestra el historial del paciente así como sus datos personales.
- **AddMark:** añade una nueva marca a la imagen del paciente. El doctor como usuario normal puede seleccionar la que desee y añadir un comentario específico para esa marca.
- **DeleteMark:** borra una marca anteriormente creada. Se borrará de la imagen, pero permanecerá registrada en la base de datos.
- **PrintMarks:** Imprime en PDF la marcas del paciente actualmente seleccionado.

## 8.1 Subsistema Usuarios

### 1. *CreateUser*

#### **Precondiciones:**

El usuario que ejecuta la acción debe ser administrador del sistema y además estar registrado e identificado correctamente (*"logueado"*).

#### **Flujo Normal:**

1. Seleccionar en el menú "Usuarios", opción "Añadir usuario".
2. La aplicación pide al administrador que rellene los datos del nuevo usuario y el rol que va a desempeñar.
3. Para guardar el nuevo usuario seleccionar la opción "Guardar" y el usuario quedará registrado y almacenado en la base de datos.

**Alternativas:**

- Del paso 1 al 2 especificados anteriormente el administrador puede cancelar el proceso de registro.

**2. ModifyUser****Precondiciones:**

El usuario que ejecuta la acción debe ser administrador del sistema y además estar registrado e identificado correctamente (*"logueado"*).

**Flujo Normal:**

1. Seleccionar en el menú "Usuarios", opción "Gestor de Usuarios".
2. La aplicación muestra un buscador por nombre de usuario para localizar aquel usuario que se desea modificar. Presionando el botón "Buscar" realiza la consulta y lista sus resultados.
3. Al seleccionar el usuario que se desea modificar se mostrará un formulario con todos sus datos.
4. Introducir los nuevos datos del usuario en el formulario.
5. Para guardar los cambios se selecciona la opción "Guardar" quedando la nueva información almacenada en la base de datos.

**Alternativas:**

- Del paso 1 al 4 el administrador puede cancelar el proceso de modificación.

**3. DeleteUser****Precondiciones:**

El usuario que ejecuta la acción debe ser administrador del sistema y además estar registrado e identificado correctamente (*"logueado"*).

**Flujo Normal:**

1. Seleccionar la opción del menú "Usuarios", opción "Gestor de Usuarios".
2. La aplicación muestra un buscador por nombre de usuario para localizar aquel usuario que se desea modificar. Presionando el botón "Buscar" se realiza la consulta y se listan sus resultados.

3. Al seleccionar el usuario que se desea eliminar se mostrará un formulario con todos sus datos.
4. Pulsar el botón “Eliminar”.
5. Aparece una ventana de aviso sobre la eliminación de usuarios. Para la eliminación definitiva se pulsará el botón “Aceptar” y el usuario quedará eliminado del sistema.

**Alternativas:**

- Del paso 1 al 3 el administrador puede cancelar el proceso de eliminación del usuario.
- En el paso eliminación de usuarios aparece una ventana de verificación para garantizar que no se trata de un error. Seleccionar el botón “Cancelar” para no eliminar el usuario.

#### **4. SearchUser**

**Precondiciones:**

El usuario que ejecuta la acción debe ser administrador del sistema y además estar registrado e identificado correctamente (“logueado”).

**Flujo Normal:**

1. Seleccionar en el menú “Pacientes” la opción “Gestor de Usuarios”.
2. La aplicación muestra un buscador por nombre, apellido y DNI.
3. Presionar el botón “Buscar” para realizar la consulta y listar sus resultados.

**Alternativas:**

- En relación al punto 2, presionando el botón “Limpiar” se borrarán los datos introducidos.

#### **5. ViewUser**

**Precondiciones:**

El usuario que ejecuta la acción debe ser administrador del sistema y además estar registrado e identificado correctamente (“logueado”).

**Flujo Normal:**

1. Seleccionar en el menú “Usuarios”, opción “Consultar usuarios”.
2. La aplicación muestra un buscador por nombre de usuario. Presionando el botón “Buscar” se realiza la consulta y se listan sus resultados.
3. Al seleccionar un usuario de la lista se abrirá una ventana que muestra todos sus datos.

**Alternativas:**

- Del paso 1 al 3 el administrador puede cancelar el proceso.

## 6. Login

**Precondiciones:**

No existen.

**Flujo Normal:**

1. Al abrir la aplicación se mostrará la pantalla de autenticación de usuario.
2. Rellenar el nombre de usuario y la contraseña.
3. Al pulsar el botón “Aceptar” se comprueban los datos introducidos y aparece la ventana correspondiente al rol del usuario registrado en el dominio.

**Alternativas:**

- Del paso 1 al 2 el administrador puede cancelar el proceso de registro.

## 8.2 Subsistema Pacientes

### *NewPatient*

**Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio correctamente (“logueado”).

**Flujo Normal:**

1. Seleccionar en el menú “Pacientes”, opción “Añadir paciente”.
2. Aparecerá un formulario donde rellenar los datos del paciente.

3. Pulsar el botón "Aceptar" para guardar el nuevo paciente.
4. Aparece una ventana de confirmación de guardado. Presionar el botón "Aceptar" para cerrarla.

**Alternativas:**

- Del paso 1 al 3 el doctor puede cancelar el proceso de creación de paciente.
- En relación al punto tres, si los datos introducidos son incorrectos se mostrará una ventana advirtiéndolo. Pulsar el botón "Aceptar" para cerrarla y corregir los errores.

**ModifyPatient**

**Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio ("*logueado*").

**Flujo Normal:**

1. Seleccionar en el menú "Pacientes", opción "Gestor de Pacientes"..
2. La aplicación muestra un buscador por nombre, apellido y DNI. Presionando el botón "Buscar" realiza la consulta y lista sus resultados.
3. Seleccionar el paciente deseado de la lista y presionar "Aceptar" para ver sus datos e historial.
4. Pulsar el botón "Editar". Se habilita el formulario para poder realizar cambios.
5. Una vez efectuados los cambios pulsar el botón "Guardar".
6. Aparece una ventana de confirmación de guardado. Presionar el botón "Aceptar" para cerrarla.

**Alternativas:**

- En el paso cuatro, durante la acción "Editar", el doctor puede cancelar el proceso de edición de los datos del paciente.

- En el paso cinco, durante el guardado, si los datos introducidos son incorrectos se mostrará una ventana advirtiéndolo. Pulsar el botón “Aceptar” para cerrarla y corregir los errores.

### *DeletePatient:*

#### **Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar registrado correctamente en el dominio (“logueado”).

#### **Flujo Normal:**

1. Seleccionar en el menú “Pacientes”, opción “Gestor de Pacientes”.
2. La aplicación muestra un buscador por nombre, apellido y DNI. Presionando el botón “Buscar” realiza la consulta y lista sus resultados.
3. Seleccionar el paciente deseado de la lista y presionar “Aceptar” para mostrar sus datos e historial en pantalla.
4. Pulsar el botón “Editar”.
5. Pulsar el botón “Eliminar”.  
Aparece una ventana de confirmación de eliminación. Presionar el botón “Aceptar” para eliminar el paciente definitivamente.

#### **Alternativas:**

- Del paso 4 al 7 el doctor puede cancelar el proceso de eliminación de paciente.

### *SearchPatient*

#### **Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar correctamente registrado en el dominio (“logueado”).

#### **Flujo Normal:**

1. Seleccionar en el menú “Pacientes”, opción “Gestor de Pacientes”.
2. La aplicación muestra un buscador por nombre, apellido y DNI. Presionando el botón “Buscar” realiza la consulta y lista sus resultados.

**Alternativas:**

- En relación al punto dos, presionando el botón “Limpiar” se borrarán los datos introducidos.

**ViewPatient:****Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar registrado correctamente en el dominio (“logueado”).

**Flujo Normal:**

1. Seleccionar en el menú “Pacientes”, opción “Gestor de Pacientes”.
2. La aplicación muestra un buscador por nombre, apellido y DNI. Presionando el botón “Buscar” realiza la consulta y lista sus resultados.
3. Seleccione el paciente deseado de la lista y presione “Aceptar” para ver sus datos e historial.

**Alternativas:**

- Del paso 1 al 3 el doctor puede cancelar el proceso de búsqueda.

**AddMark:****Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar registrado correctamente en el dominio (“logueado”).

**Flujo Normal:**

1. Seleccionar la zona donde se desea registrar la marca. Aparece entonces la zona seleccionada ampliada.
2. Pulsar el botón “Seleccionar marca”. Aparece una ventana con imágenes que identifican a cada marca y su nombre.
3. Seleccionar la marca deseada y pulsar el botón “Aceptar”. Esto cierra la ventana y le devuelve a la anterior.
4. Seleccionar la zona del cuerpo en la imagen en la que se quiere registrar la marca. Aparecerá una ventana en la que se podrán añadir comentarios sobre la marca

5. Pulsar el botón “Guardar marcas” para guardar los cambios en la imagen.

**Alternativas:**

- Del paso 1 al 5 el doctor puede cancelar el proceso de añadir marca.

*DeleteMark:*

**Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar registrado correctamente en el dominio (“logueado”).

**Flujo Normal:**

1. Seleccionar la zona del paciente de la que se quieren eliminar marcas. Aparecerá ampliada.
2. A la izquierda de la imagen aparece un historial de las marcas existentes. Seleccionar con el botón derecho del ratón la marca a eliminar. Se le mostrará con un círculo rojo en la imagen.
3. Hacer doble clic con el ratón para abrir la ventana de detalles de la marca.
4. Pulsar el botón “Borrar marca”.
5. Aparece una ventana de confirmación de eliminación. Presionar el botón “Aceptar” para eliminar el paciente definitivamente.
7. Aparece una ventana informativa. Presionar el botón “Aceptar” para cerrarla.

**Alternativas:**

- Del paso 1 al 5 el doctor puede cancelar el proceso de añadir marca.

*PrintMarks*

**Precondiciones:**

El usuario que ejecuta la acción debe ser doctor y además estar registrado correctamente en el dominio (“logueado”).

**Flujo Normal:**

1. Seleccionar la zona del paciente que se desea exportar a PDF. Aparecerá ampliada.
2. Pulsar el botón “Imprimir”.

3. Se abrirá el visualizador de documentos PDF instalado en el sistema operativo que le permitirá guardarlo en la ubicación que desee.

**Alternativas:** no existen.

## 9 Diseño

El diseño realizado se basa en el modelado de patrones, que varían dependiendo de la funcionalidad o sistema según se describe a continuación.

Como un elemento relevante que resulta ser la Base de Datos, se incluye igualmente el diseño de la misma.

### 9.1 Patrón Modelo Vista Controlador

La arquitectura de la aplicación sigue el patrón Modelo Vista Controlador (MVC).

Este patrón se caracteriza principalmente por separar la lógica de negocio de la aplicación de los datos que trata y de la interfaz del usuario en tres partes, tal y como se muestra en la Figura 4.

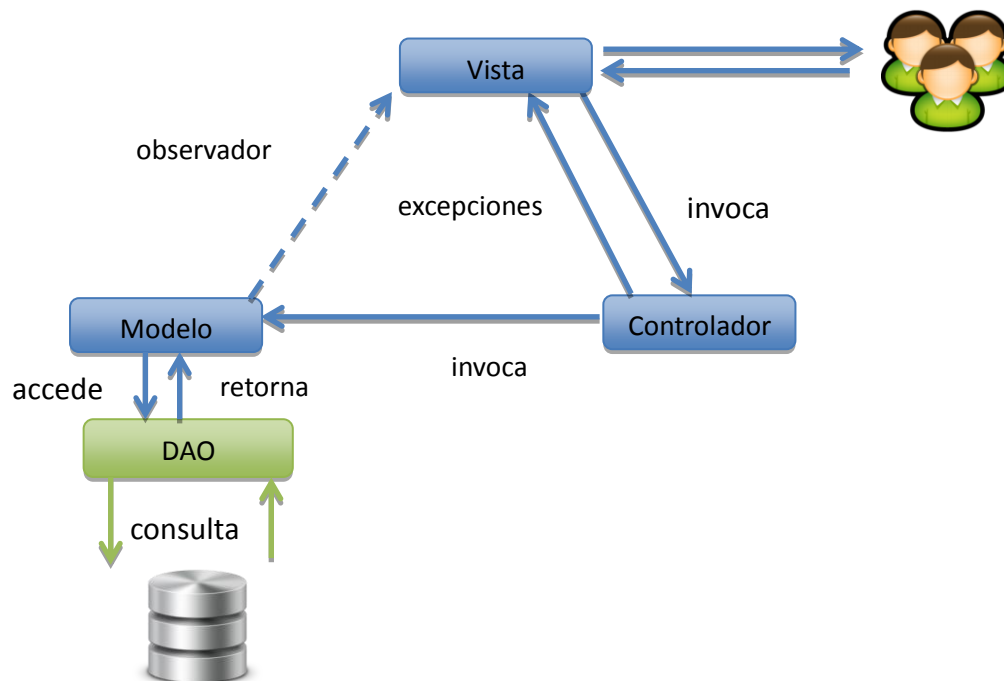
- El modelo: es el encargado de la lógica de negocio de la aplicación. Lleva a cabo las funciones propias de la aplicación e interactúa con la base de datos.
- La vista: es la interfaz de la aplicación con la que trabajará el usuario final.
- El controlador: es el encargado de la gestión y validación de los datos que recibe de la vista, normalmente a través de eventos que la vista genera cuando el usuario interactúa. Éste también puede llevar a cabo algunas acciones simples, pero nunca interactuará con la base de datos.

Se ha elegido este patrón de arquitectura software por las ventajas que presenta a la hora de hacer escalable la aplicación, que tal y como se ha mencionado previamente supone uno de los fundamentos del sistema de cara a su ampliación de futuro. Puesto que la lógica de negocio está separada de la vista, ello permite hacer cambios en la capa de presentación sin tener que modificar ningún aspecto funcional de la aplicación.

Esta estructura resulta muy útil, ya que aunque a priori la aplicación propuesta va a ser utilizada en un primer momento por una única usuaria, cuando en un futuro se decida utilizar DermaSystem en modo multiusuario la interfaz se podrá adaptar a los gustos de

cada usuario en particular, lo que confiere a la aplicación un carácter amigable de suma importancia.

Además de lo expresado anteriormente, desde el punto de vista del diseño de software, la realización de pruebas unitarias de los componentes que forman la aplicación resulta bien sencilla al estar ésta dividida en las partes que se han expuesto anteriormente.



**Figura 4: Esquema general del Modelo Vista Controlador**

Para un mejor entendimiento del funcionamiento interno de la aplicación desde el punto de vista del diseño, a continuación se describe el flujo normal de ejecución dentro del MVC:

1. El Usuario interactúa con la interfaz (Vista)
2. El Controlador recibe datos y eventos generados por la Vista.
3. El Controlador accede al Modelo pasándole los datos procesados y validados que ha recibido de la Vista.
4. El Modelo recibe los datos que le pasa el Controlador y efectúa las funciones correspondientes interactuando con la base de datos mediante el patrón DAO que se describe seguidamente, si así se requiere.

5. Al terminar, notifica a la vista mediante el patrón Observer que se han producido cambios en el Modelo.
6. La Vista recibe la notificación de cambios en el Modelo procediendo a su actualización.

Para comprobar cómo se utiliza este modelo en DermaSystem se puede observar la Figura 5, en la que se indica cómo interactúan las distintas clases de la aplicación con el Modelo, la Vista y el Controlador.

Dentro de cada bloque pueden distinguirse diferentes funciones interrelacionadas. Por otro lado, cada uno de los bloques principales se encuentran relacionados entre sí mediante los correspondientes flujos de información en doble sentido. Conectado con el Modelo se encuentra el patrón DAO, que se estudia en detalle posteriormente, la conexión se establece igualmente mediante un doble flujo de datos. El modelo de patrón DAO se encuentra a su vez conectado a la Base de Datos, que recibe y envía información bajo demanda procedente de las diferentes partes de la aplicación.

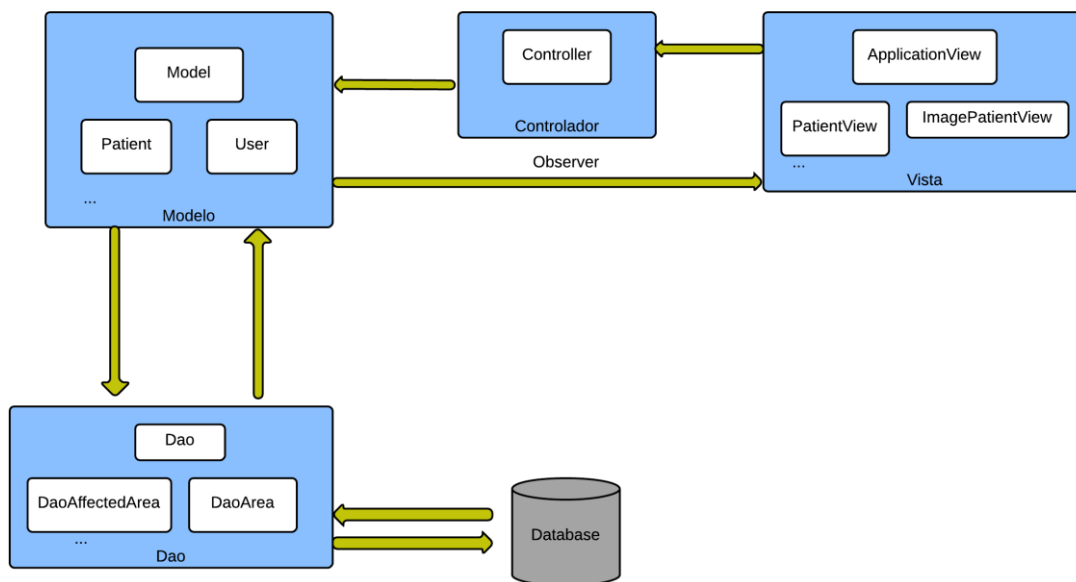


Figura 5: MVC en DermaSystem

## 9.2 Patrón Dao

Para separar la funcionalidad “core” del aplicativo respecto al acceso a la Base de Datos se ha utilizado el patrón DAO (**D**ata **A**ccess **O**bject), Figura 6. Gracias a él podemos abstraer y encapsular los accesos a la Base de Datos, gestionar sus conexiones y obtener los datos necesarios cuando éstos se requieran y a petición del usuario.

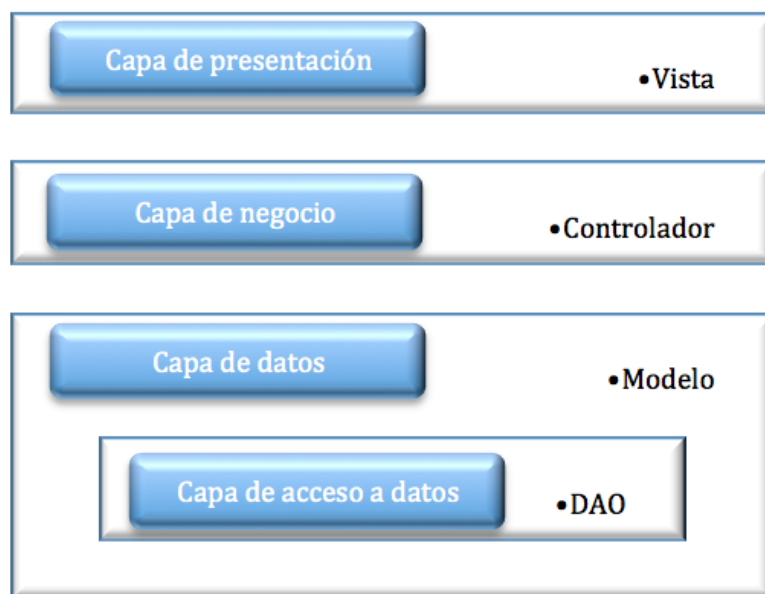


Figura 6: Esquema general del patrón DAO dentro del modelo

Para su implementación se ha utilizado el patrón Factoría Abstracta, que se describe a continuación.

## 9.3 Patrón Factoría Abstracta

Este patrón proporciona una interfaz para crear familias de objetos que dependen entre sí, sin especificar sus clases concretas.

Así pues, se ha implementado una interface en lenguaje java llamada DAO para el acceso a la Base de Datos que declara los siguientes métodos abstractos:

1. `public void create(Object object) throws Exception`: crea un nuevo objeto y lo guarda en la base de datos.

2. `public Object get(Object object) throws Exception`: devuelve un objeto de la base de datos.
3. `public boolean delete(Object object) throws Exception`: borra un objeto de la base de datos.
4. `public boolean update(Object object) throws Exception`: actualiza un objeto de la base de datos.
5. `public ArrayList<Object> getAll() throws Exception`: devuelve una lista de todos los objetos almacenados en la base de datos del tipo deseado.

Es bien conocido en el ámbito del diseño de Bases de Datos, que éstas se estructuran normalmente en tablas. Pues bien, por cada tabla existente en la base de datos se ha creado un DAO propio que implementa los métodos de la interface. Así de esta forma se identifican los siguientes:

- DaoAffectedArea
- DaoArea
- DaoMark
- DaoPatient
- DaoRole
- DaoUser

Gracias a esto, el Modelo interactúa con la Base de Datos sin necesidad de conocer el destino final de la información que procesa.

## 9.4 Patrón Singleton

El patrón Singleton garantiza que una clase sólo tenga una instancia, proporcionando un punto de acceso global a dicha instancia.

Su funcionamiento es muy sencillo y se puede resumir de acuerdo a los tres puntos siguientes:

1. Ocultar el constructor de la clase Singleton, para que no se puedan crear instancias adicionales.
2. Declarar en la clase Singleton una variable miembro privada que contenga la referencia a la instancia única que queremos gestionar.

3. Dotar a la clase Singleton de una función mediante la que se pueda acceder a la única instancia gestionada por el Singleton.

```
static private DaoAffectedarea daoSingleton = null;
static public DaoAffectedarea getDao()
{
    if (daoSingleton == null)
    {
        daoSingleton = new DaoAffectedarea();
    }
    return daoSingleton;
}
```

Figura 7: Patrón Singleton

## 9.5 Patrón Observer

Este patrón se ha diseñado para monitorizar el estado de un objeto en un programa. Se trata de un sistema cuya función consiste en ser un sistema de detección de eventos en tiempo de ejecución. Su uso es muy concreto: varios objetos necesitan ser notificados mediante un evento y cada uno de ellos decide cómo reaccionar cuando este evento se produce.

Este patrón es muy útil en el presente proyecto al utilizar un MVC, cuyo diseño tiene prevista la incorporación de diversas interfaces.

Para aclarar esta idea mejor, en los puntos siguientes se explica su funcionamiento:

1. Tal y como se muestra en la declaración de la , el objeto modelo extiende de la clase Observable. Cada vez que se realiza alguna conexión a la base de datos, el modelo cambia su estado (a “modificado”) informando a todos los Observadores acerca de su nuevo estado a través de los eventos programados al respecto.

```
public class Model extends Observable
```

Figura 8: Declaración de la clase Model

2. Las vistas, por el contrario, extienden de la clase Observador (propia del patrón). Permanecen a la espera, “escuchando” eventos que les puedan llegar de otras partes de la aplicación. Gestionan estos distintos tipos de eventos o los descartan. Su esquema básico se detalla en la Figura 9.

```
public class ApplicationView implements Observer{  
  
    private static Model model;  
  
    @Override  
    public void update(Observable observable, Object object) {  
        Event event = (Event) object;  
        if(event.getEventType().equals(Constant.DAO_ERROR)){  
            EventDaoErrors e= (EventDaoErrors) event;  
            JOptionPane.showMessageDialog(null, e.getDaoError(), null, JOptionPane.ERROR_MESSAGE);  
        }  
    }  
}
```

Figura 9: Comportamiento de las Vistas

3. Cada vez que una vista se “suscribe” como Observadora se añade una referencia de ésta en un “array”, dentro del patrón Observer, de esta forma el modelo informa a todas las vistas suscritas a él. El esquema básico de implementación es el que se muestra en la Figura 10.

```
this.setChanged();  
this.notifyObservers(new EventDaoErrors(e.getMessage()));
```

Figura 10: “Suscripción” de las vistas.

## 9.6 Varios Sistemas Operativos

Gracias a que DermaSystem ha sido programado en lenguaje Java, esta aplicación puede ejecutarse en varios sistemas operativos:

- Windows
- MacOs
- Linux

El requisito indispensable para su correcto funcionamiento es que en el computador donde se vaya a instalar la aplicación se tenga instalada la versión 1.6 o superiores de la Máquina Virtual de Java.

Además, al haber sido DermaSystem desarrollado con Java Swing, la vista de la interfaz de usuario se adaptará al sistema operativo en el que se ejecute lo que le permitirá fusionarse mejor con el estilo gráfico que utilice el usuario.

## 9.7 Base de Datos

### 9.7.1 Consideraciones previas

Debido a la imposibilidad de conseguir la estructura de la Base de Datos del Hospital Gregorio Marañón ha sido necesario crear una completamente nueva. Este hecho implica que el contenido de la nueva no estará sincronizado con la del Hospital.

Sin embargo, el diseño de esta nueva base de datos se ha realizado de la forma más sencilla, eficiente y robusta posible para que en un futuro toda la información creada y almacenada por la doctora como usuaria final del sistema pueda exportarse e incluirse en una Base de Datos de mayor capacidad.

### 9.7.2 Elección de la base de datos

Se ha llevado a cabo un análisis sobre varios Sistemas Gestores de Bases de Datos (de aquí en adelante SGBD) existentes junto con las posibilidades más relevantes que ofrecen. Asimismo se han establecido una serie de requisitos que deben cumplirse teniendo en cuenta las necesidades de la aplicación actual y su ampliación prevista de futuro:

- Al ser un proyecto que carece de financiación, un requisito indispensable ha sido que el SGBD no tuviese ningún coste, hecho que reduce bastante la lista de posibilidades quedando excluidos sistemas muy potentes como Advantage Database, Oracle, IBM DB2, PervasiveSQL, Microsoft SQL Server o NexusDB.
- El SGBD convendría que fuese multiplataforma. DermaSystem es una aplicación que puede ejecutarse en varios sistemas operativos, por lo que usar un gestor multiplataforma facilitaría la implantación de DermaSystem.

Llegados a este punto se plantean dos opciones:

1. La versión gratuita de Oracle (Oracle Express Edition) que cumplía con todos los requisitos mencionados anteriormente. Sin embargo, a este SGBD vienen unidas una serie de limitaciones en el uso (sólo puede utilizar un procesador del servidor donde esté instalada y un máximo de 1 Gb de memoria RAM) y sobre todo en el almacenamiento (un máximo de 4 GB de datos). Esto último representaba un serio problema ya que a la hora de importar datos se podría llegar a una falta de espacio físico, lo que obligaría a migrar a la versión de pago de Oracle.
2. Usar un SGBD de software libre: el elegido fue MySQL debido a que es ampliamente conocido por su simplicidad y notable rendimiento. Pese a tener algunas carencias en comparación con otros SGBD, como por ejemplo en las subconsultas o integridad de los datos, lo hacen ideal para la integración con DermaSystem ya que estas carencias se suplen con consultas sencillas y una buena implementación en el código de la aplicación respectivamente, tal y como se ha hecho.

### 9.7.3 Diseño de la Base de Datos

Se han creado seis tablas:

1. *AffectedArea*: en esta tabla se almacenan las coordenadas de las marcas para cada paciente.
2. *Area*: esta tabla almacena las áreas en las que está dividida la imagen prediseñada que se muestra al usuario. Por ejemplo: lateral izquierdo de la cara, lateral derecho de la cara, cuerpo delantero, cuerpo trasero, etc.
3. *Mark*: esta tabla almacena las afecciones que se marcarán en las imágenes. Su contenido ha sido diseñado por la doctora, inspiradora de este sistema y de acuerdo con las necesidades en dermatología.
4. *Patient*: esta tabla almacena todos los datos relacionados con los datos personales de los pacientes.
5. *Role*: esta tabla almacena los tipos de usuarios que existen en la aplicación, a saber:
  - Administrador
  - Doctor.

6. *User*: esta tabla almacena los usuarios (y datos personales) que están registrados en la aplicación y por tanto dichos usuarios podrán usarla en función del rol que tengan.

En la Figura 11 se muestra el diagrama de entidad relación en el que se pueden ver cada atributo de las tablas, su tipo y la cardinalidad.

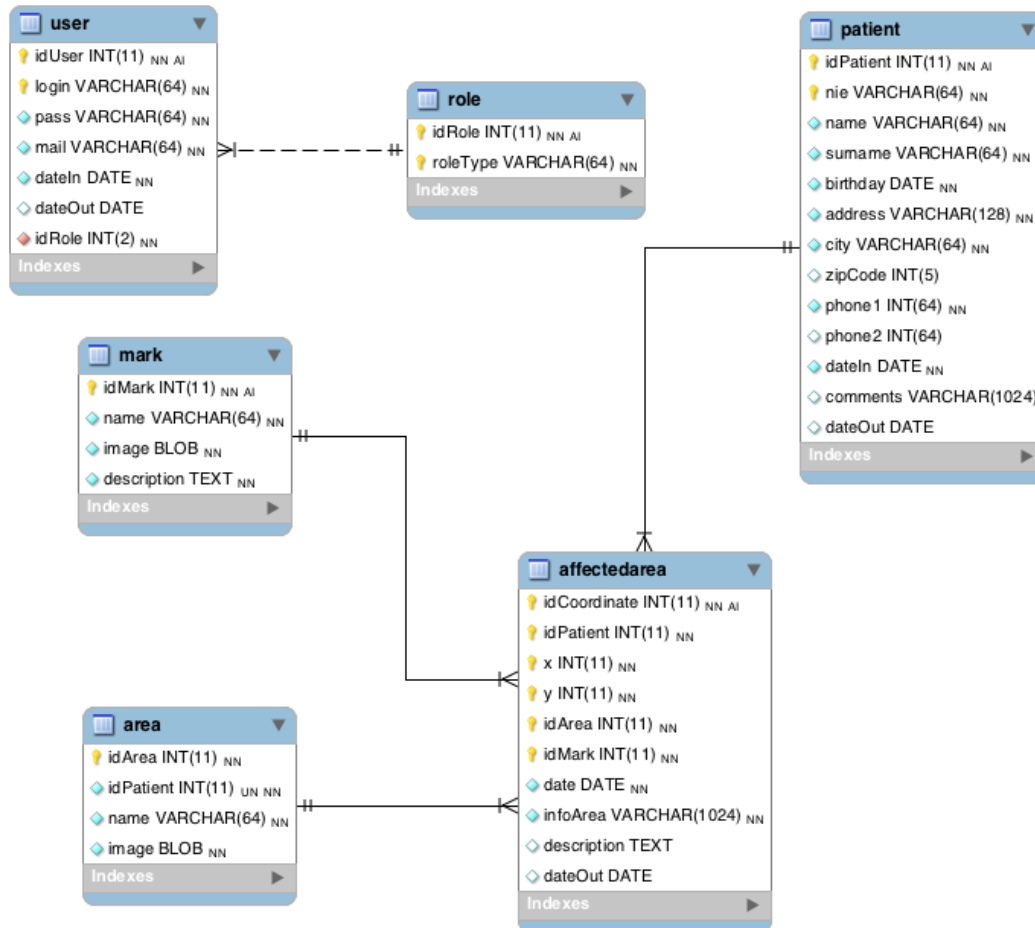


Figura 11: Base de Datos: diseño y esquema de las tablas

### 9.7.4 Hospedar la base de datos

La base de datos ha estado alojada de forma local durante el desarrollo e implementación de la aplicación. Para ello se ha usado Wampp, en el caso de los usuarios que han trabajado bajo el entorno Windows y Xampp, en el caso de aquellos que usaron MacOS.

Para la última fase de desarrollo y pruebas la Base de Datos se ha alojado en remoto, en un servidor estadounidense gratuito llamado MySQL free.

## 10 Diagramas de secuencia

Para entender mejor cómo se desarrolla la ejecución de los casos de uso se pueden observar los diagramas de secuencia de la Figura 12 y la Figura 13 mostradas a continuación.

Concretamente, se presentan dos diagramas (añadir marca y editar paciente) ya que éstos se consideran los más completos y desde los que se puede observar mejor cómo interactúan los distintos objetos de la aplicación.

### 10.1 Diagrama añadir marca

1. El usuario solicita a la aplicación añadir una nueva marca. La vista, devuelve una ventana que muestra las marcas disponibles para seleccionar.
2. El controlador pasa al modelo los datos que el usuario ha seleccionado en la ventana.
3. El modelo llama al DAO. Éste accede a la base de datos y realiza la consulta “Insert” que guarda la información, creando así la nueva marca.
4. Una vez guardada la información en la base de datos se actualiza el historial de marcas del paciente (clase log).
5. El modelo notifica los cambios a la vista y ésta muestra una ventana de confirmación.

### 10.2 Diagrama editar paciente

1. El usuario solicita a la aplicación editar un paciente. La vista habilita los campos de texto para poder modificar la información.
2. El controlador pasa al modelo los datos que el usuario ha introducido.
3. El modelo llama al DAO. Éste accede a la base de datos y realiza la consulta “Update” que guarda la información, actualizando así los datos del paciente.
4. El modelo notifica los cambios a la vista y ésta muestra una ventana de confirmación.

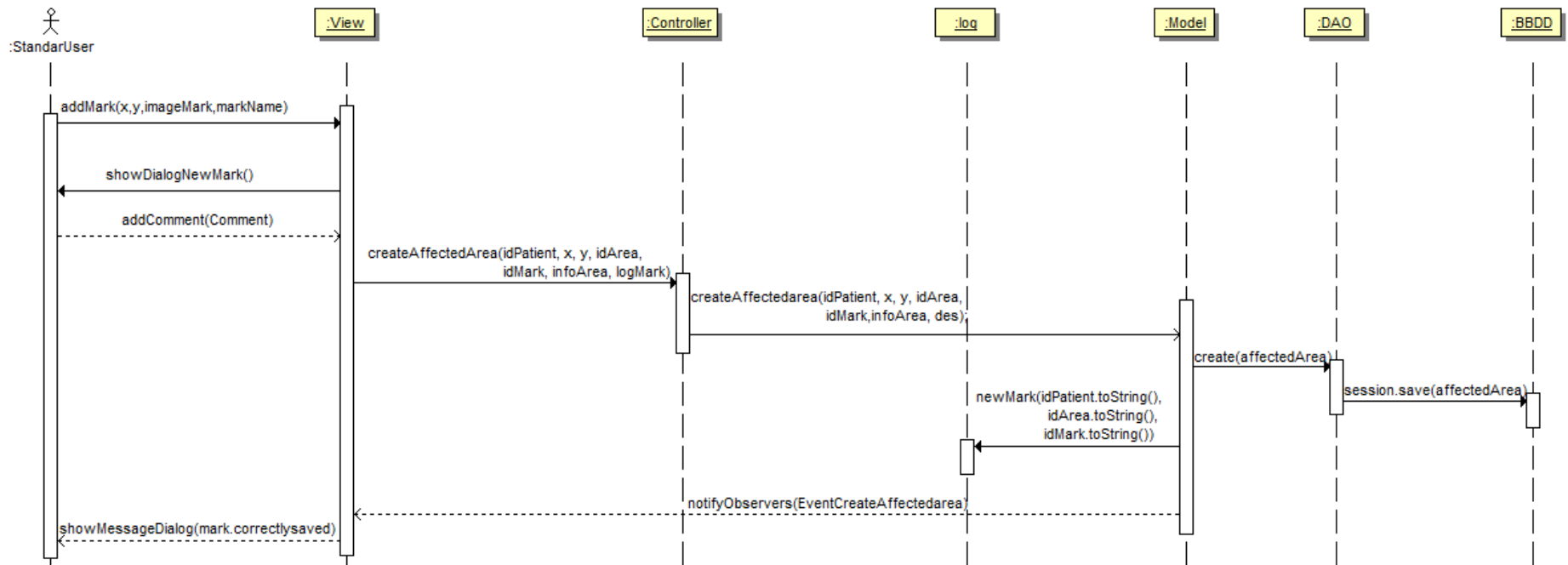


Figura 12: Diagrama de secuencia añadir marca

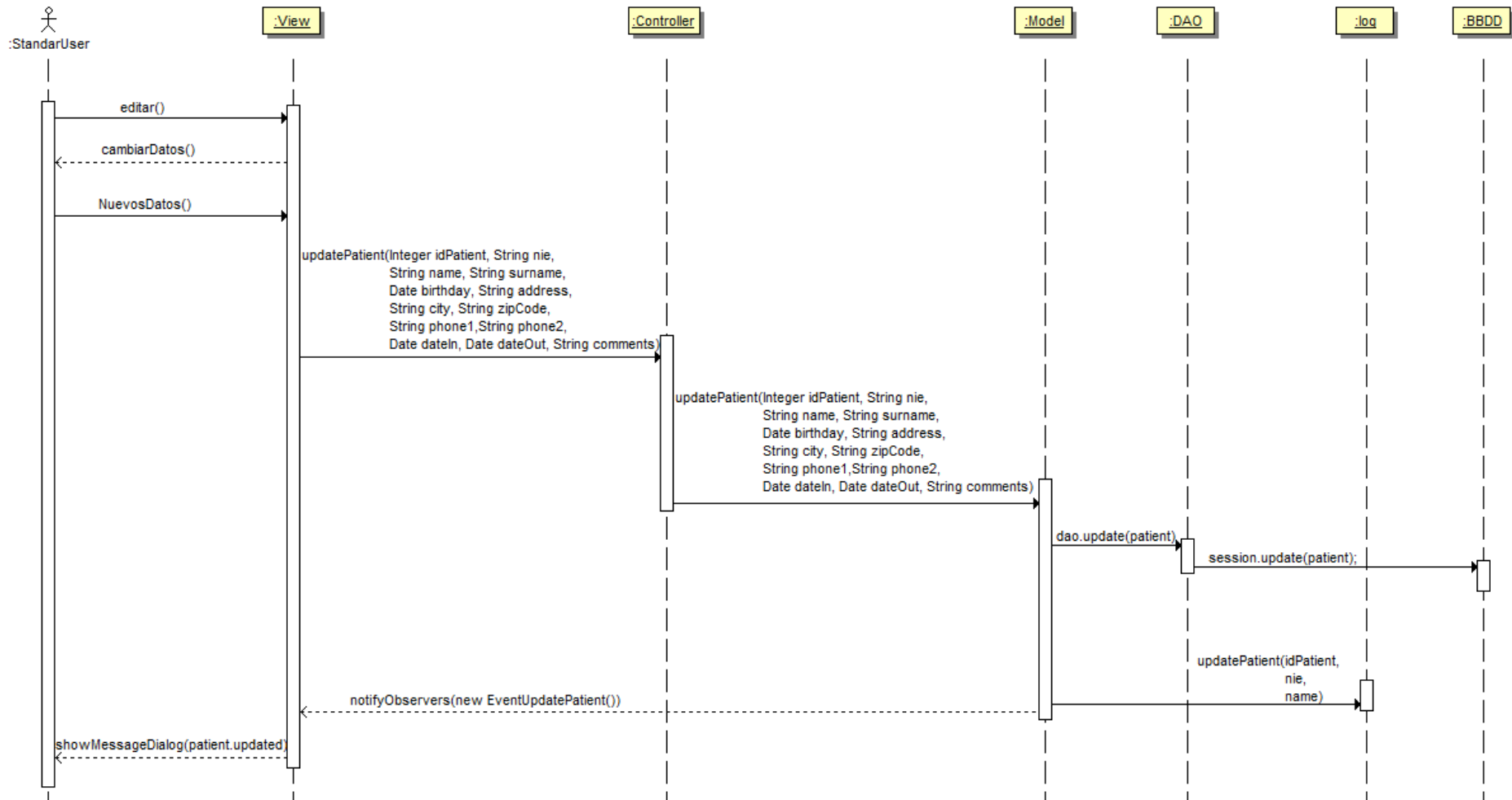


Figura 13: Diagrama de secuencia editar paciente

# 11 Diagrama de clases

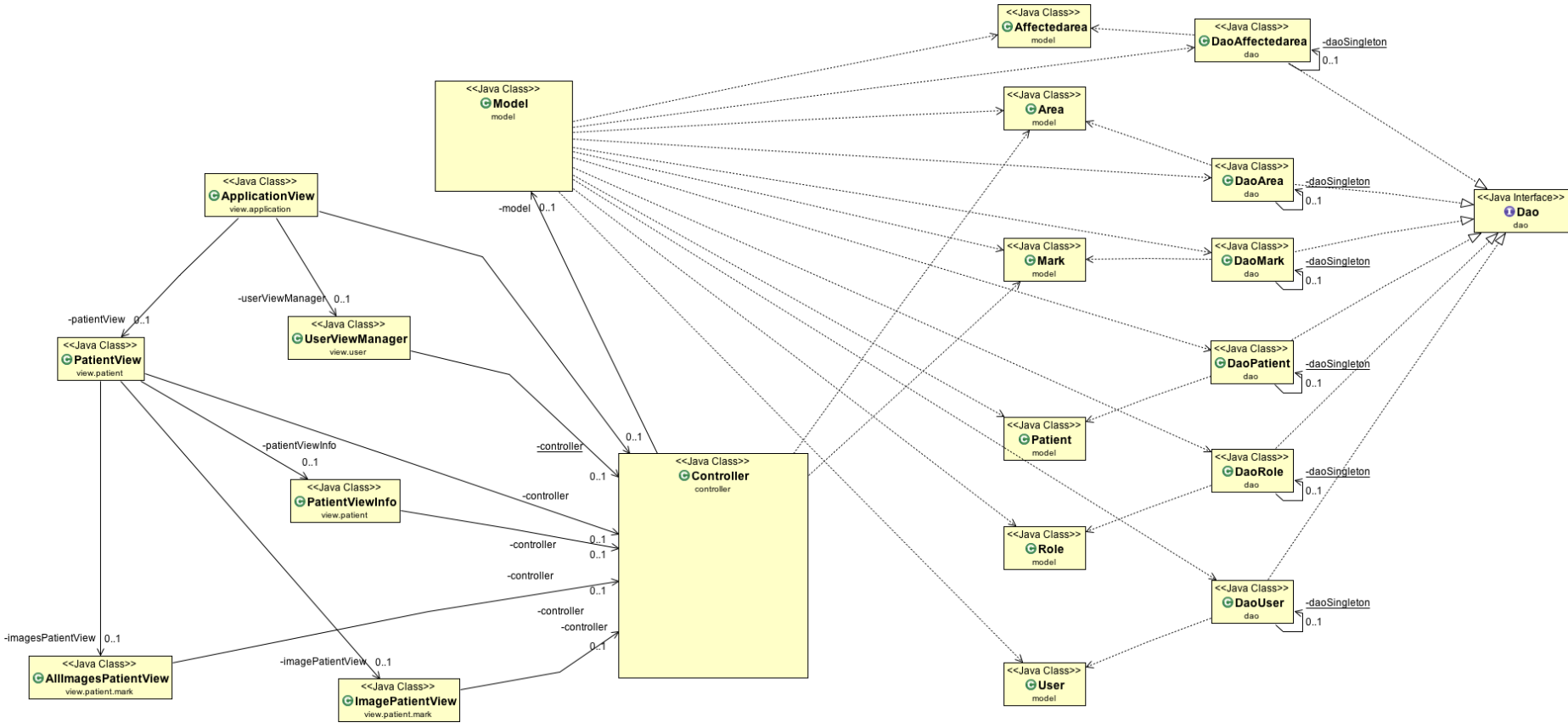


Figura 14: Diagrama de clases

## 12 Tecnologías utilizadas

A continuación se describen los aspectos básicos relacionados con la implementación del diseño realizado previamente.

Concretamente, se describen las tecnologías utilizadas así como la configuración de la aplicación.

### 12.1 Java

DermaSystem ha sido enteramente programado en Java, ya que, al tratarse de una aplicación con la que va a interactuar el usuario, era importante que el lenguaje seleccionado permitiese la programación dirigida por eventos.

Por tanto, este “tipo” de programación ha permitido controlar de una manera eficiente todas las actividades que llevan a cabo el usuario por un aparte y la aplicación por otra.

### 12.2 Java Swing

Java Swing es la principal herramienta de Java para crear interfaces gráficas de usuario, más conocidas por sus siglas GUI, del inglés *graphical user interface*. Además es un framework basado en el Modelo-Vista-Controlador, patrón que, como se ha indicado previamente, sigue la arquitectura de DermaSystem. Su popularidad y gran número de componentes lo han hecho perfecto para desarrollar una interfaz gráfica fácil e intuitiva para el usuario.

### 12.3 JCalendar

JCalendar es una librería de java que muestra un calendario en el que poder seleccionar una fecha de una forma gráfica y cómoda.

Está compuesto por Java Beans, tales como JDayChooser, JMonthChooser y JYearChooser. Todos estos Beans tienen propiedades tales como iconos o configuraciones que permiten que puedan ser utilizados fácilmente en constructores GUI. También forma parte del paquete JDateChooser, un Bean compuesto de un IDateEditor (para la edición de fecha) y un botón para la apertura del JCalendar que permite la selección de la fecha.

## 12.4 Hibernate

Hibernate es una herramienta de Mapeo Objetos Relacional, más conocida como ORM, que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.

Gracias a la ayuda de Hibernate se soluciona el problema que existe entre los datos con los que trabaja la aplicación, que, en el caso de DermaSystem, serían los objetos y los datos que posteriormente se guardarán en la base de datos.

Además, el uso de la Base de Datos es totalmente transparente pudiendo cambiar dicha base sin necesidad de modificar código en la aplicación, tan sólo se modificarían los ficheros de configuración de Hibernate y se añadirían las consultas deseadas en las clases DAO. Esto es muy útil a la hora de ampliar las funcionalidades de DermaSystem.

Su funcionamiento es sencillo:

- Hibernate asocia a cada tabla de la base de datos un Plain Old Java Object, más conocido como POJO. Un POJO es similar a una clase, con propiedades accesibles mediante métodos setter y getter.
- Para poder asociar el POJO a su tabla correspondiente en la base de datos, Hibernate usa ficheros XML, en los que se declaran las propiedades del POJO y sus correspondientes nombres de columna en la base de datos para cada tabla. También se puede incluir la asociación de tipos de datos, referencias, relaciones x a x con otras tablas, etc.

Hibernate incorpora también un lenguaje con el que realizar consultas a la base de datos y recibe el nombre de Hibernate Query Language HQL.

El uso de HQL nos permite usar un lenguaje intermedio que, según la base de datos que usemos y el dialecto que especifiquemos será traducido al SQL propio de cada base de datos de forma automática y transparente.

Esta última funcionalidad no se ha incorporado a DermaSystem ya que se ha considerado que el lenguaje SQL está ampliamente extendido y la mayor parte de bases de datos lo soportan. Al carecer de la estructura de la base de datos original del Hospital Gregorio Marañón y considerar que DermaSystem puede ser ampliada en un futuro próximo no se ha querido complicar la manera de hacer las consultas.

## 12.5 Assembla

Assembla es una empresa que provee herramientas de colaboración y seguimiento de tareas basadas en la nube para organizar y administrar proyectos de código abierto y comerciales para el desarrollo de software. Para este proyecto se ha usado su producto Espacios de Trabajo (Project Workspaces) que ofrece, además de lo anteriormente citado, control de versiones o Subversion.

### *Subversion*

Es un sistema de control de versiones. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como *svn* por ser el nombre de la herramienta utilizada en la línea de comando.

Subversion (Tigris.org Open Source Software. (2013). *Subversion*) puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. Esto permite a los desarrolladores trabajar sobre la misma versión del proyecto e ir actualizando sus fuentes y que éstas estén disponibles para el resto del equipo.

## 12.6 IDEs

Para programar DermaSystem se ha usado Eclipse, un es un programa informático compuesto por un conjunto de herramientas de programación. Es de código abierto y multiplataforma lo cual ha sido indispensable para el desarrollo de este proyecto ya que no todos sus componentes han usado el mismo sistema operativo.

Las versiones utilizadas fueron Indigo y Juno, esto ha sido debido a que para el sistema operativo Mac la última versión de Eclipse (Juno) presentaba algunos errores (bugs) que dificultaban su uso.

## 12.7 Subclipse

Subclipse (Tigris.org Open Source Software. (s.f.). *Subclipse*) es una extensión, conocida en terminología inglesa como “plug-in”, que proporciona soporte para trabajar con Subversion en Eclipse. Lo que ha permitido trabajar a cada miembro del equipo de desarrollo del proyecto individualmente y actualizar su trabajo pudiéndolo subir a Subversion.

## 12.8 iText

Se trata de una librería para Java que permite crear un archivo PDF imprimiendo en dicho archivo los datos que se deseen.

iText es de código abierto, con una licencia Open-Source AGPL y está ampliamente documentada. También distribuyen licencias de pago con múltiples beneficios, sin embargo, esta opción no ha sido considerada debido a que DermaSystem es un proyecto de fin de carrera. No obstante, en un futuro se podría considerar la adquisición de otras licencias si el manejo de datos lo requiere

Se ha elegido el formato PDF porque es un estándar que todos los sistemas operativos actuales son capaces de leer. Igualmente permite ser guardado, imprimido, firmado de forma digital así como personalizado con comentarios.

## 13 Implementación

### 13.1 Internacionalización (i18n)

Conscientes de que vivimos en un mundo globalizado, se hace prácticamente imprescindible que cualquier aplicación informática posea la mayor difusión y aceptación posible, particularmente en lo que respecta a los idiomas de mayor difusión, por ello DermaSystem se ha diseñado de forma que está disponible en dos idiomas: Español e Inglés.

Para ello se ha hecho uso de los paquetes de internacionalización de Java, que permiten tal diseño:

- *java.util.Locale* (Oracle. *Locale*): este paquete permite especificar la región geográfica, política y cultural en la que se ejecuta la aplicación.
- *java.util.ResourceBundle* (Oracle. *ResourceBundle*): define una manera estándar para acceder a las traducciones en java. Junto con este paquete se utilizan clases o archivos de propiedades (de extensión conocidas como *.properties*) que representan las traducciones. Están formados por atributos (en el caso de las clases) o claves (en el caso de los archivos de propiedades). Cada clase o archivo *.properties* contiene exactamente los mismos elementos pero traducidos relativos a los distintos idiomas disponibles.

En DermaSystem se han utilizado archivos de propiedades ya que en el supuesto de querer añadir otros idiomas resulta más sencillo pues no hace falta compilar estos

archivos, cosa que con las clases esta acción se hace absolutamente necesaria. El método que se ha seguido para la internacionalización (i18n) ha sido el siguiente:

1. Se ha creado una clase *AppLanguage* cuyo único atributo es de tipo *ResourceBundle*, si bien con los modificadores *final* (para asegurar que su valor no se va a modificar durante la ejecución de la aplicación) y *static* (para poder acceder a él sin necesidad de crear instancias).
2. Cuando se instancie la clase se llamará a aquella que gestiona la configuración de la aplicación, que se explica en más adelante en el apartado siguiente. Se crea un objeto de tipo *Locale* que guardará el valor que contenga la clave destinada a identificar el idioma del usuario, que en este caso se llama “*user.locale*” almacenada en un archivo de propiedades *.properties*.
3. Una vez especificado el idioma gracias al objeto de tipo *Locale* se asigna al atributo de la clase *AppLanguage* de tipo *ResourceBundle* su valor. Éste vendrá definido por el nombre del archivo *.properties* que contiene las traducciones y el objeto *Locale* creado anteriormente.

Para entender mejor este proceso se muestra una captura del código de la clase *AppLanguage*, que se muestra en la Figura 15.

```
/**
 * @author Sara Llorente Navarro
 * @author José Carlos Olivares Conde
 * @author Carlos Manuel Muñoz Díaz
 */

package resources.lang;

import java.util.Locale;

public class AppLanguage
{
    private final static ResourceBundle bundle;

    /**
     * Recupera el bundle del idioma configurado
     */
    static
    {
        Locale locale = new Locale(AppConfiguration.getBundle().getString("user.locale"));
        bundle = ResourceBundle.getBundle("resources.lang.language", locale);
    }

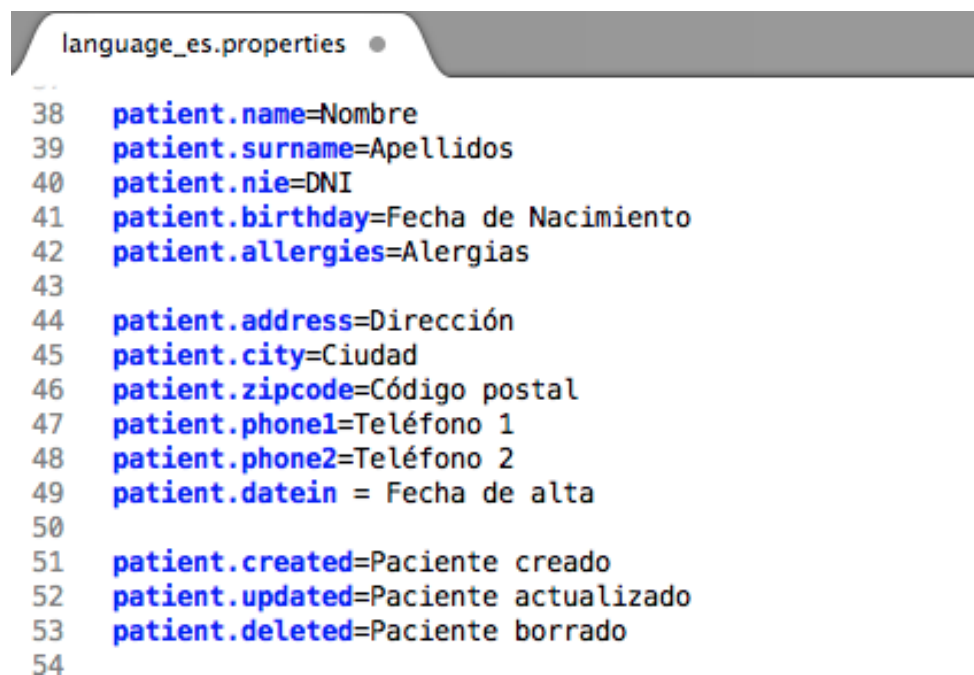
    public static ResourceBundle getBundle() {
        return bundle;
    }
}
```

Figura 15: Clase *AppLanguage*

## Archivos .properties

El nombre de estos archivos resulta ser muy importante para identificar el idioma de la aplicación, por ello seguimos el siguiente procedimiento.

- Se sigue el estándar ISO 639-1 (ISO - International Organization for Standardization. (s.f.)) que proporciona un código de dos letras diseñado para representar la mayoría de los idiomas más importantes del mundo.
- De acuerdo con ISO 639-1 (ISO - International Organization for Standardization. (s.f.)), el estándar de java para nombrarlos resulta ser: *nombreBase\_códigoLengua*. El nombre base ha sido “*language*”, por lo que para el Español, quedaría: “*language\_es.properties*”.



```
language_es.properties
--
38 patient.name=Nombre
39 patient.surname=Apellidos
40 patient.nie=DNI
41 patient.birthday=Fecha de Nacimiento
42 patient.allergies=Alergias
43
44 patient.address=Dirección
45 patient.city=Ciudad
46 patient.zipcode=Código postal
47 patient.phone1=Teléfono 1
48 patient.phone2=Teléfono 2
49 patient.datein = Fecha de alta
50
51 patient.created=Paciente creado
52 patient.updated=Paciente actualizado
53 patient.deleted=Paciente borrado
54
```

Figura 16: Contenido del archivo de idioma

## 13.2 Configuración de la aplicación

En conjunto con la internacionalización se ha creado también una clase `AppConfiguration` con las mismas características que `AppLanguage`. Esta clase tiene como finalidad cargar el archivo de configuración `.properties` de la aplicación, a la vez que implementa métodos que permiten modificar este archivo.

A priori almacena exclusivamente el idioma seleccionado por el usuario, lo que permite a `DermaSystem` guardar las preferencias de idioma del usuario.

## 13.3 Impresión de marcas PDF

Como complemento a la gestión de las marcas de los pacientes, se ha añadido una función que permite imprimir en un documento PDF la vista de un área específica del cuerpo de un paciente. Ello ayudará al dermatólogo a exportar información almacenada en DermaSystem al historial físico del paciente si así lo desea.

En concreto, se han utilizado unos datos mínimos del paciente, el área actualmente seleccionada con sus marcas y una lista de todas las marcas que aparecen en dicha área.

## 13.4 Interfaces

Para facilitar la escalabilidad del proyecto, la reutilización de código y poder realizar cualquier cambio en el aspecto de la interfaz sin que ello provoque tener que realizar grandes cambios en otras partes del código de la aplicación, se ha optado por un diseño de interfaces modular basado en unidades o módulos construidos con *JInternalFrames* y *JDialogs*.

Aunque existen relaciones entre estos *JInternalFrames* necesarias para la construcción de la aplicación, se ha prestado especial atención a la independencia de estos módulos, de modo que en un futuro, el diseñador pueda modificar cada uno de ellos de manera individual sin que dicha modificación afecte al resto de módulos y menos aún a la aplicación en general.

Para comprender mejor la estructura de las vistas, podemos imaginar las interfaces como un patrón en árbol cuyas ramas son los módulos (*JInternalFrame*).

La estructura de interfaces se ha diseñado de manera que pueda resultar lo más fácil e intuitiva a un usuario no experto, con formularios sencillos, sin excesivos botones ni menús, para no incrementar en modo alguno la complejidad del sistema con la consiguiente pérdida de amigabilidad.

A continuación, en las siguientes subsecciones, se describen las relaciones entre las interfaces.

### 13.4.1 ApplicationView

La vista principal es la interfaz *ApplicationView*, cuya función es la de ejecución de la aplicación y la creación y gestión del resto de *JInternalFrames*. Es la única interfaz que implementa un *Frame* y muestra los menús de *DermaSystem*.

En función de los permisos que tenga asignados el usuario cuando se ha registrado e identificado en el dominio (“logueado”), esta vista oculta o desactiva algunas opciones; aunque, incluso sin realizar el log-in, permite cambiar el idioma de la aplicación o visualizar el JDialog AboutTo que funciona como ventana de información.

Gestiona los siguientes módulos de la aplicación:

1. *pacientes* con su interfaz PatientView
2. *usuarios* con su interfaz UserView.

Asimismo, participa en la creación de nuevos pacientes y usuarios cuyas interfaces son PatientViewCreate y UserViewCreate respectivamente.

Cabe destacar que UserLogin es el único JInternalFrame que se cierra realmente ya que la aplicación no se ha concebido como un uso similar al de una web, y no permite volver a realizar un nuevo “log-in” sin cerrar y volver a abrir la aplicación.

El resto de JInternalFrames no se cierra, sino que se ocultan y desactivan su “focus” para dar paso a otros JInternalFrame.

### 13.4.2 UserView

Una vez que el usuario con permisos de administrador ha completado una acción de “log-in”, se puede acceder al módulo de usuarios, gestionado por la interfaz UserView, que en esencia lista y edita usuarios ya existentes o nuevos.

### 13.4.3 PatientView

Debido a que el objetivo de la aplicación reside en gestionar pacientes y las marcas asociadas a ellos, el módulo de pacientes es el más complejo.

PatientView es un JInternalFrame siendo el único que añade un nivel más, creando y separando los "sub-módulos" de información de pacientes y la información de marcas por paciente.

Esta vista permite buscar (vista PatientViewSearch), listar y seleccionar los pacientes, o incluso realizar la búsqueda de pacientes. Una vez se seleccione el paciente es cuando se desciende un nivel más, creando tres JInternalFrames:

- PatientViewInfo
- AllImagesPatientView
- ImagePatientView.

Por decirlo de alguna manera, las marcas del paciente actúan como un “sub-módulo”, casi independiente, cuya única relación es el identificador del paciente.

El “sub-módulo” de marcas lo forman dos unidades del tipo `JInternalFrame`, `AllImagesPatientView` y `ImagesPatientView`, de los que siempre existe uno visible, mientras el otro permanece oculto.

- *AllImagesPatientView*: permite navegar entre las imágenes prediseñadas del cuerpo humano, una vez seleccionada la zona, es `ImagePatientView` el `JInternalFrame` que muestra con más detalle el área zona seleccionada, así como las marcas existentes en ella.
- *ImagesPatientView* es la que dispone de toda la lógica para mostrar las marcas asociadas a una zona del paciente, creando `JDialogs` para editar o crear nuevas marcas.

Para facilitar la comprensión del diseño y toda la estructura de interfaces, en la página siguiente se muestra la Figura 17 donde podemos observar el diagrama de vistas, que explica de forma visual todo lo explicado anteriormente.

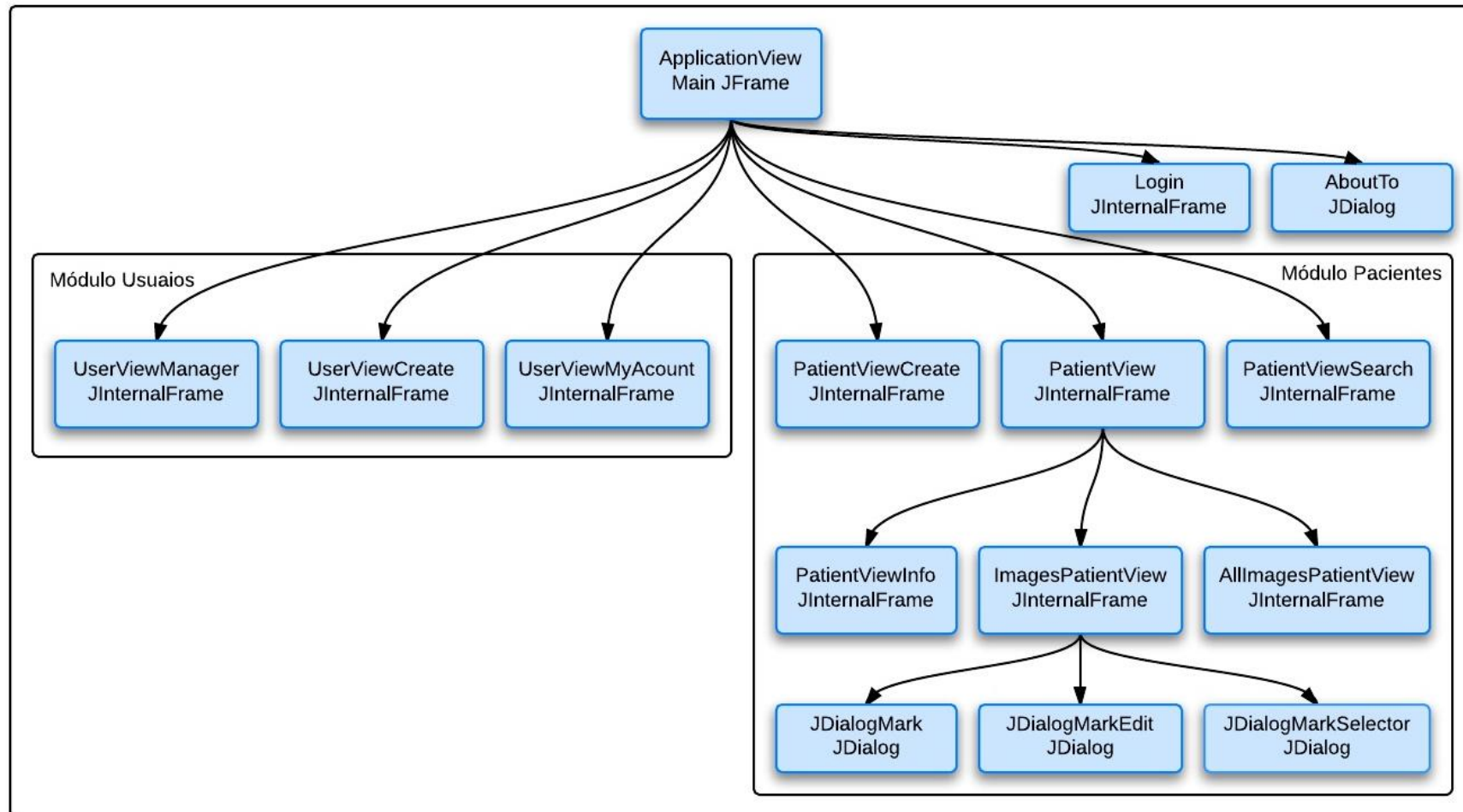


Figura 17: Diagrama de vistas

## 14 Conclusiones y futuro

### 14.1 Conclusiones

DermaSystem es un sistema informático específicamente diseñado para la gestión de pacientes en dermatología, cuya finalidad es servir de ayuda al especialista en la gestión de pacientes, facilitándole de forma gráfica y textual información de gran utilidad.

Se ha diseñado bajo una metodología y estructura apropiadas para conseguir una aplicación robusta y útil. Su interfaz gráfica es intuitiva y amigable. Ha sido diseñada para el uso diario con repetitivos cambios, por lo que la totalidad de las funciones del sistema están disponibles “a golpe de clic”.

Se ha incluido un sistema de autenticación de usuarios que garantizará la seguridad de los datos evitando accesos indebidos. Además, permite separar la administración de la aplicación de las funciones propias que utilizará el especialista dermatólogo.

Con todo esto se ha conseguido una aplicación destinada a facilitar el día a día en las consultas dermatológicas. Inicialmente, inspirada en las ideas de la doctora María Isabel Longo se ha diseñado e implementado con previsión de futuro para posibles ampliaciones.

En este momento se dispone de una aplicación que permitirá manejar los historiales clínicos de los pacientes dermatológicos y, lo más importante, se posee un sistema de marcado de las afecciones dermatológicas que será estándar para todos los diversos pacientes, en principio según las necesidades previstas por la mencionada doctora.

### 14.2 Futuro de la aplicación

Pese a que DermaSystem ha sido un proyecto de fin de grado podemos añadir que ha establecido las bases para su posterior evolución. En efecto, ha sido diseñado para que se continúe en un futuro, de forma que los ingenieros informáticos que participen en ese nuevo desarrollo puedan entender fácilmente hasta dónde ha llegado el alcance de este proyecto, pudiendo así añadirle funcionalidad sin el esfuerzo de estudiar la implementación anterior o iniciar el proceso desde cero.

Se ha proporcionado una amplia documentación del proyecto, no sólo reflejada en esta memoria, sino también en el JavaDoc que se ha generado, donde se comenta el código de la aplicación.

A continuación se indican una serie de ideas sobre las funcionalidades y aspectos que en un futuro podrían añadirse a DermaSystem como continuación y mejora del mismo. Destacan las siguientes.

- Sistema de manejo de imágenes reales de las lesiones de los pacientes.
- Sistema de procesamiento de imágenes con el fin de analizar la evolución de las mismas mediante técnicas de análisis de texturas, tamaño, forma u otros elementos de interés.
- Marcado automático de imágenes tras su procesamiento y análisis de su evolución.
- Posibilidad de añadir nuevas marcas gráficas relativas a diversas lesiones.
- Elaboración de informes sobre la base de datos.
- Impresión de todas las áreas del paciente.

Los tres primeros puntos previos se refieren a la ampliación de la aplicación de forma que capture imágenes directamente desde el propio paciente a través del sensor o sensores correspondientes tal y como se refleja en el esquema mostrado en la Figura 1: Arquitectura general del sistema. Como se ha indicado en la descripción de esta figura, cabe la posibilidad de que en el futuro se puedan incorporar otro tipo de sensores además de las cámaras mencionadas.

## 15 Glosario

SO: Sistema Operativo.

SGBD: Sistema Gestor de Bases de Datos.

GUI: Graphical User Interface.

Focus: Es el mecanismo que determina cuál de los componentes en una ventana va a recibir eventos de entrada de teclado.

## 16 Bibliografía

- [1] Tigris.org Open Source Software. (2013). *Subclipse*. Obtenido de Tigris.org Open Source Software Engineering Tools: <http://subclipse.tigris.org>
- [2] Tigris.org Open Source Software. (2013). *Subversion*. Obtenido de Tigris.org Open Source Software Engineering Tools: <http://subversion.tigris.org>
- [3] Tödter, K. (2013). *toedter*. Obtenido de <http://www.toedter.com/en/jcalendar/>
- [4] Assembla. (2013). *Assembla Workspaces*. Obtenido de <https://www.assembla.com/workspaces>
- [5] Andrés Gómez de Silva Garza, I. d. (2008). *Introducción a la Computación*. Cengage Learning Editores.
- [6] Eclipse Foundation. (2013). *Eclipse*. Obtenido de <http://www.eclipse.org>
- [7] Fernando Alonso Amo, L. A. (2005). *Introducción a la ingeniería del software*. Delta Publicaciones.
- [8] Harvey M. Deitel, P. J. (2004). *Cómo programar en Java*. Pearson Educación.
- [9] Gamma, E. (2002). *Patrones de diseño: elementos de software orientado a objetos reutilizable 7*. Pearson Educación.
- [10] ISO - International Organization for Standardization. (2013). *ISO Standards*. Obtenido de [http://www.iso.org/iso/home/standards/language\\_codes.htm](http://www.iso.org/iso/home/standards/language_codes.htm)
- [11] iText Software Corporation (California, USA) and 1T3XT BVBA (Ghent, Belgium). (2013). *iTextPDF*. Obtenido de <http://itextpdf.com>
- [12] JBoss Community. (2013). *Hibernate*. Obtenido de <http://www.hibernate.org>
- [13] Oracle. (2013). *Locale*. Obtenido de <http://docs.oracle.com/javase/6/docs/api/java/util/Locale.html>
- [14] Oracle. (2013). *Java Tutorials*. Obtenido de <http://docs.oracle.com/javase/tutorial/uiswing/>
- [15] Oracle. (2013). *ResourceBundle*. Obtenido de <http://docs.oracle.com/javase/6/docs/api/java/util/ResourceBundle.html>

# APÉNDICE

## A1. Acceso a la aplicación

Para acceder a la funcionalidad de la aplicación DermaSystem para la gestión informática en dermatología deberá autenticarse con un usuario válido

En cualquier caso, la aplicación le permitirá acceder al manual de usuario y cambiar el idioma de la aplicación sin necesidad de realizar la autenticación.

Una vez ejecutada la aplicación podrá introducir el usuario y la contraseña válidos, en caso de que no lo sean, la aplicación le mostrará un mensaje de error.

Dependiendo de los permisos del usuario registrado, la aplicación le limitará el acceso a una o varias secciones.

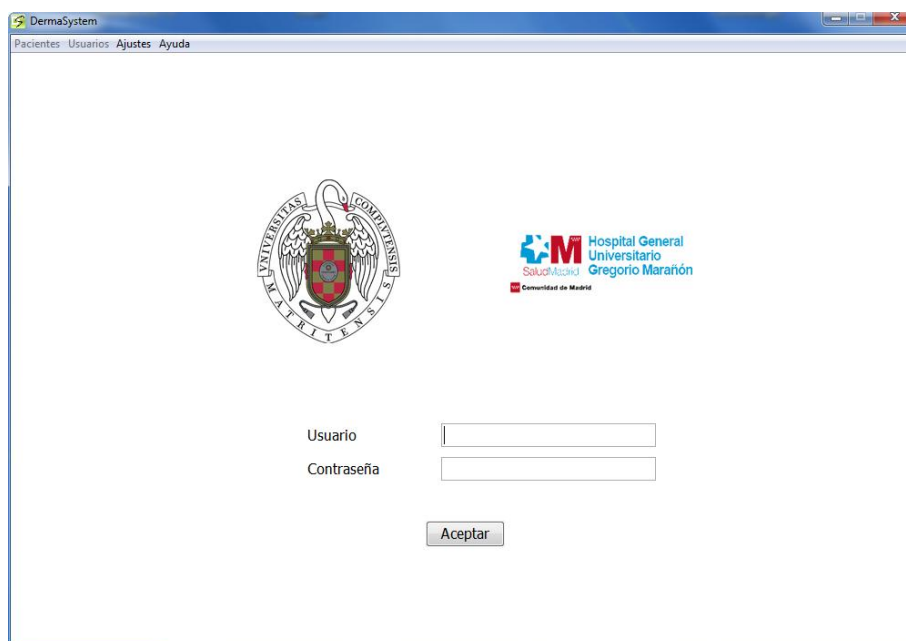


Figura A1: Ventana principal de DermaSystem

## A2. Cambio de idioma

La aplicación le permite cambiar el idioma de uso en cualquier momento, incluso sin necesidad de realizar la autenticación, desde el menú superior **Ajustes -> Idioma**

La aplicación está disponible en el idioma español e inglés. Para un cambio de idioma, será necesario reiniciar la aplicación.

### A3. Información y ayuda

La aplicación le permite acceder al manual de usuario en cualquier momento, incluso sin necesidad de realizar la autenticación, desde el menú superior **Ayuda -> Manual de usuario**

También le mostrará un cuadro de información desde **Ayuda -> Acerca de**.

### A4. Mi cuenta

#### Requerimientos iniciales

- Tener acceso a la aplicación.

Una vez accedido a la aplicación con un usuario válido, indistintamente con permisos de administrador o médico, puede acceder a la información de su usuario, incluso editar su información

Para acceder a su cuenta, desde el menú superior haga clic en **Usuarios -> Mi Cuenta**.

Esta vista le mostrará la información asociada a su usuario registrado.

### A5. Editar mi cuenta

#### Requerimientos iniciales

- Tener acceso a la aplicación.

Desde la vista de **Mi cuenta** en **Usuarios -> Mi Cuenta**, puede editar la información de su usuario haciendo clic en el botón **Editar**, y rellenando los cambios necesarios.

Una vez en el modo edición la operación de editar puede ser cancelada con el botón **Cancelar**

Si los campos de contraseña no se rellenan no se efectuará el cambio de contraseña, por el contrario, si desea cambiar su contraseña deberá rellenar ambos campos.

## A6. Manual de usuario: Gestión de usuarios

### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de administrador

Nota: El acceso al gestor de usuarios con permisos de médico está restringido

### 1) Gestor de usuarios

Si dispone de permisos de administrador, se mostrará el gestor de usuarios.

The screenshot shows the 'Gestor de usuarios' window in the DermaSystem application. The window title is 'DermaSystem' and the menu bar includes 'Pacientes', 'Usuarios', 'Ajustes', and 'Ayuda'. The interface features a search section with a text input for 'Nombre de usuario', 'Buscar' and 'Limpiar' buttons, and a list box containing 'admin' and 'medico'. To the right, there are form fields for 'Nombre de usuario' (admin), 'Correo (\*)' (admin@admin.com), 'Fecha de alta' (2013-04-23), and 'Permisos' (Admin), along with an 'Editar' button.

**Figura A2: Ventana principal de gestión de usuarios**

Esta vista le mostrará todos los usuarios registrados en el sistema, puede acceder por defecto o desde el menú superior **Usuarios -> Gestor de usuarios**.

Haciendo clic en la lista de usuarios registrados le mostrará la información correspondiente al usuario seleccionado.

## 2) Búsqueda de usuarios

### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de administrador

Desde la ventana de **Gestor de Usuarios** en el menú superior **Usuarios -> Gestor de usuarios**, puede filtrar la búsqueda de uno o varios usuarios, escribiendo el nombre en el campo **Nombre de usuario** y haciendo clic en **Buscar**

Puede limpiar todos los campos haciendo clic en **Limpiar**

## 3) Añadir un nuevo usuario

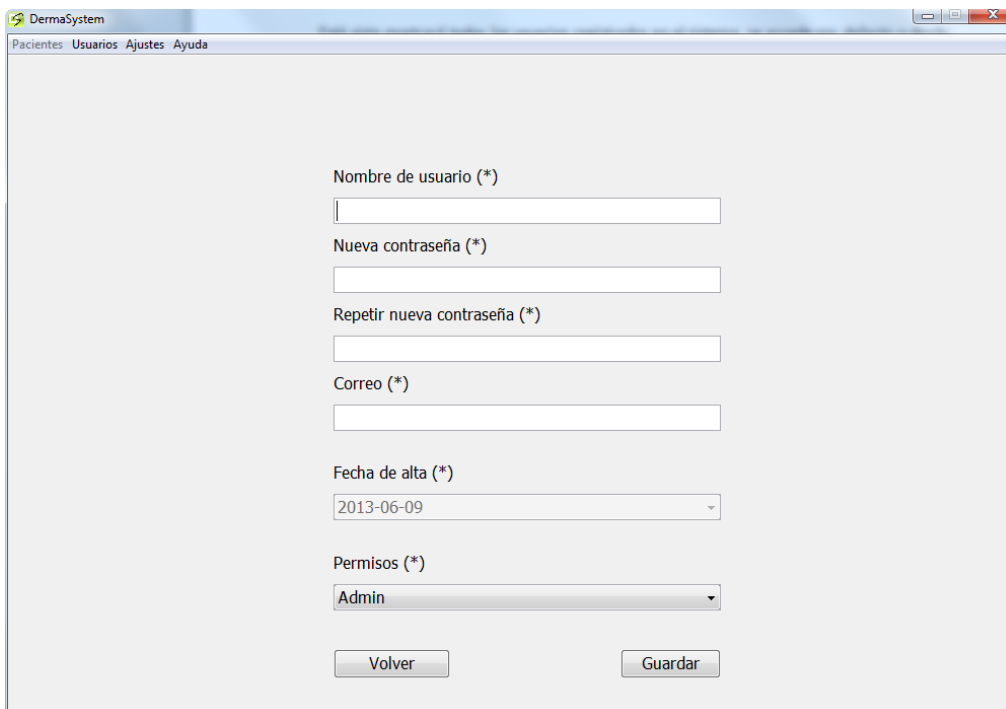
### Requerimientos iniciales

- Tener acceso a la aplicación como permisos de administrador

Puede añadir un nuevo usuario desde el menú superior **Usuarios -> Añadir usuarios**

Se mostrará la vista con los campos necesarios para añadir un nuevo usuario, además del selector que establecerá los permisos del nuevo usuario: Administrador o Médico.

La aplicación le mostrará un mensaje de aceptación si todo ha ido correctamente o un mensaje de alerta si ha ocurrido algún problema.



The screenshot shows a web application window titled 'DermaSystem'. The menu bar contains 'Pacientes', 'Usuarios', 'Ajustes', and 'Ayuda'. The main content area is a form for adding a new user. The form has the following fields and controls:

- Nombre de usuario (\*)**: A text input field.
- Nueva contraseña (\*)**: A text input field.
- Repetir nueva contraseña (\*)**: A text input field.
- Correo (\*)**: A text input field.
- Fecha de alta (\*)**: A date picker showing '2013-06-09'.
- Permisos (\*)**: A dropdown menu with 'Admin' selected.
- Volver**: A button to return to the previous screen.
- Guardar**: A button to save the new user.

Figura A3: Añadir nuevo usuario

## 4) Editar un usuario

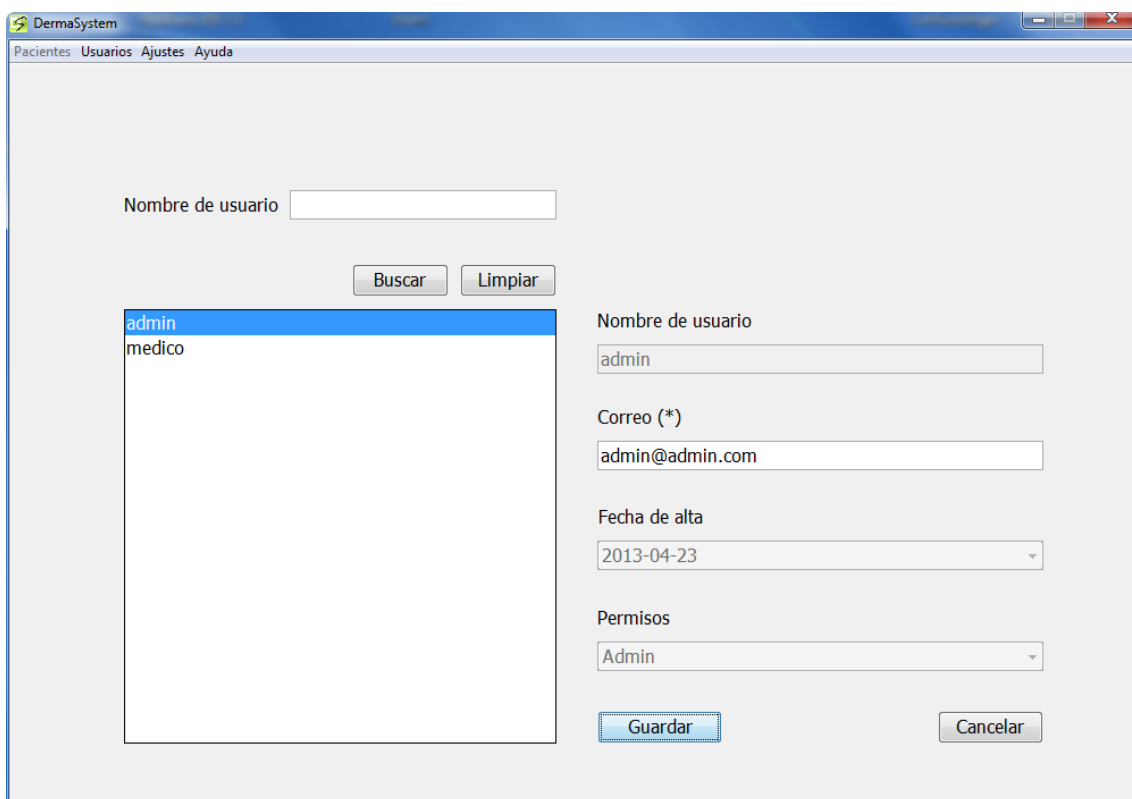
### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de administrador

Desde la ventana de **Gestor de Usuarios** en el menú superior **Usuarios -> Gestor de usuarios**, haciendo clic en un usuario de la lista, se activará el botón **Editar**

Al hacer clic en el botón **Editar**, se activarán los campos para modificar los datos del usuario seleccionado, puede editar y guardar o cancelar la operación.

Nota: Un usuario no podrá modificarse los permisos asimismo.



The screenshot shows a web application window titled 'DermaSystem'. The navigation menu includes 'Pacientes', 'Usuarios', 'Ajustes', and 'Ayuda'. The main content area is divided into two sections. On the left, there is a search section with a text input field labeled 'Nombre de usuario', and two buttons: 'Buscar' and 'Limpiar'. Below this is a list of users: 'admin' (highlighted in blue) and 'medico'. On the right, there is an edit form for the selected user. The form contains the following fields: 'Nombre de usuario' (text input with 'admin'), 'Correo (\*)' (text input with 'admin@admin.com'), 'Fecha de alta' (date picker showing '2013-04-23'), and 'Permisos' (dropdown menu showing 'Admin'). At the bottom of the form are two buttons: 'Guardar' and 'Cancelar'.

Figura A4: Edición de usuario

## 5) Eliminar un usuario

### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de administrador

Desde la ventana de **Gestor de Usuarios** en el menú superior **Usuarios -> Gestor de usuarios**, haciendo clic en un usuario de la lista, se activará el botón **Editar**.

Al hacer clic en el botón **Editar**, se activará el botón **Eliminar**, al hacer clic en este último botón, puede eliminar el usuario tras un mensaje de confirmación.

Nota: Un usuario no podrá modificarse los permisos asimismo, ni siquiera tendrá disponible esta opción ya que el botón Eliminar no se muestra.

DermaSystem

Pacientes Usuarios Ajustes Ayuda

Nombre de usuario

Buscar Limpiar

admin  
medico

Nombre de usuario  
medico

Correo (\*)  
carlos.munoz@gmail.com

Fecha de alta  
2013-06-06

Permisos  
Medic

Guardar Eliminar Cancelar

Figura A5: Eliminar usuario

## A7. Manual de usuario: Gestión de pacientes

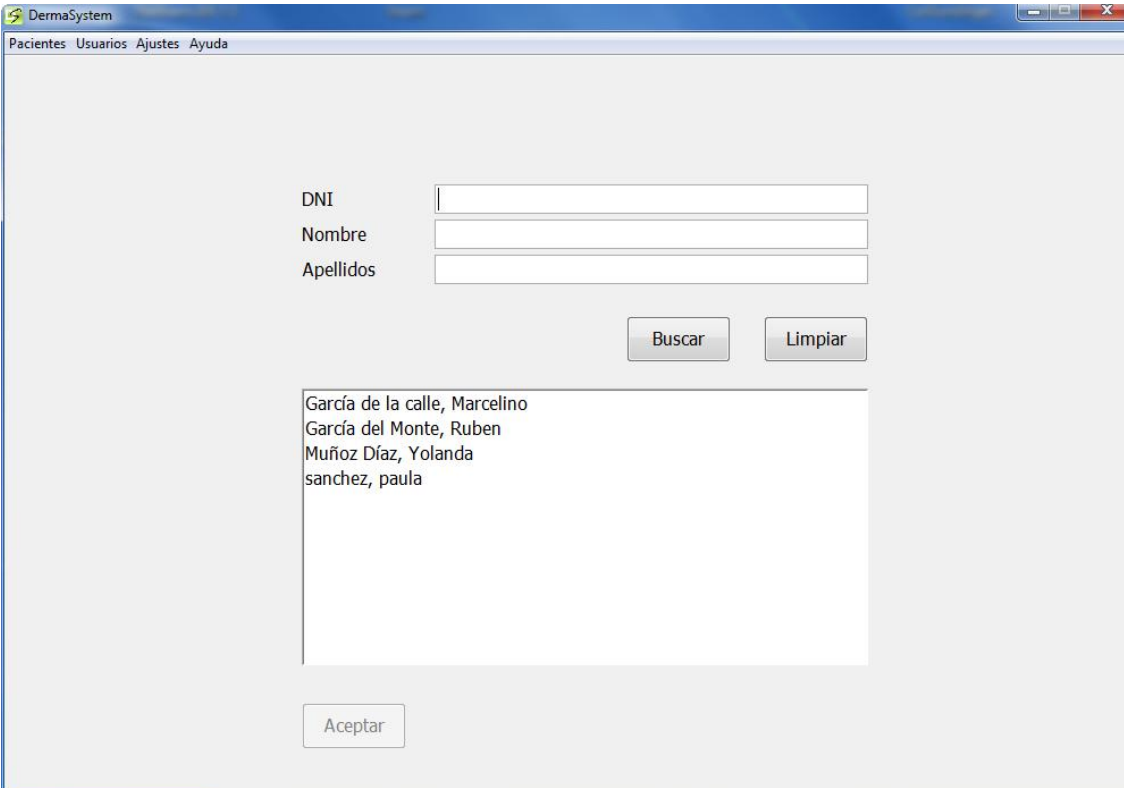
### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de médico

Nota: El acceso al gestor de pacientes con permisos de administrador está restringido

### 1) Gestor de pacientes

Si dispone de permisos de médico, se mostrará el gestor de pacientes.



**Figura A6: Ventana principal de gestión de pacientes**

Esta vista mostrará todos los pacientes registrados en el sistema, se accede por defecto o desde el menú superior **Pacientes -> Gestor de pacientes**.

Haciendo doble clic en la lista de usuarios registrados o el botón **Aceptar** le mostrará la información correspondiente del paciente seleccionado.

La siguiente vista le mostrará la información completa del paciente a la izquierda y el panel de selección de área a la derecha.

Las áreas marcadas en rojo indican que existe una marca en esa zona del cuerpo.

The screenshot shows the 'Vista Principal del Paciente' window in the DermaSystem application. On the left, there is a form with the following fields: DNI (12345678Z), Nombre (Ruben), Apellidos (García del Monte), Fecha de Nacimiento (1980-01-01), Fecha de alta (2013-06-06), Dirección (C/Roberto alarido), Ciudad (Sevilla), Código postal (04009), Teléfono 1 (947589632), and Teléfono 2 (645725369). Below these fields is an 'Editar' button. At the bottom left, there is an 'Observaciones' section with a text area containing 'Gran melanoma en la nariz.' and a 'Volver' button. On the right, the 'Vista Principal del Paciente' section features a red instruction bar: 'Haga click para seleccionar la zona donde desea registrar una marca'. Below this are two human body diagrams: the left one is labeled 'Parte frontal del Paciente' and the right one is labeled 'Parte de atrás del Paciente'. Both diagrams have a red outline and a red circle on the forehead, indicating a selected marking area.

Figura A7: Información del paciente

## 2) Búsqueda de pacientes

### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de médico

Desde la ventana de **Gestor de pacientes** en el menú superior **Pacientes -> Gestor de pacientes**, puede filtrar la búsqueda de uno o varios pacientes, escribiendo el nombre, apellidos o DNI en sus respectivos campos y haciendo clic en **Buscar**.

Los resultados obtenidos serán la unión de cada campo de búsqueda, evitando resultados repetidos.

Se podrá limpiar todos los campos haciendo clic en **Limpiar**.

## 3) Editar un paciente

### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de médico

Desde la ventana de **Gestor de Pacientes** en el menú superior **Pacientes -> Gestor de usuarios**, una vez seleccionado un paciente con doble clic o aceptar, puede acceder a la vista de información.

Al hacer clic en el botón **Editar**, se activarán los campos para modificar los datos del paciente seleccionado, pudiendo guardar o cancelar la operación.

DermaSystem

Pacientes Usuarios Ajustes Ayuda

**Vista Principal del Paciente**

Haga click para seleccionar la zona donde desea registrar una marca

DNI: 00000000T

Nombre: Marcelino

Apellidos: García de la calle

Fecha de Nacimiento: [1964-01-22]

Fecha de alta: 2013-06-06

Dirección: C/Bravo Murillo

Ciudad: Madrid

Código postal: 28092

Teléfono 1: 914528963

Teléfono 2:

Guardar Eliminar

Cancelar

Observaciones

Paciente nervioso.

Volver

Parte frontal del Paciente

Parte de atrás del Paciente

Figura A8: Información del paciente

#### 4) Eliminar un paciente

##### Requerimientos iniciales

- Tener acceso a la aplicación con permisos de médico

Desde la ventana de **Gestor de Pacientes** en el menú superior **Pacientes -> Gestor de usuarios**, una vez seleccionado un paciente con doble clic o aceptar, se accede a la vista de información.

Al hacer clic en el botón **Editar**, se activará el botón de **Eliminar**, al hacer clic en este último botón, se podrá eliminar el paciente tras un mensaje de confirmación.

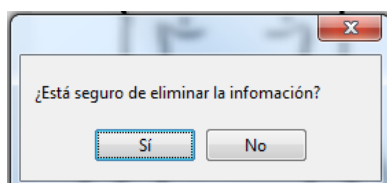


Figura A9: Confirmar eliminar paciente

## A8. Manual de usuario: Marcas

### 1) Seleccionar un área

#### Requerimientos iniciales

- Tener acceso a la aplicación
- Tener un paciente seleccionado

**Paso 1:** Debe hacer **doblo clic** en el área del cuerpo que desee:

*Nota: Las áreas con sombreado rojo indican que el paciente tiene guardas marcas en esa zona.*

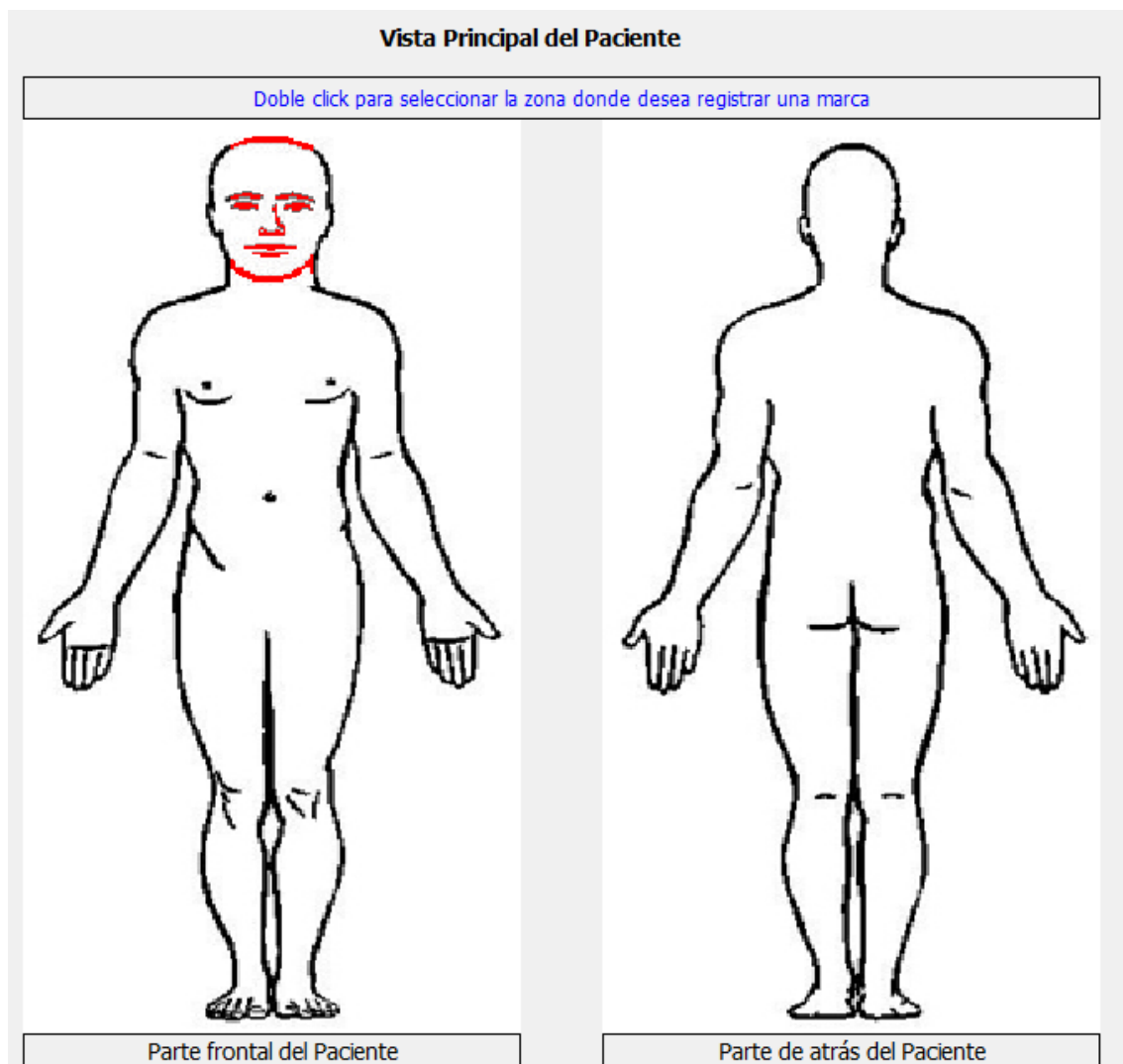


Figura A10: Seleccionar área

**Paso 2:** Tras entrar en una zona le aparecerá la siguiente ventana, Figura A11.

Nota: Preste atención a las indicaciones en azul y en rojo.

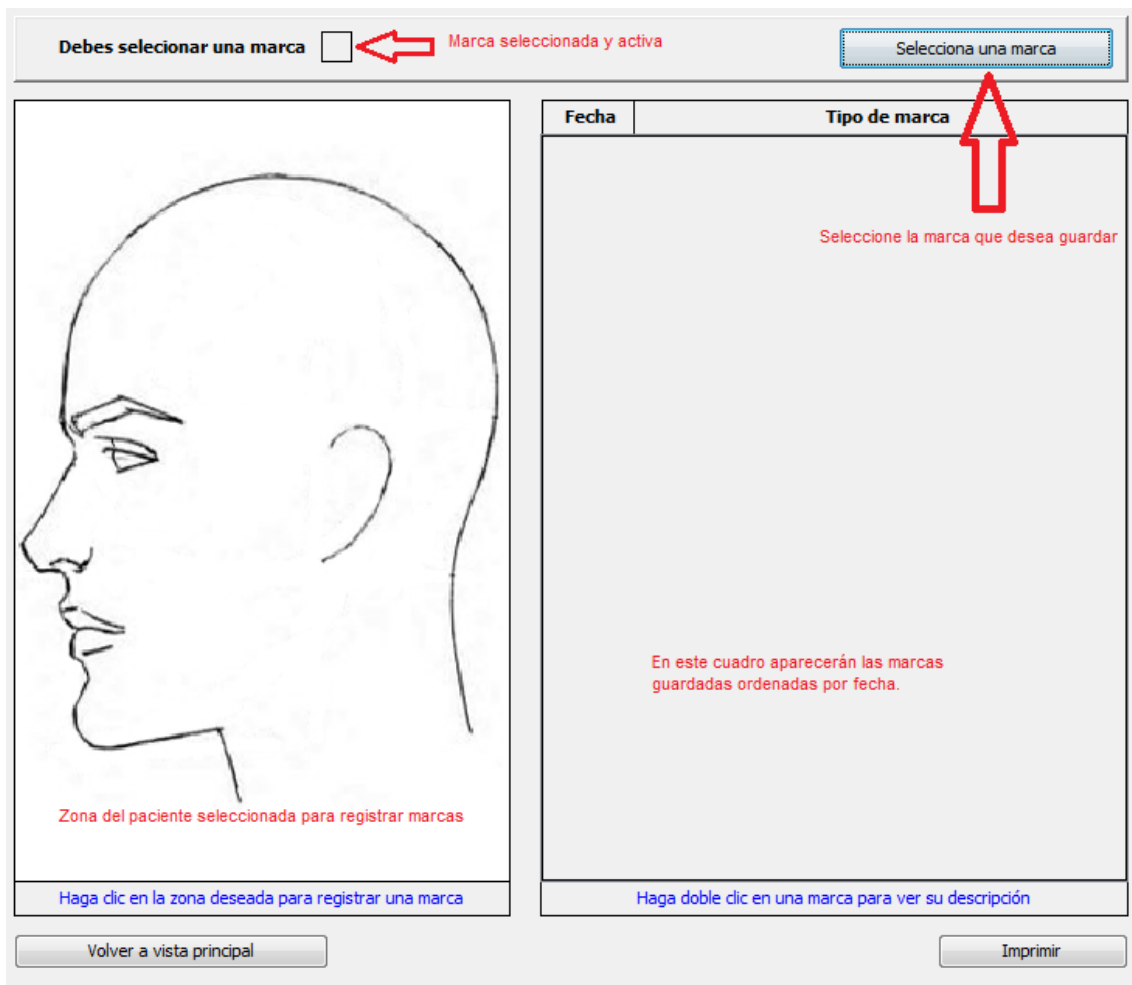


Figura A11: Ventana de marcas detallada

## 2) Registrar una marca

### Requerimientos iniciales

- Tener acceso a la aplicación
- Tener un paciente seleccionado
- Tener un área seleccionada.

**Paso 1:** Seleccionar la marca deseada pulsando en el botón

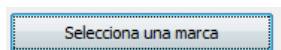


Figura A12: Botón de selección de marca

Tras pulsar en él, aparecerá el siguiente selector de marcas:

**Selecciona una marca**

**Carcinoma basocelular**

**Escisión**

Resección incompleta +

Resección incompleta

Margen lateral afecto +

Margen profundo afecto +

**Tratamiento no quirúrgico** ●

**Carcinoma epidermoide**

**Escisión**

Resección incompleta +

Resección incompleta

Margen lateral afecto +

Margen profundo afecto +

**Tratamiento no quirúrgico** ●

**Queratosis actínica**

**Escisión** +

Tratamiento no quirúrgico ●

**Melanoma**

Escisión +

Tratamiento no quirúrgico ●

Nevus displásico +

Recidiva ✕

Otros □

Aceptar Cancelar

Figura A13: Ventana selector de marcas

**Paso 2: Seleccionar la marca que desea.** Para ello marque con el ratón la deseada en su respectivo “checkbox”  y pulsar el botón Aceptar para confirmar.

Si desea cancelar la operación pulse el botón Cancelar.

*Nota: El apartado “Marca seleccionada” siempre le dirá la marca seleccionada actualmente.*

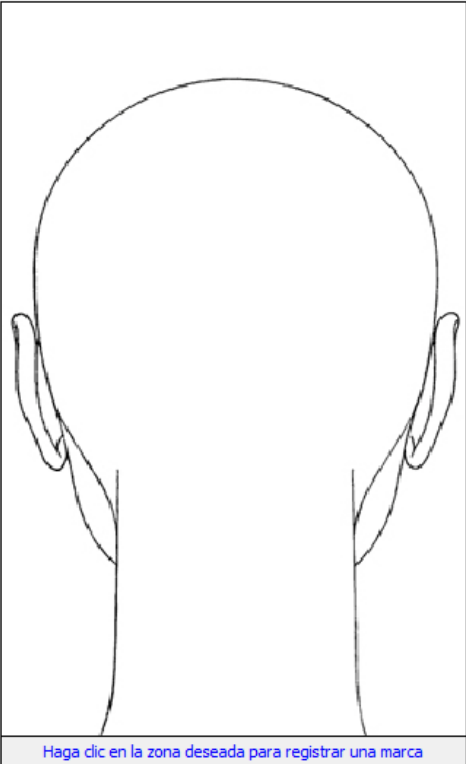
Marca seleccionada ->  Carcinoma basocelular: Resección incompleta

Figura A14: Indicador de marca seleccionada

Tras seleccionar su marca volverá a la ventana anterior:

Marca seleccionada -> + Carcinoma basocelular: Resección incompleta Selecciona una marca

Fecha	Tipo de marca



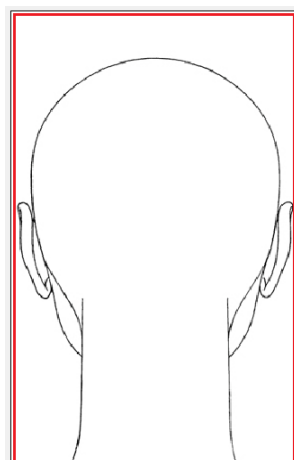
Haga clic en la zona deseada para registrar una marca

Haga doble clic en una marca para ver su descripción

Volver a vista principal
Imprimir

**Figura A15: Ventana de marcas**

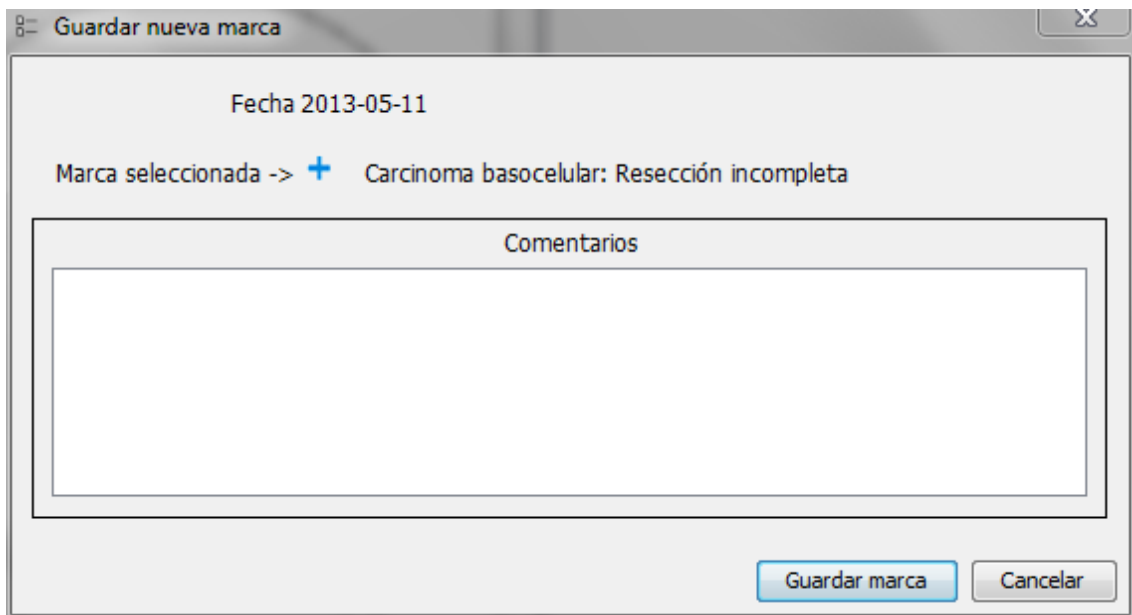
**Paso 3: Registrar la marca en la zona deseada.** Para ello sólo tiene que hacer clic en la zona que desee dentro de los límites (marcados por la línea roja representada en la Figura A7).



**Figura A16: Zona de inserción de marcas**

Tras hacer clic le aparecerá un cuadro de dialogo como el mostrado en la Figura A17.

*Nota: En comentarios puede guardar un comentario en relación a la marca que va a guardar.*



**Figura A17: Detalle de marca**

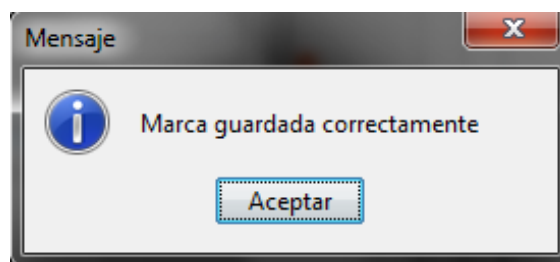
Para guardar esta marca pulse en el botón

Guardar marca

Para cancelar esta operación pulse en

Cancelar

Tras pulsar en el botón “guardar marca” le aparecerá un cuadro de dialogo indicándole que se han guardado los cambios de forma correcta (Figura A18). Pulse el botón Aceptar para cerrar la ventana.



**Figura A18: Confirmación de guardado de marca**

El resultado final será el mostrado en la Figura A19.

*Nota: Preste atención a las indicaciones en rojo*



Figura A19: Resultado de registro de marca

### 3) Borrar marcas

#### Requerimientos iniciales

- Tener acceso a la aplicación.
- Tener un paciente seleccionado.
- Tener un área seleccionada.
- Dicho área debe tener marcas registradas.
- 

**Paso 1:** Seleccionar la marca a editar haciendo un solo clic en la lista de marcas registradas de la derecha:

Fecha	Tipo de marca
2013-05-11	Carcinoma basocelular: Resección incompleta
2013-05-11	Queratosis actínica: Tratamiento no quirúrgico
2013-05-11	Carcinoma basocelular: Margen profundo afecto
2013-05-11	Melanoma: Tratamiento no quirúrgico
2013-05-11	Carcinoma epidermoide: Tratamiento no quirúrgico

Figura A20: Historial de marcas

Al pulsar una sola vez la marca en el área de la derecha aparecerá con un círculo rojo indicándole que la marca seleccionada es esa en la representación:

Fecha	Tipo de marca
2013-05-11	Carcinoma basocelular: Resección incompleta
2013-05-11	Queratosis actínica: Tratamiento no quirúrgico
2013-05-11	Carcinoma basocelular: Margen profundo afecto
2013-05-11	Melanoma: Tratamiento no quirúrgico
2013-05-11	Carcinoma epidermoide: Tratamiento no quirúrgico

Figura A21: Selección de marca en el historial

**Paso 2:** Una vez localizada su marca, hacer **double clic** sobre ella en la **lista**. De esta forma puede ver los comentarios que se guardaron al registrar la marca.

Aparecerá un cuadro de dialogo como el de la Figura A22:

Detalle de la marca

Fecha 2013-05-11

Tipo de marca Carcinoma basocelular: Resección incompleta

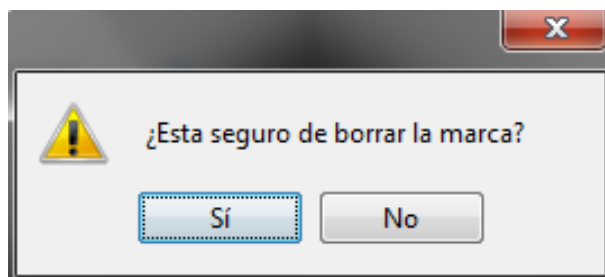
Imagen de marca: + Comentarios

Borrar marca Salir

Figura A22: Detalle de marca

**Paso 3:** Pulsar en el  botón. Si desea cancelar el proceso de eliminación pulsar en el botón  para cerrar esta ventana.

Tras esto le aparecerá un cuadro de confirmación como el de la Figura A23.



**Figura A23: Confirmación eliminación de marca**

Si confirma la operación de borrado la marca desaparecerá completamente del paciente, no obstante seguirá guardada en base de datos.

#### 4) Imprimir el área de un paciente

##### Requerimientos iniciales

- Tener acceso a la aplicación.
- Tener un paciente seleccionado.
- Tener un área seleccionada.
- Dicho área debe tener marcas registradas.
- Tener instalado un visor de archivos PDF.

**Paso 1:** Pulsar en el botón . Tras esto se abrirá un archivo de tipo PDF donde se encontraran los datos que desea imprimir.

Aparecen los datos del paciente, sus marcas en el área y una lista con dichas marcas como se muestra en la Figura A24 (página siguiente).

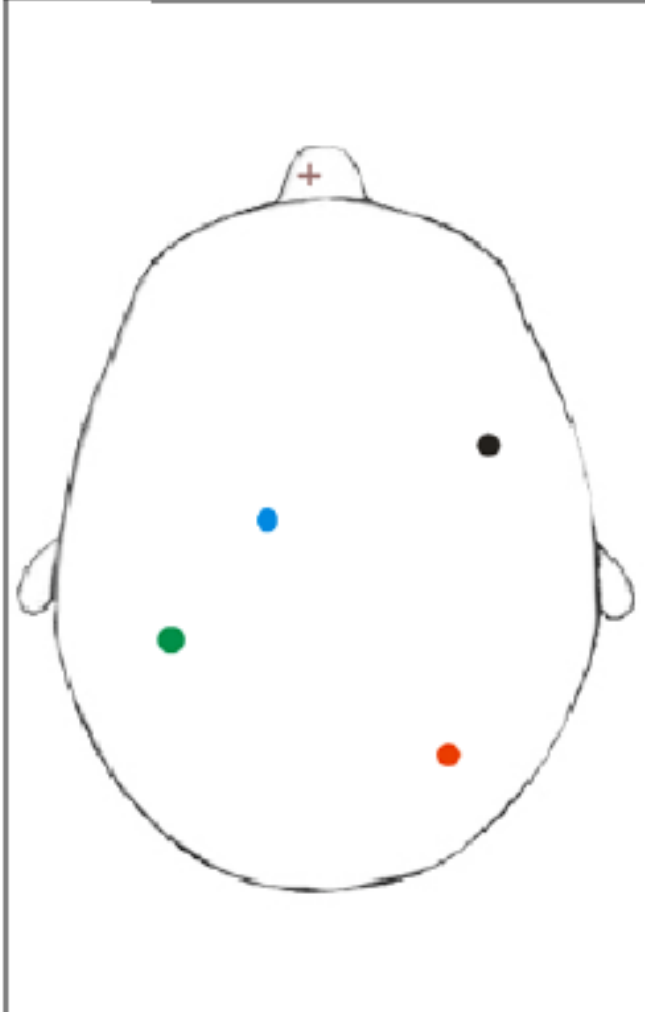
Guarde el documento donde desee (Menú Archivo, Guardar Como...).

70456123Q.pdf - Adobe Reader

Archivo Edición Ver Ventana Ayuda

1 / 1 50% Herramientas

Información del paciente:  
Nombre: Pedro  
Apellidos: Dominguez Rodriguez  
DNI: 70456123Q



Información de marcas

- 0: Nevus displásico
- 1: Carcinoma basocelular: Tratamiento no quirúrgico
- 2: Carcinoma epidermoide: Tratamiento no quirúrgico
- 3: Queratosis actínica: Tratamiento no quirúrgico
- 4: Melanoma: Tratamiento no quirúrgico

Figura A24: Impresión de marcas

