



**Facultad de Matemáticas y informática
Universidad de Barcelona**

GRADO DE INGENIERÍA INFORMÁTICA

**APLICACIÓN PARA EVALUAR EL GRADO
DE CAQUEXIA EN ENFERMOS DE
CÁNCER PARTE II: VISUALIZACIÓN Y
GESTIÓN DE DATOS**

Albert Guzmán Espadas

Director: Eloi Puertas Prats
Trabajo final de grado
Realizado en: Departamento de
Matemáticas y informática. UB
Curso 2016-2017

INDICE

1. INTRODUCCION	5
1.1 Definiciones básicas	6
1.2Objetivos	9
2. PLANIFICACION	10
3. ANALISIS	13
3.1 Usuarios	13
3.2 Descripción de los casos de uso más importantes	16
4. DISEÑO	26
4.1 Cliente / Servidor	26
4.2 Tecnologías	27
4.3 Modelo vista controlador	28
4.4 Base de datos	30
4.5 Librerías externas	32
5. IMPLEMENTACION	34
5.1 Actualizaciones	34
5.2 Nuevas implementaciones	36
6. CONCLUSIONES	50
7. BIBLIOGRAFIA	53
8. ANEXOS	54

RESUMEN

Este trabajo de final de grado es la continuación de otro proyecto que tiene como título: Aplicación para evaluar el grado de caquexia en enfermos de cáncer realizado por Javier Gómez Sánchez.

El proyecto se apoya en un grupo de investigación de la Facultad de Biología de Barcelona. La aplicación ya dispone de alojamiento en un servidor de la Facultad y actualmente ya está en funcionamiento.

La finalidad de seguir con este proyecto es profundizar en la herramienta y dotarla de nuevas funcionalidades. Agrandando y robusteciendo la aplicación.

El proyecto se va a centrar en la visualización de datos con la creación de gráficas para facilitar a los usuarios hacer un seguimiento de sus pacientes. De esta manera les resultará más sencillo ver la evolución de los pacientes.

Se agrupará los distintos usuarios en roles. Conseguiremos separar los permisos de cada uno de ellos y tener más control sobre sus secciones y sus funcionalidades.

Se implementarán métodos para tener un mayor control sobre los usuarios. Estos métodos se encargarán de borrar, modificar y visualizar usuarios. Por otro lado, los usuarios dispondrán de un sistema de recuperación de cuenta.

Para que la aplicación sea más segura durante el desarrollo de la herramienta se migrara a https para tener los datos seguros cuando viajen por internet.

ABSTRACT

This end-of-grade paper is the continuation of another project that has as its title: Application to assess the degree of cachexia in cancer patients by Javier Gómez Sánchez.

The project is supported by a research group of the Faculty of Biology of Barcelona. The application already has accommodation on a server of the Faculty and is currently in operation.

The purpose of continuing with this project is to deepen the tool and provide it with new functionalities. Enhancing and strengthening the application.

The project will focus on visualizing data with the creation of graphs to make it easier for users to keep track of their patients. In this way it will be easier to see the evolution of the patients.

The different users will be grouped into roles. We will be able to separate the permissions of each one of them and to have more control over its sections and its functionalities.

Methods will be implemented to have greater control over users. These methods will delete, modify and view users. On the other hand, users will have an account recovery system.

To make the application more secure during tool development it will be migrated to https to have the data safe when traveling over the internet.

1. INTRODUCCION

Casco es una aplicación web con el objetivo de hacer un seguimiento y evaluación de los pacientes que padecen algún tipo de cáncer y poder evaluar y clasificar su grado de caquexia.

Esta herramienta forma parte del proyecto de investigación de la Facultad de Biología de la Universidad de Barcelona. El proyecto está dirigido por Silvia Busquets que nos ha facilitado el acceso a esta aplicación y nos ha guiado según las necesidades de la herramienta y del proyecto.

Este proyecto es la segunda parte de un primer TFG llevado por Javier Gómez Sánchez. La finalidad de hacer una segunda parte es mejorar y añadir nuevos elementos y nuevas funcionalidades.

La aplicación está en fase de preproducción. Con ello queremos decir que está en desarrollo y queremos obtener una versión final que será el producto. Está destinado para uso real en medicina.

La herramienta está siendo usada en distintos países como Brasil e Italia, entre otros, a modo de prueba para demostrar que puede llegar a ser una herramienta potente y útil. Muchos de los datos que se muestran son de pacientes reales.

Antes de iniciar la explicación del TFG, es necesario explicar previamente unos conceptos que se usan a lo largo del proceso. Algunos de ellos son triviales, pero para evitar futuras confusiones se explican algunas ideas a continuación.

1.2. Definiciones básicas

1. Caquexia (Definición extraída de la Tesis Doctoral Marta Villarino Sanz):
“La palabra caquexia procede del griego y su significado es Kakos (malo) y hexis (estado). Existen distintos tipos de caquexia, pero nos centraremos en la cancerosa que es a la que va dirigida esta aplicación. Este tipo de caquexia consiste en los procesos metabólicos que provocan en la paciente anorexia, pérdida de peso y pérdida de la masa muscular.

Las principales consecuencias a nivel metabólico son el aumento de la gluconeogénesis hepática, del recambio de glucosa, disminución de la captación de la misma a nivel muscular, hiperlipidemia, aumento de la lipólisis, del recambio proteico, de la síntesis proteica hepática, de la degradación muscular, resistencia a la insulina, reducción de la secreción y de las hormonas reguladoras como son el cortisol o las catecolaminas y su posterior liberación junto con factores inflamatorios de respuesta rápida.

Además, se activa la respuesta linfocitaria ante un estado inflamatorio y, a nivel clínico, el paciente experimenta una serie de cambios que afectan no solo a su estado actual sino a su situación frente a la enfermedad. Existe una pérdida de peso importante, debiéndose principalmente a esa depleción del músculo, afectando a la masa muscular esquelética y cardíaca. Por otro lado, el paciente presenta o puede presentar: náuseas, vómitos, astenia, fatiga, malestar general y una disminución de su ingesta.”

2. Paciente:

Es la persona que padece la caquexia. Los datos extraídos se obtendrán a partir de los síntomas de este.

3. Doctor:

Usuario principal de la herramienta. Es el encargado de evaluar los síntomas del paciente correctamente e introducirlos en el formulario de la aplicación para obtener su índice de severidad de caquexia.

4. Formularios:

Con la investigación y desarrollo de Silvia y su equipo, han establecido una serie de componentes y cálculos para obtener una puntuación final que hace referencia al nivel de severidad de la caquexia. Esta puntuación se dividirá en 'severa', 'moderada', 'inicial' o 'sin caquexia' según unos valores de corte preestablecidos por el equipo.

El formulario consta de una serie de preguntas que el doctor deberá rellenar con la finalidad de obtener dicha puntuación. Las preguntas del formulario están desglosadas en cinco apartados distintos: BWC, IMD, PHP, ANO y QOL. Cada apartado generará una puntuación distinta y a su vez, cada uno de ellos tiene un peso distinto al valor final de CASCO.

Cada paciente puede tener muchos formularios realizados con distintas fechas. De esta manera es posible realizar un seguimiento de la evolución del paciente.

5. BWC ("Body weight composition") 5 preguntas:

Componente que hace referencia a la pérdida de peso del paciente. Está dividida en:

- Porcentaje de peso total que ha perdido.
- Porcentaje de masa magra perdida.

Tiene un peso del 40% de la puntuación que se obtiene de CASCO. Es la componente importante a la hora del diagnóstico.

6. IMD ("Inflammation metabolic") 4 preguntas:

- Determina la inflamación a partir de la proteína C reactiva i la interleuquina-6.
- Determina alteraciones metabólicas.
- Estima la respuesta inmune del paciente a partir del valor absoluto de linfocitos.

Representa un 20% en la puntuación final.

7. PHP (“Actividad física”) 5 preguntas:

Hace referencia a la actividad física que realiza el paciente.
Tiene un peso del 15% en la puntuación final.

8. ANO (“Anorexia”) 4 preguntas:

Identificar el grado de anorexia a gran escala que puede presentar el paciente. También tiene un peso del 15%.

9. QOL (“Quality of life”) 25 preguntas:

Mide la calidad de vida a corto plazo para identificar anomalías.
Normalmente se rellenan los datos de esta sección en relación a la calidad de vida de la semana anterior. Tiene un peso del 10%.

10. MiniCASCO:

A parte del formulario anterior es posible realizar un formulario a menor escala obteniendo una puntuación orientativa. Este formulario mantiene los requisitos del formulario original, pero se lleva a cabo con menos preguntas. Del mismo modo lo tiene que rellenar el doctor sobre un paciente y también queda registrado en la aplicación para su evaluación y clasificación.

11. Casc-IN:

Es otro formulario en menor escala que el MiniCASCO. Se obtiene una puntuación con menor exactitud. La gran diferencia es que no requiere estar identificado en la aplicación para poder hacer uso de él y no queda guardado en la aplicación. Es una “demo” para poder mostrar el funcionamiento básico.

1.3. Objetivos

Queremos que esta herramienta vaya evolucionando y para ello es necesario hacer modificaciones y añadir funcionalidades. La primera versión de CASCO estaba más enfocada a la implementación del formulario, la obtención de los datos del paciente y al cálculo de la severidad de la caquexia. En este proyecto nos centraremos en los siguientes puntos:

- Mejorar funcionalidades existentes.
- Determinar que necesita la aplicación para ser más robusta.
 - Detectar posibles errores.
 - Añadir distintos roles a los usuarios con la finalidad de aumentar la seguridad en la aplicación CASCO.
 - Verificar el diseño de las páginas y modificar si fuese necesario.
- Gestión de datos.
 - Dar mayor control de los datos a los doctores
- Visualizar datos de los formularios
 - Generación de nuevas gráficas.
 - Pensar nuevas formas de visualización de datos que puedan ser útiles a los doctores para agilizar la evaluación y seguimiento de los pacientes.
- Añadir tablas en la base de datos para poder procesar cambios mencionados en los puntos anteriores.

2. PLANIFICACION

El proyecto se ha desarrollado durante el primer cuatrimestre del curso. La implementación de la herramienta se divide tres partes:

1. Familiarización con la herramienta:
 - a. Entender todo el código.
 - b. Familiarizarse con los distintos lenguajes de programación.
 - c. Instalaciones.

2. Investigación:
 - a. Investigar las distintas posibilidades de realizar que la herramienta sea más robusta.
 - b. Mejoras nuevas necesarias.
 - c. Mejorar funcionalidades ya existentes.

3. Implementación:
 - a. Realización de los métodos nuevos.
 - b. Implementación de mejoras.
 - c. Testear todas funcionalidades.
 - d. Volver al paso dos y repetir el proceso.

En la figura 1 que se muestra a continuación se observa los días aplicados a cada tarea de forma global para poder visualizarlo de manera gráfica.

Las horas que se han dedicado al proyecto están distribuidas según el tiempo que se ha podido aprovechar por temas de trabajo y asignaturas. Las horas expuestas son para tener una orientación de la dedicación.

- Octubre-noviembre-diciembre
 - Lunes a viernes: entre una y tres horas diarias.
 - Sábados y domingos: entre siete y nueve horas.

- Enero
 - Entre siete y doce horas diarias

Plantilla de diagrama de Gantt



Figura 1: Diagrama de Gantt.

En el diagrama de Gantt que se muestra en la figura 1 se ha desglosado las tareas más importantes. Cada barra indica la fecha de inicio y fecha final de cada tarea para visualizar la evolución del proyecto. Se han aprovechado los meses de octubre, noviembre y diciembre para la implementación del proyecto y el mes de enero para finalizar las últimas modificaciones y memoria.

3. ANALISIS

En la fase anterior del proyecto se profundizó más en crear la aplicación y en poder registrar los formularios en una base de datos. En esta fase se ha centrado en la gestión y visualización de datos.

Previamente se hizo una investigación de las funcionalidades que se necesitaba como la distinción de usuario y gráficas. A lo largo del proyecto ha habido implementaciones que no se habían pensado en un principio pero que eran de vital importancia y que no se había contemplado.

Para ello es necesario un previo análisis de cómo estructurar el proyecto, tecnologías, actualización en la herramienta, modificaciones en la base de datos etc.

Esta herramienta se ha implementado siguiendo una serie de patrones para obtener un código de calidad, estructurado y escalable. También se ha investigado las tecnologías que cumplían con los requerimientos de la aplicación para poder ser usadas.

3.1 Usuarios

Para poder utilizar la aplicación CASCO es necesario tener identificación y contraseña para acceder. Se ha pensado en distribuir los usuarios que van a utilizar la aplicación por agrupación de roles para poder dar una capa de seguridad a la aplicación.

Al tener agrupaciones por roles también se consigue distribuir sus funcionalidades y acceso en función del rol del usuario hacia la aplicación. Con ello podemos separar las funcionalidades que desempeñan cada uno en unos diagramas de caso de uso.

Los distintos roles que desempeñan los usuarios en CASCO y sus funcionalidades más destacadas son:

- Administrador:
 - Este perfil es el máximo responsable en la aplicación.
 - Su finalidad es asegurarse el correcto funcionamiento que tiene la aplicación y llevar un seguimiento de los usuarios que estén identificados en ella.
 - Para que un usuario pueda utilizar la aplicación debe ser registrado mediante el administrador. El administrador puede registrar un usuario indicando el rol que éste desempeñará en la aplicación y se le facilitará un nombre de identificación y una contraseña.
 - Puede visualizar todos los usuarios de la aplicación sea cual sea su rol en esta.
 - Es capaz de ver todos los formularios y pacientes.
 - Tiene la posibilidad de modificar los datos personales de cualquier usuario y también su contraseña por si dado el caso fuera necesario.
 - Puede borrar cualquier doctor. Esto implica borrar su perfil por lo que no podrá volver a utilizar la aplicación. Si se quisiera volver a utilizar, debería contactar con el administrador y volver a generar un nuevo usuario ya que no se mantiene ningún dato una vez borrado para no tener datos innecesarios.
 - También tiene la posibilidad de borrar pacientes. Por la misma razón que la anterior, si un paciente ya no va a tener relación con su doctor y este no volverá a realizar formularios de dicho paciente se tiene la posibilidad de borrarlo para mantener la base de datos lo más limpia posible.

- Tiene a su disposición las funcionalidades para generar y descargar un Excel o CSV de formularios que hayan realizado el doctor a su elección.

- Doctor:
 - El perfil encargado de tener relación directa con el paciente enfermo de cáncer y realizar el formulario para obtener su índice de afectación de la caquexia.

 - Debe tener un nombre de identificación y contraseña facilitado por el administrador.

 - Puede generar distintos formularios por paciente para llevar un seguimiento de este y poder ver su evolución.

 - Tiene a su disposición una serie de gráficas para poder tener una visualización de datos de sus pacientes con la finalidad de facilitar una visión global de estos.

 - Puede modificar sus datos personales, pero no su contraseña. Se le impide modificarla para que el administrador tenga un mayor control de este.
En el caso de que no recuerde su contraseña, la aplicación dispone de una recuperación de contraseña que se explicara en el apartado de mejoras en la sección de implementación.

 - Puede ver el historial de un paciente. Esto recupera sus formularios y datos para tener un seguimiento de manera rápida en forma de tabla.

- Researcher:
 - Este perfil se ha generado para tener una agrupación de usuarios con la finalidad de visualizar datos. Al igual que el doctor, debe tener un nombre de identificación y contraseña facilitado por el administrador. No tienen a su disposición ninguna funcionalidad que sea registrar o modificar datos en la aplicación. Se ha pensado este rol para cualquier tipo de persona que no vaya desempeñar el mismo papel que el doctor ni el administrador. Aunque no desempeñe ninguna funcionalidad aparte de visualizar datos, este perfil es muy útil para que CASCO disponga de una agrupación de usuarios que puedan utilizar la aplicación y sean capaces de ver datos de pacientes reales para investigaciones, estadísticas, etc.
 - Tiene a su disposición gráficas para ver datos de todos los pacientes que hay registrados en CASCO.
 - Pueden visualizar todos los formularios que se han realizado en la aplicación, así como su fecha, puntuación, doctor, paciente etc.

3.2 Descripción de los casos de uso más importantes.

Seguidamente explicaremos los **nuevos** casos de uso y los **modificados** mediante tablas que facilitan su descripción.

Los casos de uso de los tres roles **compartidos** son los siguientes:

- UC-01 : LogIn*.
 - Es la funcionalidad para que los usuarios puedan identificarse en la aplicación para hacer uso de ella. Es necesario que el usuario disponga de un nombre de identificación y contraseña que se la ha debido de facilitar un administrador. El login se encuentra en la página inicial de la herramienta.

En login es el mismo para todos los usuarios de la aplicación pero cada usuario accederá a su apartado correspondiente con sus funcionalidades respectiva.

- UC-02 : LogOut*.
 - Funcionalidad para salir de la aplicación y borrar cualquier rastro del usuario en la aplicación.
En la versión previa de la aplicación existían algunos problemas con esta funcionalidad ya que no borraba todo rastro y era posible volver a entrar al refrescar la página sin escribir el nombre de identificación ni contraseña.

- UC-03: Recuperación password.
 - Es un requisito que cualquier aplicación web que sea utilizada por usuarios debe tener. Da la posibilidad de modificar la contraseña en caso de ser olvidada.

En la anterior versión no se podía realizar ningún cambio en los datos de los usuarios. Por esa razón se ha creído conveniente crear esta funcionalidad para que cada usuario pueda obtener una nueva contraseña al momento.

El único requisito es saber el nombre de identificación del usuario, su 'lastname' y 'country'. Estos datos se guardan al crear el usuario. Por lo que si es un usuario de casco deberá recordar estos valores.

En el caso de no recordar deberá contactar con el administrador.

Los casos de uso del rol **Administrador** son los siguientes:

- UC-04 : Registrar Doctor*.
 - Métodos para registrar un nuevo usuario en la aplicación. Debe de rellenar un formulario con los datos del doctor, así como la contraseña y nombre de identificación. Si el nombre de identificación ya existe se le informa para que lo modifique. Una vez rellenado correctamente se pulsa en 'Register' y se registra en la aplicación.

- UC-05 : Modificar Doctor.
 - Funcionalidad para modificar los datos de usuarios. El administrador puede modificar cualquier dato si es necesario. Inicialmente se ha pensado esta funcionalidad para la modificación de la contraseña. Se obliga al administrador a estar disponible para poder cambiarlo. Esto es una desventaja porque si el administrador no está disponible para realizar la modificación el usuario que requiera de su ayuda no podrá identificarse en casco. Por esta razón se creó el UC-04. De todas maneras, la funcionalidad es útil en el caso de que el usuario no pueda modificarla o en otras situaciones concretas.

- UC-06 : Borrar Doctor.
 - Esta implementación sirve para borrar doctores que no vayan a utilizar más la herramienta. Así se pueden borrar de los registros y no podrá acceder de nuevo. Si quisiera volver a usarla deberá contactar con el administrador para que le genere una nueva identificación.

- UC-07 : Transferir pacientes.
 - Su uso es esencialmente visualizar todos los pacientes y sus formularios. Se muestra una tabla con los formulario, doctor, paciente, fecha y índice de severidad de la caquexia de dicho formulario.

En la versión anterior también se podía ver los formularios de los pacientes, pero se requería saber el nombre de un paciente. En este caso muestra todos para una mejor visualización global. Es mucho más dinámico esta visualización por si el usuario quiere buscar formulario, pero no recuerda el paciente y si el doctor, o solo recuerda la fecha, o recuerda la fecha y el índice, etc.

- UC-08 : Visualizar usuarios.
 - Esta funcionalidad muestra una tabla con todos los usuarios identificados en la aplicación. No tiene ninguna funcionalidad sobre ellos. Simplemente es para que el administrador tenga una visualización rápida de todos los usuarios registrados en ella.

- UC-09: Exportar CSV
 - Esta funcionalidad es parecida al UC-16. Obtiene datos de los formularios de un paciente y lo exporta a un fichero CSV. Se creó conveniente su implementación porque UC-16 exporta un Excel y dependiendo el sistema operativo y el programa con el que es abierto, mostraba columnas desplazadas.

En cambio, los ficheros de tipo CSV es un fichero sin formato en el que los datos están separados por un único carácter. Al abrirlo con algún programa a elección del usuario, este separa los elementos a partir de ese carácter. Se ha testeado con los programas que daban conflictos con el Excel y funciona correctamente.

Los casos de uso del rol **Doctor** son los siguientes:

- UC-10 : Registrar Paciente*.
 - Funcionalidad para registrar un paciente en la aplicación. Esto es necesario si se quiere rellenar un formulario de dicho paciente.

- UC-11 : Modificar Doctor.
 - Esto permite a los doctores modificar sus datos personales registrado en la base de datos de la aplicación CASCO.

- UC-12 : Dashboard pie.
 - Visualiza una gráfica de tipo 'pie'. Clasifica todos los pacientes del doctor según su índice de severidad en : severe, mild, moderate y none.

- UC-13 : Dashboard tree.
 - Es una gráfica de tipo árbol en la que se filtra por los pacientes del doctor y se separa entre sexos. Podemos clicar en la sección que se desee y esta nos mostrará los elementos que tiene en su interior mostrando los datos de cada uno.

- UC-14 : Dashboard bubble.
 - Grafica para visualizar los formularios de los pacientes del doctor durante todo un año. Cada formulario se representa con un círculo. El tamaño del círculo depende del índice de caquexia y cada color de dicho círculo representa un paciente distinto.

Con este tipo de grafica podemos hacer un seguimiento rápido de las fechas en las que se han realizado los formularios y saber rápidamente si el paciente necesita otra visita o simplemente recordar las que ha realizado.

- UC-15 : Dashboard compare.
 - La finalidad de esta funcionalidad es ver la evolución de un paciente. La grafica muestra todos sus formularios ordenados por fecha. Para cada formulario muestra en forma de barras los resultados de los componentes QOL, BWC, PHP, ANO y IMC. También muestra el índice.

Los casos de uso comunes entre **Administrador** y **Doctor** son los siguientes:

- UC-16: Exportar Excel*.
- - Funcionalidad para exportar formularios realizados. En el caso del administrador puede elegir el doctor para exportar sus formularios. En el caso del doctor, exportara sus formularios.

Los casos de uso del rol **Researcher** son los siguientes:

Los siguientes casos de uso son los mismos diseños que los UC-15, UC-14, UC-13 y UC-12. La diferencia es que los datos mostrados en ellas no están filtrados por doctor, en cambio muestras todos los datos independientemente del doctor que haya generado los datos a visualizar. Otra diferencia es la manera de acceder a ellos.

- UC-17: Dashboard pie.
- UC-18: Dashboard tree.
- UC-19: Dashboard bubble.
- UC-20: Dashboard compare.

(Los casos de uso del apartado anterior son parecidos a los del doctor. La diferencia es que se accede de manera distinta y los datos que se muestran son distintos. Por esas razones se deben crear nuevos casos de uso para esas acciones)

Los casos de uso comunes entre **Administrador** y **Researcher** son los siguientes:

- UC-21 : Visualización global de formularios

* Caso de uso existentes pero modificados.

La figura 2 es la leyenda de los diagramas de caso de uso separado por roles para identificar las acciones que pueden realizar cada usuario en función del grupo. Para diferenciar los casos de uso de la versión anterior con los nuevos o modificados se utilizan tres colores:

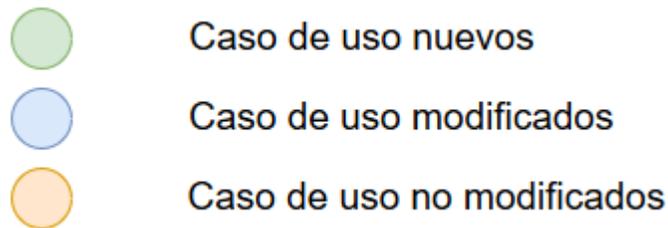


Figura 2: Leyenda casos de uso de usuarios.

1. Caso de uso nuevos: hacen referencia a las nuevas implementaciones que se necesitaban en la aplicación CASCO.
2. Caso de uso modificados: son funcionalidades existentes que se han modificado con pequeños matices para mejorar su funcionalidad o rendimiento.
3. Caso de uso no modificados: aquellas implementaciones que no se han modificado.

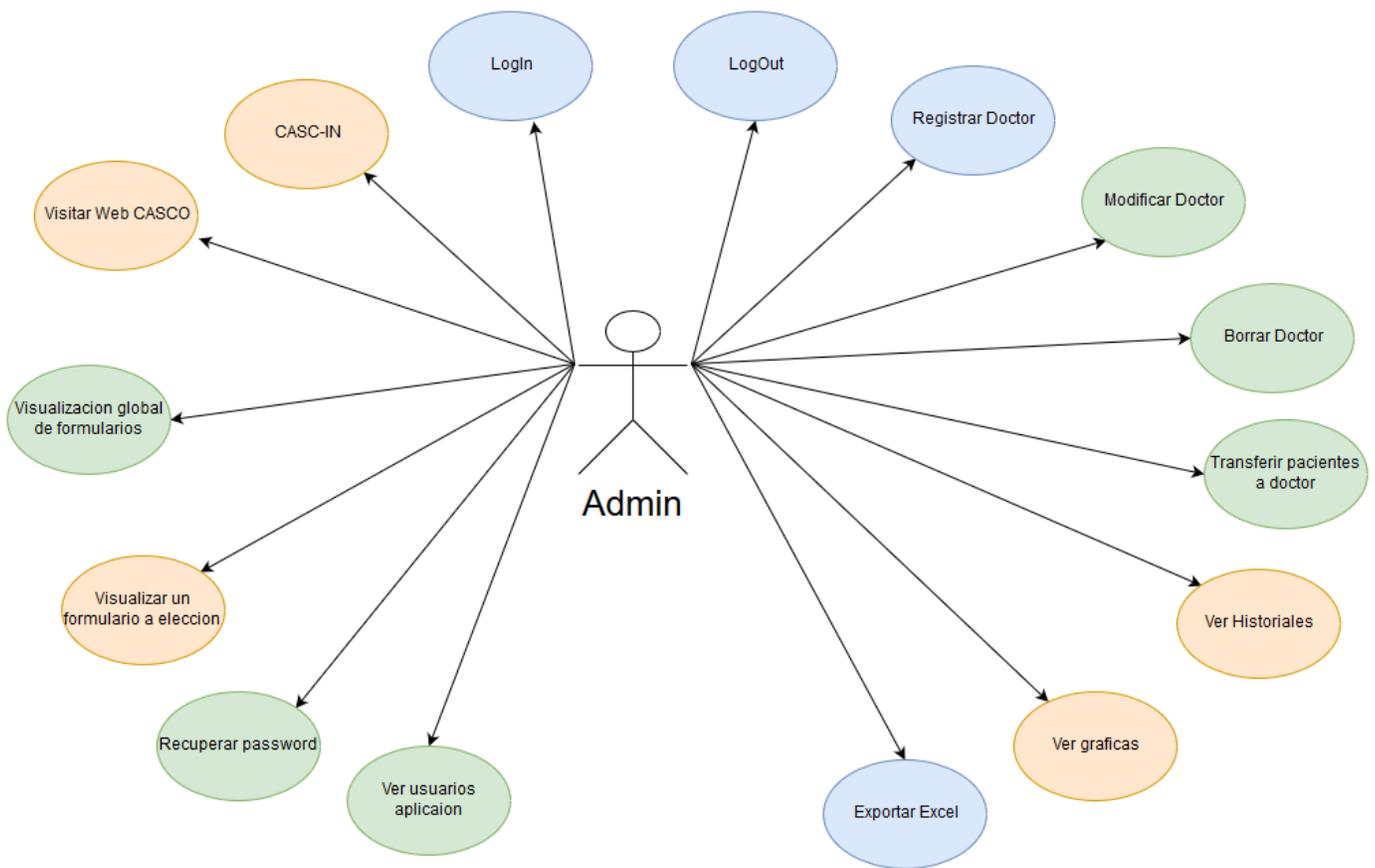


Figura 3. Casos de uso usuario administrador

En la figura 3 se observa un diagrama de caso de uso del administrador. Se diferencia sus funcionalidades con los distintos colores mencionados en la leyenda de la figura 2.

En la figura 3, el actor con el nombre Admin, representa los usuarios administradores de la aplicación. Las flechas indican la capacidad que tiene este actor en realizar la funcionalidad a la que se apunta. Finalmente, cada círculo muestra la acción que dicho actor tiene la posibilidad de realizar. En el diagrama se muestran los casos de uso más importantes de dicho actor con su respectivo color.

Existen funcionalidades y casos de uso internamente a cada funcionalidad que os indicamos en la figura 2. Simplemente se da una idea global de las funcionalidades más importantes

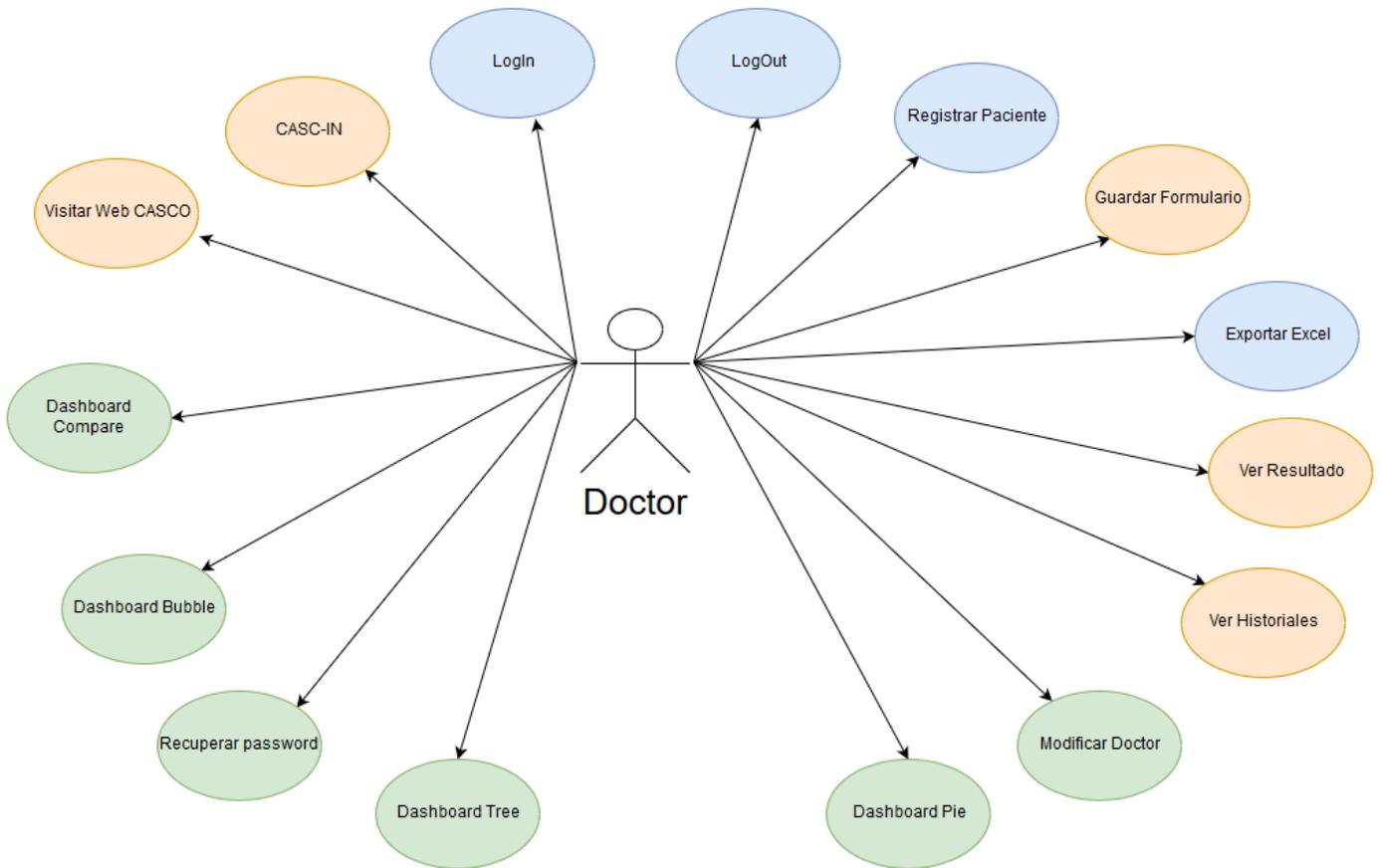


Figura 4. Casos de uso usuario Doctor

En la figura 4, el actor representa la agrupación de usuarios del rol Doctor. Las flechas la capacidad del actor en realizar las acciones. Y por último, cada círculo las funcionalidades que puede realizar el actor.

Algunas funcionalidades son las mismas, pero se generan a partir de distintos datos o de accesos diferentes.

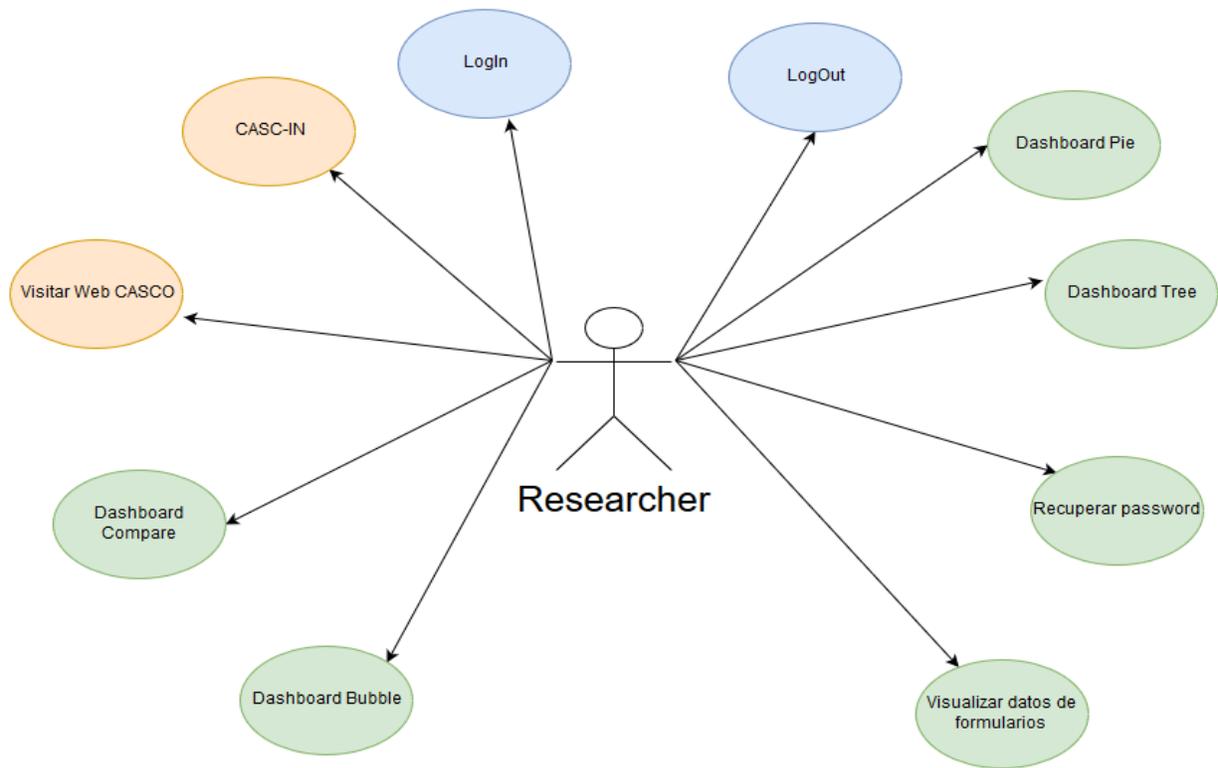


Figura 5. Casos de uso researcher

La figura 5 es el diagrama de casos de los usuarios con el rol re ‘researcher’.

Como en la figura 2 y 3 , el actor representa la agrupación de usuarios de ese rol. Las flechas la capacidad del actor en realizar las acciones y los círculos las funcionalidades que puede realizar.

Se puede diferenciar claramente con los otros actores, que tiene menos funcionalidad. Tal y como esta explicado en apartados anteriores, la finalidad de esta agrupación de usuarios es visualizar datos.

Las distintas graficas que visualizan los doctores y los ‘researchers’ tienen el mismo nombre porque la gráfica tiene el mismo diseño. Los datos que se visualizan en cada una de ellas son distintos. Se explica con más detalle en la próxima sección.

4. DISEÑO

En esta sección se explica el diseño de la aplicación, las tecnologías que se utilizan en ella, las librerías que se han importado para dar soporte a la aplicación, el sistema cliente servidor, la estructura del proyecto y la base de datos utilizada.

4.1 Cliente/Servidor

Esta aplicación web está diseñado e implementado en un sistema cliente/servidor. Es un sistema distribuido donde la idea principal es que el cliente genere una serie de peticiones y el servidor de soporte a estas peticiones. El cliente y servidor pueden ejecutarse en la misma maquina o en distintas. Generalmente el servidor es capaz soportar muchas peticiones y de distintos clientes a la vez. El servidor es el motor, parte lógica de la aplicación.

De esta manera se facilita que el usuario no tenga que tener toda la gestión y lógica de la aplicación instalada en su máquina local ni los datos de las BBDD que pueden llegar a ser muy elevados. Este modelo cliente/servidor permite la distribución de información de manera ágil, eficaz y a distancia.

Los usuarios utilizarán la aplicación mediante un navegador web cualquiera. Este navegador será el responsable de invocar recursos de la aplicación casco generando peticiones.

Proporciona al usuario final un acceso transparente a sus peticiones, gestiones, datos o cualquier recurso solicitado.

Las acciones más comunes por parte del cliente son:

- Ejecutar la lógica necesaria de manera local (Ej.: formatear datos)
- Interactuar con el usuario.
- Visualizar la interfaz gráfica.
- Recibir respuesta del servidor.
- Generar peticiones y lanzarlas al servidor

Por parte del servidor, las funciones más comunes son:

- Recoger las peticiones del cliente.
- Actuar con la base de datos de la aplicación.
- Procesar la lógica necesaria para ejecutar las peticiones.
- Devolver resultados.

4.2 Tecnologías

En la implementación de la aplicación web CASCO se utiliza el lenguaje de programación PHP. Es uno de los más utilizados para el desarrollo de aplicaciones web. Es un lenguaje simple y versátil con infinitas posibilidades.

Para la creación de las vistas (páginas web) se utiliza HTML5 + CSS. Generan las vistas que el usuario final visualiza en su máquina local. Permite ver los menús, tablas, títulos etc.

Adicionalmente, para dar más soporte y dinamismo en las funcionalidades, hacemos uso de:

1. JavaScript: Lenguaje de programación para crear dinamismo en la página web. Principalmente se ejecuta en la parte del cliente. (Ej.: Despliegues de menús).
 - a. Para facilitar la implementación en JavaScript se utiliza jQuery. Es una librería para agilizar la implementación en JavaScript y poder generar código más eficiente, más eficaz y más limpio.
2. Ajax (*Asynchronous JavaScript and XML*): El principal objetivo de Ajax es el transporte de datos entre cliente y servidor sin necesidad de refrescar la página web entera. Proporciona más control de aquello que queremos enviar o recargar.

4.3 Modelo vista controlador

En la aplicación se utiliza el diseño de software MVC (Modelo vista controlador). Generalmente se usa para aplicaciones con interfaces de usuario y sirve para potenciar la separación de conceptos, reutilización de código, mejor mantenimiento y código estructurado. En general, para una mejor calidad de código.

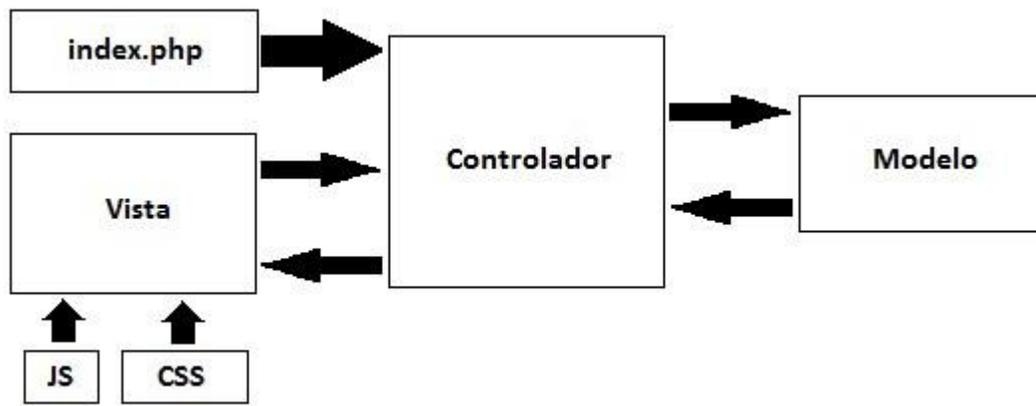


Figura 6. MVC

En la figura 6 se muestra un esquema de este diseño. Se basa en separar el código en tres capas llamadas modelo, vista y controlador. A continuación, explicaremos brevemente en que consiste cada una de ellas.

- **Modelo:** Principalmente tiene la lógica necesaria para trabajar con los datos de la aplicación. Generalmente los datos estarán almacenados en base de datos por lo que es en el modelo donde estarán las conexiones necesarias para acceder a ella y los métodos para trabajar con ella.
- **Vista:** Como su nombre indica, contiene todo el código necesario para renderizar las interfaces de usuario, es decir, los estados de nuestra aplicación en HTML. En las vistas, el usuario tiene la percepción de trabajar con los datos, pero no accede a ellos directamente. Para eso está la capa de modelo anterior mencionada que accederá a los datos y los mostrará a partir de esta capa.

- Controlador: Contiene el código necesario para dar respuesta a las llamadas de la vista (peticiones que se solicitan del cliente). Es una capa de enlace entre el modelo y la vista. Su finalidad no es manipular ni realizar ningún tipo de lógica, si no que su función principal es identificar la petición y accionar los métodos para realizar dicha acción.

A continuación, se muestra la estructura de casco según el diseño MVC.

Para identificar entre las nuevas aportaciones, las modificadas y las no modificadas utilizamos lo siguiente:

- Vistas nuevas
- Vistas modificadas
- Vistas sin modificar

La estructura de CASCO siguiendo el diseño MVC es:

:

- Vistas: 32 vistas que usamos como interfaz entre usuario y aplicación.

admin.php	Cascin.php	create_csv.php	create_excel.php
dashboardBubble.php	dashboardComp.php	dasboardCompSelect.php	dashboardPie.php
dashboardPieResearch.php	dashboardTree.php	delete_paciente.php	doctores.php
doctorescsv.php	export.php	export_admin.php	formulario.php
grafica.php	historial.php	historial_admin.php	login.php
pacientes.php	recoverpass.php	recoverpassOK.php	recoverpassKO.php
tableall.php	tabledoctors.php	tablepatients.php	updateself.php
researcher.php	usuariosdelete.php	usuariosupdate.php	Index.php

- Controlador: Un controlador por donde se procesa cualquier petición que genere el cliente y requiera de accionar los mecanismos correspondientes para llevar a cabo la petición.

Controlador.php

- Modelo: dispone de 4 modelos distintos para manipular datos y devolver el resultado de las peticiones del controlador para que lo pueda mostrar en las vistas.

Usuario.php
Historial.php
Formulario.php
Dashboard.php

4.4 Base de datos

Para el desarrollo y gestión de la base de datos se utiliza el sistema relacional MySQL. Es openSource y está basado en un lenguaje de consulta estructurado (SQL).

Diagrama entidad relación de CASCO (ER)

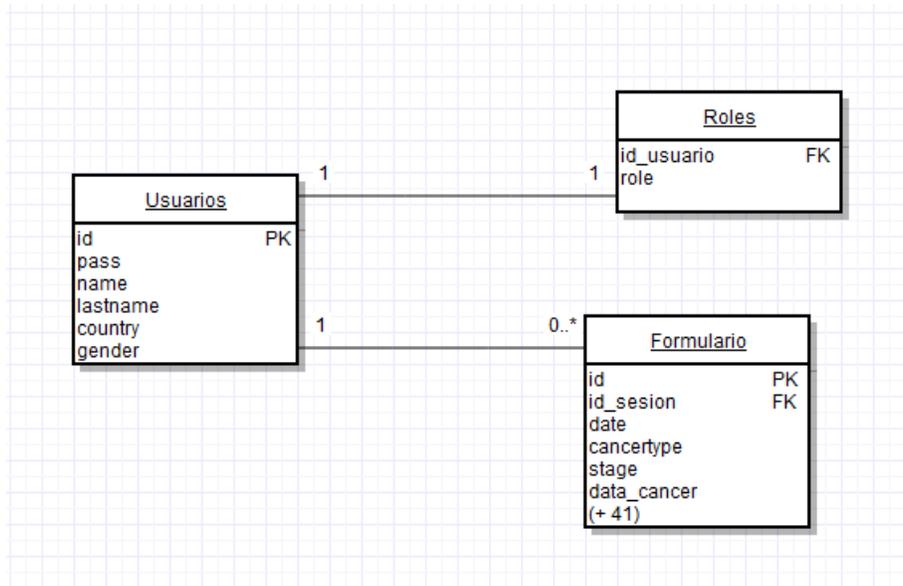


Figura 7. ER Base de datos

El sistema de almacenamiento de la aplicación consta de tres tablas relacionales tal y como se muestra en la figura 7.

Las tablas de usuario y formulario ya estaban en funcionamiento en la aplicación casco. La tabla de formularios consta de 47 columnas para guardar todas las respuestas. Se pensó en modificarla y separarla por apartados según las preguntas y datos del formulario para tener la base de datos mejor estructurada pero finalmente para agilizar el proyecto se decidió no modificarlas.

- Tabla usuarios: en esta tabla se guarda los datos personales de todos los usuarios de la aplicación. Su atributo 'id' es la primary key (PK) que utilizamos como valor único para ese registro. A su vez, este componente nos sirve de clave secundaria o foreign key para identificar registros de otra tabla con el usuario correspondiente.

- Tabla Roles: guardamos el rol de los distintos usuarios.
- Tabla Formulario: se almacena todas las respuestas realizadas del formulario, así como su puntuación por componente, puntuación global, tipo de cáncer etc.

4.5 Librerías externas

Adicionalmente a los lenguajes de programación nombrados, se utilizan librerías openSource como soporte a la implementación en ciertos puntos de la aplicación.

Las dos primeras librerías que comentaremos a continuación no se ha utilizado en este TFG pero si son necesarias para algunas funcionalidades.

1. Charts.js:

Da soporte para generar graficas mediante componentes canvas.

Es una biblioteca opensource, basada en JavaScript y html5. Es una de las primeras opciones de multitud de personas a la hora de realizar gráficas sencillas. Tiene poca variedad de gráficas, entre ellas se encuentran las barras, líneas, áreas, etc. pero su potencia radica en que es muy sencilla de utilizar.

2. Jspdf:

Permite generar PDF desde JavaScript. Generalmente se crean los archivos desde el servidor pero con esta biblioteca permite que el mismo cliente genere los pdf. Conseguimos que el servidor no tenga que crearlos y así lo liberamos de algunos procesos. Esta librería esta utilizada en la primera parte del proyecto. Se implementó la funcionalidad de exportar PDF pero la idea se desechó por problemas que causaba entre los distintos sistemas operativos y sus compatibilidades.

3. Google charts:

Permite la generación de graficas basado en HTML5 y CSS3.

No es necesario aumentar el tamaño de nuestro proyecto porque esta librería se carga directamente de los servidores de google.

Es de uso gratuito, pero al ser una versión libre tiene sus limitaciones. La desventaja más significativa es que limita las peticiones a 250.000 usos al día. En nuestra aplicación no llegamos a tantas peticiones de los usuarios por lo que no tendremos ningún problema.

Se ha elegido esta librería porque facilita a gran escala la implementación de gráficas. Dispone de muchos tipos, así como mezclar distintas graficas en una sola. Además, es 'responsive' por lo facilita una interacción entre el usuario y la gráfica automáticamente.

Dispone de muchas opciones que podemos aplicar a las visualizaciones, así como recargas de datos simple, animaciones, eventos, etc. Pero estas últimas ya no son tan triviales de utilizar.

5. IMPLEMENTACION

El apartado de la implementación se ha distribuido en varias secciones:

1. Actualizaciones: se explica modificaciones en funcionalidades que ya existían en la versión anterior de la herramienta.
2. Nuevas implementaciones: indica las nuevas aportaciones a la aplicación, así como su descripción.

5.1. Actualizaciones

En este apartado se indica las actualizaciones en la aplicación, pero no son funcionalidades nuevas ni acciones adicionales a la aplicación de casco que requieran de diagramas ni casos de uso. Son pequeños cambios en el código para corregir, agilizar o facilitar acciones al usuario.

5.1.1. Selects

En la versión anterior de la aplicación, en algunos apartados en concreto, el usuario debe escoger opciones a partir de un componente llamado 'select' que se muestra en la figura 8.

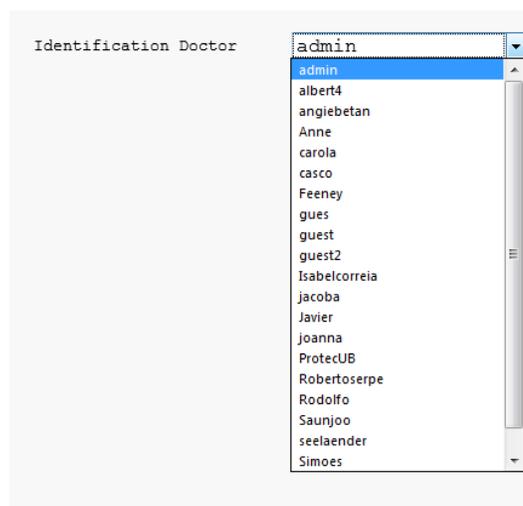


Figura 8: select

La ventaja de este tipo de entrada es que fuerza al usuario que seleccione un valor de la lista. Cada elemento de la lista es codificado por el programador por lo que al obtenerlo con jquery te aseguras que no haya errores de escritura.

El problema que se nos presenta es cuando esta lista de selección es demasiado grande y el tedioso buscar la opción entre cientos e incluso miles.

Se ha modificado los selects más importantes por inputs editables. Este componente tiene las propiedades del select anterior mencionado y también da la posibilidad al usuario de escribir en el input. Al escribir en él, muestra una lista con los elementos filtrados por el valor escrito por el usuario.

Podemos observar en la figura 9 como el usuario ha escrito los caracteres 'al' y se le visualizan los valores filtrados por esa cadena escrita en vez de mostrar todas las opciones

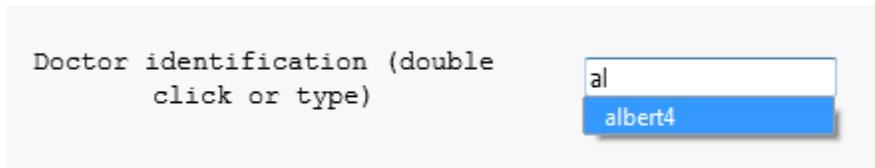


Figura 9: input con funciones del select y editable

5.1.2. Exports Excel

Se ha modificado ligeramente el export del Excel. En un punto en concreto se producía una división en la que, si los valores de la operación eran mayores a cero, se efectuaba el cálculo sin problemas. Pero existen casos en los que un componente podía ser cero o nulo.

En este caso la operación no se puede realizar y la variable que contiene ese nulo es exportada al Excel. Dependiendo el programa utilizado, es capaz de interpretar un nulo como una celda vacía, otros colocar algún símbolo extraño, otros colocan un cero. Se ha modificado para que ese valor nulo, sea cero.

También existía el problema de que según el sistema operativo y los programas que se utiliza para abrir este tipo de ficheros utilizan distintos signos a la hora de identificar números con decimales. Algunos utilizan puntos y otros comas. A la hora de abrir un fichero puede dar problemas o simplemente estropea el formato del documento. Se ha modificado para formatear toda la salida de números.

5.1.3. Input fecha

En registrar un nuevo formulario se debe de informar del año. Para indicarlo se abre un calendario donde se puede seleccionar la fecha y rellena el campo automáticamente con el año. Este método generalmente funciona correctamente, pero a veces hace que la aplicación se bloquee por temas de compatibilidad entre versiones.

Se ha decidido eliminar el calendario e indicar el formato de fecha que debe escribir el doctor para registrar el nuevo formulario.

Otra solución es rellenar automáticamente el año obteniendo la fecha del sistema, pero se deja de la manera indicada para que el doctor tenga el control.

5.2. Nuevas implementaciones

En esta sección mostraremos algunas de las nuevas implementaciones que visualiza cada usuario de la aplicación dependiendo del rol que desempeña en la herramienta.

Todas las implementaciones están descritas en la sección de casos de uso. En este apartado solo se pretende mostrar algunas de las funcionalidades.

5.2.1 Menús de cada rol



Figura 10: Admin

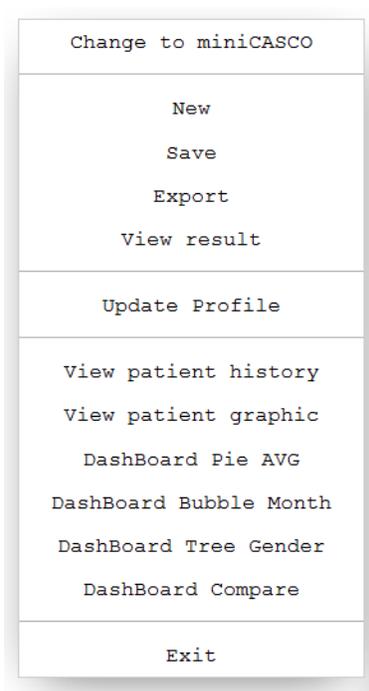


Figura 11: Doctor



Figura 12: Researcher

En la figura 10 se muestra el menú que dispone el administrador. En la figura 11 el menú de doctor y por último el menú del researcher en la figura 12.

En la siguiente tabla mostramos las secciones de los menús de las figuras 10, 11 y 12 según se hayan implementado de cero, modificado o sin modificar. Para diferenciarlo se utilizan los colores:

-  Sección nueva
-  Sección modificada
-  Sección sin modificar

Administrador	Doctor	Researcher
Register user	Change to miniCasco	Table patients
Delete user	New	Dashboard Pie AVG
Update user	Save	Dashboard Bubble month
Table patients	Export	Dashboard tree gender
Table users	View Result	Dashboard compare
Export to Excel	Update Profile	Exit
Export to CSV	View patients history	
View patients history	View patients graphic	
View patients graphic	Dashboard Pie AVG	
Delete patient	Dashboard Bubble month	
Exit	Dashboard tree gender	
	Dashboard compare	
	Exit	

5.2.2. Doctor: Dashboard pie

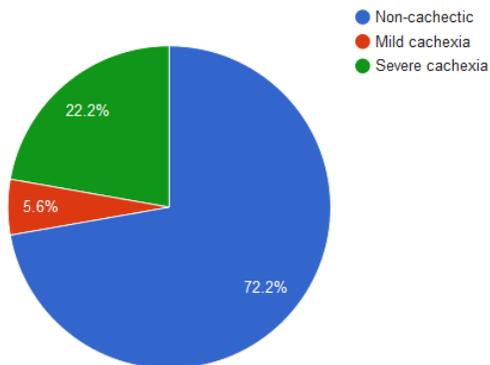


Figura 13: Dashboard Pie.

La figura 13 es una gráfica que tienen disponible todos los doctores y recopila información de todos sus pacientes y muestra el porcentaje de cuantos hay entre:

- No-cahetic: el índice es muy bajo y no presenta indicios de afectación.
- Mild-cachexia: paciente presenta los síntomas iniciales
- Moderate-cachexia: el paciente tiene síntomas de estar afectado por su enfermedad en un grado moderado.
- Severe-cachexia: tiene un índice muy alto.

5.2.3. Researcher y doctor: Dashboard Bubble

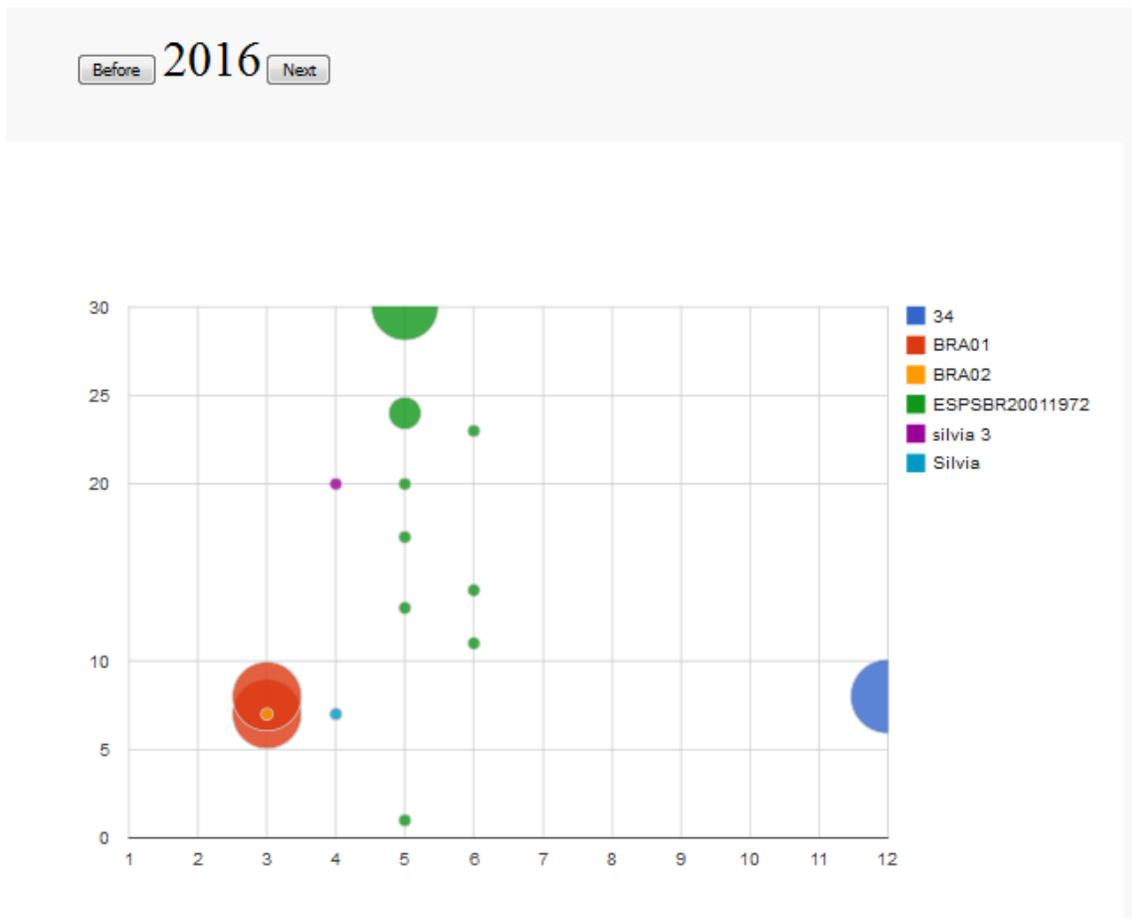


Figura 14: Dashboard Bubble.

En la figura 14 se muestra los formularios de todos sus pacientes por fecha durante todo un año. Cada círculo representa un formulario realizado y cada color es un paciente distinto. Los tamaños de los círculos indican el índice de caquexia que presenta ese formulario. Se han incorporado unos botones para navegar por las gráficas y consultar distintos años. La gráfica está distribuida en los ejes 'x' los meses del año y en los ejes 'y' los días.

5.2.4. Researcher y doctor: Dashboard Compare

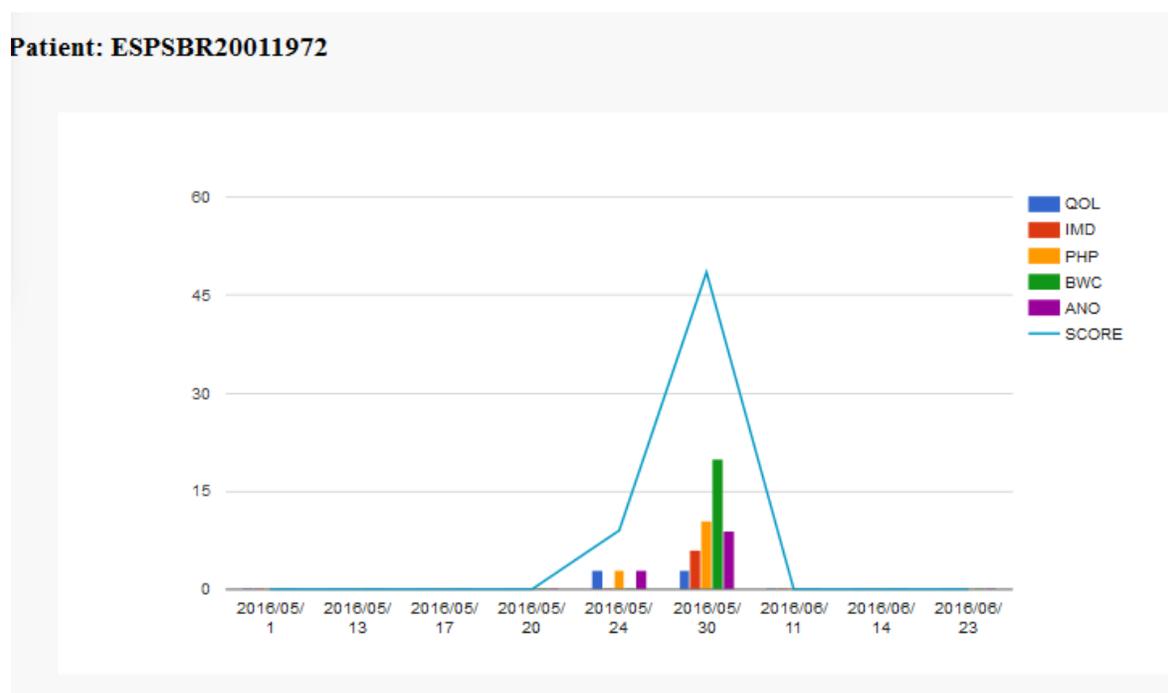


Figura 15: Dashboard Compare.

En la figura 15 se muestra la imagen de la sección 'Dashboard Compare' de los doctores. Es necesario que el doctor previamente escoja un paciente para poder generar la gráfica.

Se genera automáticamente dependiendo de los formularios realizados que tiene el paciente. Para cada formulario se muestran los resultados de los componentes QOL, IMD, PHP, BWC y ANO en forma de barras. Se incluye en forma de línea el resultado total de ese formulario. Este tipo de gráfica es muy útil para ver la evolución de un paciente.

5.2.5. Doctor: Actualizar datos personales

UPDATE USER

Name	Casco_updated
Lastnames	casco_updated
Country	
Gender	

Update

Figura 16: Update profile.

En la figura 16 se muestra la funcionalidad de modificar datos personales. Cada doctor tiene la posibilidad de modificar sus datos personales si está identificado en la aplicación. De esta manera le damos parte de control de sus propios datos sin tener que contactar con el administrador.

5.2.6. Administrador: Tabla de pacientes y formularios

	Doctor	Patinet	Date	Result
1	angiebetan	ITCA23/11/1953	2012/02/02	22
2	angiebetan	ITME07/07/1951	2012/02/08	45.5
3	angiebetan	ITRG03/02/1941	2012/02/02	30.5
4	angiebetan	ITSH08/05/1941	2012/02/22	18.5
5	Anne	01001	2013/12/23	0
6	casco	34	2018/12/08	58.75
7	casco	BRA01	2018/03/07	52
8	casco	BRA01	2018/03/08	52
9	casco	BRA02	2018/03/07	0
10	casco	Silvia	2018/04/07	0
11	casco	PANA001	1972/01/20	0
12	casco	silvia 3	2018/04/20	0
13	casco	ESPSBR20011972	2018/05/1	0
14	casco	ESPSBR20011972	2018/05/13	0
15	casco	ESPSBR20011972	2018/05/17	0
16	casco	ESPSBR20011972	2018/05/20	0
17	casco	ESPSBR20011972	2018/05/24	9
18	casco	ESPSBR20011972	2018/05/30	48.5
19	casco	ESPSBR20011972	2018/06/11	0
20	casco	ESPSBR20011972	2018/06/14	0

Figura 17: Tabla pacientes y formularios.

En la figura 17 tenemos una tabla donde se muestra todos los formularios ordenados por doctor. Aparte de sus numerosos datos que disponemos del formulario se visualizan los más relevantes: el doctor que ha generado el formulario, el paciente, fecha y puntuación final del formulario.

Como hay cientos de formularios se ha incorporado una paginación para no mostrar todos a la vez, ya que suponía cargar una página web con cientos de resultados. De esta manera mostramos 20 formularios y con la paginación el usuario puede navegar por la tabla.

5.2.7. Administrador: Tabla de usuarios

Doctor	Role	Name	LastName	Country	Gender
admin	admin	Admin			
albert4	user	albert	last	coun	other
anglebetan	user	Angelica	Betancourt	España	
Anne	user	Anne	van der Werf	Netherland	female
carola	user	Joanna	Carola	Brasil	
casco	user	Casco_updated	casco_updated		
Feeney	user	Sinéad	Feeney	Canada	
gues	resea	guest	guest	guest	guest
guest	resea	guest_Name	guest_lastname	guest_country	guest_gender
guest2	resea	guest2_updated	guest2	guest2	guest2
Isabelcorreia	user				
jacoba	user	Ysis Angeline	P. Jacoba	Philippine	female
Javier	user	Javier	Gómez Sánchez	Spain	Male
joanna	user	Joanna	Carola	Brasil	
ProtecUB	user	Protecció de Dades UB		Spain	
Robertoserpe	user	Roberto	Serpe	Italy	
Rodolfo	user	Rodolfo	Gonzalez Camargo	Brazil	
Saunjoo	user	Saunjoo	Yoon	USA	female
seelaender	user	Marilia	Seelaender	Brasil	
Simoes	user	Estefania	Simoes	Spain	
sina	user	Sina	Zobel	Germany	

Figura 18: Tabla pacientes y formularios.

En la figura 18 se muestra una sección que dispone el administrador. Tiene la posibilidad de ver todos los usuarios que están registrados en la aplicación, así como sus datos más relevantes.

Anteriormente no se podían ver los usuarios de la aplicación. Se gestionaba de manera externa. De esta manera se puede ver a tiempo real que usuario está registrado.

5.2.8. Administrador: Modificar datos de usuarios

Figura 19: Modificar datos de usuario

The screenshot shows a web form titled "UPDATE USER". At the top, there is a label "Identification Doctor" followed by a dropdown menu containing the text "albert4" and a small downward arrow. To the right of the dropdown is a dark grey button with the text "Update". Below this is a large grey-bordered box containing several input fields. The first two are "Password*" and "Retype password*", both with red error bars and the text "Password can't be empty" to their right. Below these are four text input fields: "Name" with the value "albert", "Lastnames" with the value "last", "Country" with the value "coun", and "Gender" with the value "other". At the bottom of the grey box is a long, empty text input field.

Como su nombre indica, esta implementación facilita al administrador la modificación de datos de cualquier usuario. Como no se tenía ninguna funcionalidad similar, se ha creído conveniente disponer de ella para un mayor control del administrador.

La idea base de esta funcionalidad es modificar la contraseña de cualquier usuario en caso de pérdida. Anteriormente si esto sucedía no se podía recuperar por lo que es una nueva implementación muy importante. La figura 19 representa esta funcionalidad en la aplicación.

5.2.9. Researcher : Dashboard pie

En la sección del researcher tenemos casi los mismos gráficos que encontramos en los apartados del administrador y del doctor. La diferencia es que para el researcher no están filtrados por pacientes. Se muestra un apartado para no repetir información.

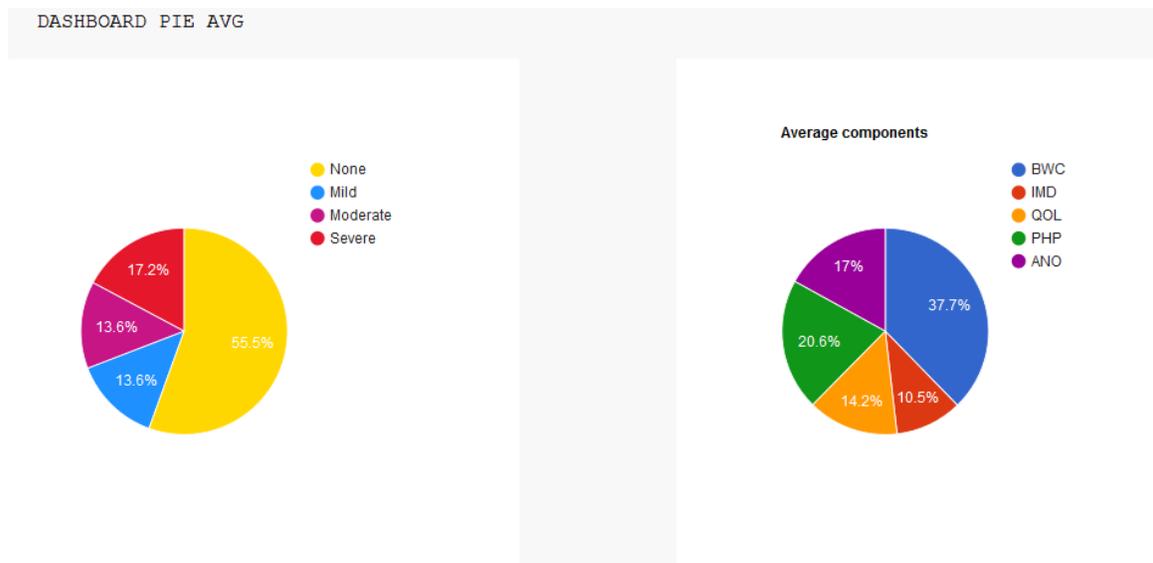


Figura 20: Dashboard pie researcher

En la figura 20 mostramos una de las secciones del researcher. La primera 'pie' es igual a la figura 13. Obtiene la puntuación de los formularios y los clasifica según la severidad de la caquexia.

Por otro lado, incluimos otro 'pie' para ver el porcentaje que tienen cada componente QOL, BWC, PHP, IMD y ANO calculando la media de cada atributo entre todos los formularios.

5.2.10. Researcher and doctor: Dashboard tree

En la figura 21 se muestra la dashboard tree. Es una gráfica árbol y esto quiere decir que el usuario puede interactuar con ella pulsando una sección de la gráfica. En este caso tiene la posibilidad de pulsar Male o Female. En cualquier caso, entrara dentro de esa sección y se le muestra los apartados que están en el interior, así como sus valores.

La finalidad de esta visualización es mostrar a que género afectan más los distintos tipos de cáncer y ver a qué nivel de severidad están sus pacientes.

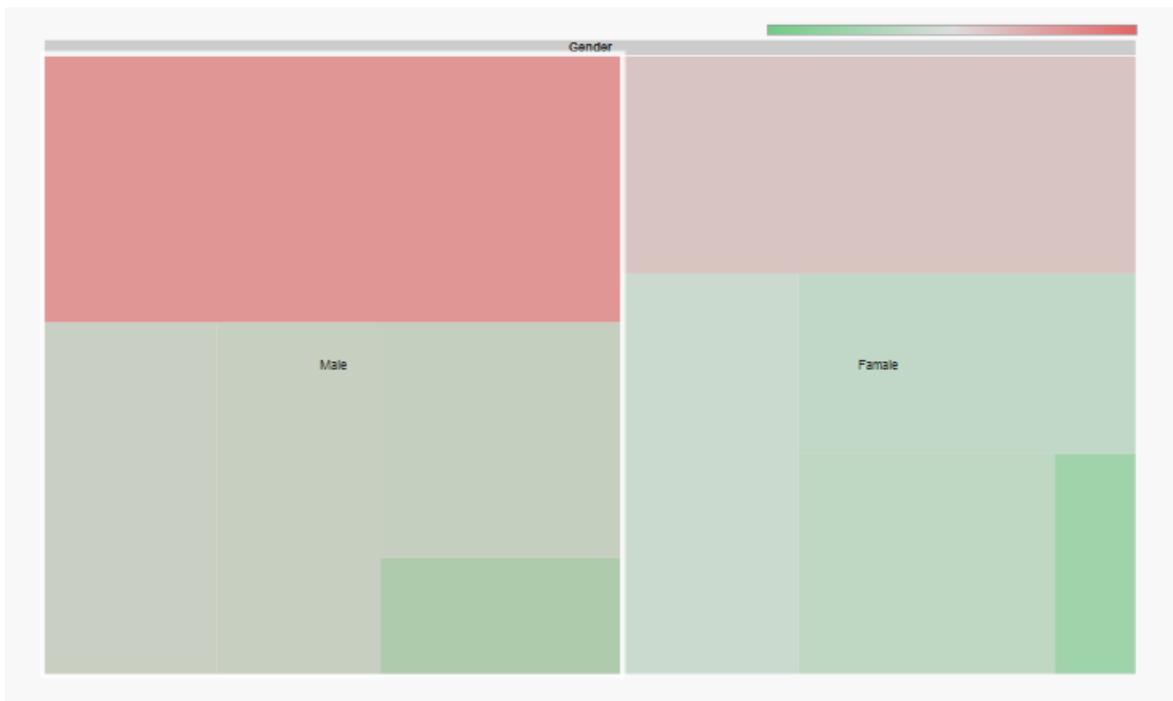


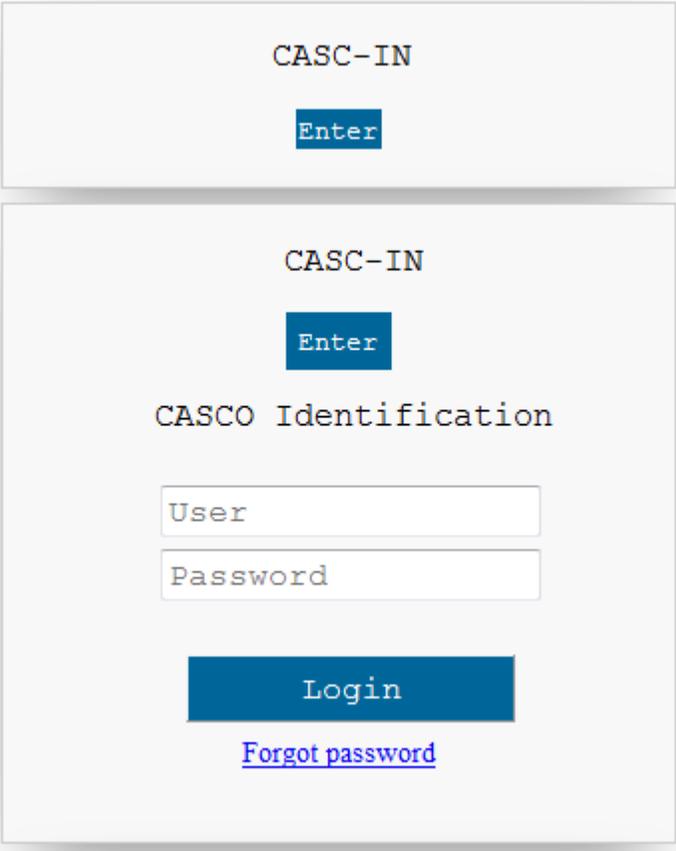
Figura 21: Dashboard tree

5.2.10. Login

La página inicial de la aplicación ha sido modificada para tener un diseño más atractivo e intuitivo para el usuario. Inicialmente se tenía que entrar en una sección aparte para poder entrar a la aplicación con el nombre y contraseña. Se ha modificado para poner el login en la página inicial dejando la información que ya existía.

Con ello conseguimos que el usuario sepa donde identificarse para entrar en la aplicación sin necesidad de navegar por la web de la herramienta. En la figura 22 se visualiza la entrada del login y la del CASC-in.

También se puede visualizar que debajo del botón 'Login' tenemos la opción de recuperar la contraseña



The image displays two panels of a login interface. The top panel is a light gray box containing the text "CASC-IN" and a blue button labeled "Enter". The bottom panel is a larger light gray box containing the text "CASC-IN", a blue button labeled "Enter", the text "CASCO Identification", two input fields labeled "User" and "Password", a blue button labeled "Login", and a blue underlined link labeled "Forgot password".

Figura 22: Login

5.2.11. Recuperación de contraseña

En la figura 23 se muestra el apartado de recuperación de contraseña. Tiene un diseño muy simple y se está modificando para que tenga una visualización más agradable para el usuario. Como ya hemos explicado anteriormente, el usuario deberá rellenar todos los campos correctamente para poder realizar el cambio de contraseña con éxito.

Recovery Password

Identification

Lastname

Country

New password

[return to Index](#)

Figura 23: Login

En el caso de tener éxito en la modificación se muestra un mensaje informativo al usuario, la fecha y se le comunica que se le envía un email al administrador para indicar la modificación.

User: casco

Password changed on Thu, 26 Jan 2017 14:22:33 +0100

Sended email to administrator.

For more info contact to gestion.casco@gmail.com.

[return to Index](#)

Casco Recovery Failed.

User identification or data are incorrect.

For more info contact to gestion.casco@gmail.com.

[return to Index](#)

Figura 24: Mensaje de modificación correcto no correcto

Figura 25: Mensaje de modificación

En la figura 24 tenemos el mensaje que recibe el usuario al modificar su contraseña y en la figura 25 el mensaje que recibe en el caso de no realizar la modificación. Ambos mensajes disponen de un link para volver a la página inicial por si el usuario ha llegado a esa página por error.

5.2.12. Http a https

La aplicación realizaba todas las operaciones vía http. Http es un protocolo de transferencia de datos por internet. Este protocolo permite indicar a las aplicaciones como enviar o recibir las páginas web o datos a nuestra aplicación.

Cuando los paquetes viajan a través de internet es muy sencillo captar esos paquetes de datos. Si esos paquetes no son seguros pueden ver la información contenida en ellos y esto puede llegar a ser muy grave. Pueden ver contraseñas, cuentas bancarias etc.

Migrando a https se consigue que esos paquetes queden encriptados y no sea posible abrirlos, por lo que nuestros datos estarán seguros.

La aplicación no disponía de un protocolo seguro para la transferencia de datos por internet. La idea de migrar a https fue determinada por la funcionalidad de recuperar la contraseña.

Cuando el usuario escribe su contraseña nueva e inicia la acción de registrarla. La contraseña viaja por internet unos instantes y podía ser visible para cualquiera que interceptara el paquete.

Actualmente ya funciona por https y los datos no pueden ser descifrados ni visibles por cualquiera que capture dichos paquetes.

6. CONCLUSIONES

Se ha conseguido finalizar los objetivos del proyecto dotando a la aplicación web CASCO de una serie de funcionalidades nuevas para facilitar su uso a los usuarios de la aplicación.

Los objetivos más importantes que se han conseguido implementar en la gestión de datos y la visualización:

- Se ha migrado de http a https.
- Los administradores:
 - Tienen mayor control sobre los usuarios. Pueden ver los usuarios registrados. Tienen la funcionalidad de modificar sus datos y pueden borrar usuarios ya sean administradores, doctores o researchers.
 - Referente a los formularios, pueden ver todos los datos en la base de datos de cualquier formulario de manera rápida y simple.
 - Pueden exportar formularios en distintos tipos de documento para disponer de más opciones.
- Los doctores:
 - Tienen a su disposición nuevas graficas que da soporte a la hora de hacer un seguimiento de sus pacientes. Con estas gráficas, también pueden visualizar la evolución de sus pacientes de manera ágil.
 - Son capaces de modificar sus propios datos en cualquier momento.
- Los researchers:
 - Nuevo rol de la aplicación con la finalidad de estudiar o investigar los datos que hay registrados en ella.
 - Cuanta con la ayuda de gráficas y tablas para visualizar todos los datos disponibles.

6.1 Futuras mejoras

Se ha pensado en nuevas implementaciones para que esta herramienta siga evolucionando. En este apartado nombramos unas cuantas ideas que podrían hacer mejorar la aplicación.

La aplicación podría contar con un sistema de control de acceso de usuarios. La idea es que cada vez que un usuario se identifique en CASCO, guardar un registro de si inicio. Parece una idea muy simple, pero puede solucionar temas importantes si desaparecen registros o sucede alguna anomalía en los datos. Con ello conseguiríamos saber los usuarios que han accedido a casco en cierta fecha y poder investigar si sucediera algún problema o simplemente para tener un seguimiento de los usuarios y sus accesos.

Otra mejora que se podría implementar es tener un sistema donde se visualice el estado del servidor de la UB o el servidor en el que está alojada la aplicación. Visualizar si está respondiendo con normalidad o si alguna de sus máquinas ha caído, cuando, porque etc. Esto no es responsabilidad de la aplicación CASCO, pero nunca esta demás controlar mínimamente el sitio en el que está funcionando la aplicación para evitar problemas y asegurar la integración de datos que se tiene en ella.

Otra idea es realizar una investigación para la interacción del usuario con la aplicación. No es implementar ninguna funcionalidad nueva si no realizar un proyecto basado en 'use experiencer'. Investigar como modificar los interfaces de usuario según los propios usuarios. Donde suelen buscar para acceder a CASCO, donde suelen buscar cierta información, que datos les gusta visualizar a simple vista etc. Con ello conseguiríamos unas interfaces de usuario más amigables para las personas que utilicen esta herramienta.

6.2. Valoraciones personales

- Personales:
 - Me ha gustado mucho formar parte de este proyecto y poder aplicar conocimientos aprendidos a lo largo de la carrera.
 - Es interesante estar en un proyecto medio-grande en el que has de investigar, implementar, testear, planificar, solucionar y comunicarte con el resto de equipo. Durante la carrera se realizan proyectos, pero son bastante más pequeños.
 - Un punto muy a favor de hacer un TFG y que estoy muy agradecido es que ganas mucha experiencia respecto en lo que hayas trabajado.

Es muy importante para los estudiantes que finalizamos nuestra carrera aprender todo lo posible y obtener conocimientos. Por mi parte, no había programado nunca con PHP y muy poco en aplicaciones web y estoy contento de haberlo hecho.

- En definitiva, estoy contento por haber realizado este proyecto y poder ayudar al Silvia en su herramienta. Aunque desconozco el ámbito de la medicina me ha gustado y era un punto por el que elegí este trabajo final de grado.
- Proyecto:
 - En mi opinión la aplicación CASCO está a un nivel decente y se puede trabajar en ella de manera estable.
 - Dispone de todas las funcionalidades básicas que una aplicación web necesita tener, aparte de las específicas de casco.

Creo que es una aplicación que tiene mucho potencial y puede mejorarse enormemente a lo largo del tiempo.

7. BIBLIOGRAFIA

- <http://www.w3schools.com/php/>
- <http://www.w3schools.com/jquery/>
- <https://dev.mysql.com>
- <https://secure.php.net/manual/es/index.php>
- <http://www.desarrolloweb.com>
- <https://www.gliffy.com>
- <https://developers.google.com/chart/interactive/docs/gallery>
- <https://jquery.com/>
- <http://www.chartjs.org/docs/>
- <https://github.com/MrRio/jsPDF>
- Betancourt Suarez, A. (2015). DESARROLLO DE UNA ESCALA NUMÉRICA PARA CLASIFICAR GRADOS DE CAQUEXIA EN CÁNCER. Tesis de Doctorado. Universitat de Barcelona.
- Tesis Doctoral Marta Villarino Sanz

8. ANEXOS

En los anexos se quiere detallar explicaciones de los casos de uso ya mencionados en su correspondiente sección para ampliar la información de cada proceso. También en consta de las valoraciones propias del autor del trabajo en la última sección de este apartado.

8.1 Descripción detallada de los casos de uso.

En esta sección se explica los casos de uso descritos en las figuras 2,3 y 4 con más detalle. Indicando el nombre, la versión (con esto se indica si el caso de uso ha sido modificado y por ello es una segunda versión o por el contrario no se ha modificado o ha sido un nuevo caso. Entonces es una versión que se indica con '1.0')

	LogIn	
Versión	2.0	
Objetivo	Identificarse en la aplicación CASCO.	
Descripción	Permitir el acceso a CASCO a los actores registrados previamente por un administrador. Dependiendo del rol del actor y de si es un actor con la contraseña valida, accederá a su zona correspondiente.	
Modificación	Previamente los actores no tenían el atributo 'role' implementado por lo que la diferencia entre entrar en la sección doctor y admin era el nombre del actor. Si era 'admin' literalmente, entraba a la sección del administrador y si no a la del doctor. Se ha añadido dicho atributo a los actores para tener una capa más de seguridad referente a los usuarios y a la aplicación misma.	
Actores	Administrador, Doctor y Researcher	
Secuencia	Paso	Acción

	1	Escribir el usuario y contraseña en el apartado del login que se muestra en la página inicial de CASCO.
	2	Seleccionar el botón Login
Excepciones	Paso	Acción
	1	Si el usuario no es válido, se muestra la página inicial.
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Acceder a la web inicial de CASCO.
	2	No ha de estar identificado.
	3	Ha de conocer su usuario y password.
	4	Debe de haber sido registrado por un administrador previamente.
Resultado	El actor accederá al apartado correspondiente a su rol.	

	LogOut
Versión	2.0
Objetivo	Salir de la aplicación CASCO.
Descripción	Permitir a los actores salir de la aplicación. Se borra la sesión y las variables en caché.
Modificaciones	En la versión posterior solamente se borraba la sesión. Al no vaciar las variables internas, están seguían con los valores del anterior actor logueado. Esto implicaba que al refrescar la página inicial, se

	ejecutase el login y obtenía los valores de están variables de la aplicación y se entraba a CASCO. Por seguridad se ha implementado esta modificación. El resto de ejecución es la misma que en la versión 1.0.	
Actores	Administrador, Doctor y Researcher	
Secuencia	Paso	Acción
	1	Selección de Exit en el menú principal.
Excepciones	Paso	Acción
	1	No hay excepciones.
Precondiciones	Paso	Condición
	1	Estar previamente logueado.
Resultado	El actor saldrá de su zona en la aplicación y se le redirige a la página inicial de CASCO.	

	Recuperación de password
Versión	1.0
Objetivo	Dar la posibilidad a los usuarios de generar una nueva contraseña para poder identificarse en la aplicación.
Descripción	Los usuarios que hayan olvidado la contraseña, tiene la posibilidad de contactar con el administrador y que este se la modifique. En el caso de que el administrador no esté disponible se ha implementado esta funcionalidad para que el usuario pueda generar una nueva contraseña al momento. Una vez sea capaz de generarla, se le comunica al administrador vía email del usuario que ha modificado la contraseña.

	El usuario que quiera cambiar su contraseña deberá recordar sus datos personales 'lastname' y 'country'. Estas dos palabras sirven de palabra clave para dar seguridad al cambio de password.	
Modificaciones	-	
Actores	Administrador, Doctor y Researcher	
Secuencia	Paso	Acción
	1	Clicar en 'Forgot password'
	2	Rellenar los campos que se solicita.
	3	Clicar a 'Recovey'
Excepciones	Paso	Acción
	1	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
	2	Si el usuario no es válido, se le informa que no ha sido posible el cambio y si desea puede contactar con el administrador.
Precondiciones	Paso	Condición
	1	Ser usuario de CASCO
Resultado	El usuario se le muestra una notificación del resultado del cambio. Si ha funcionado correctamente se le informa al usuario que su contraseña ha sido modificada y se envía un email al administrador indicando del usuario y fecha.	

	Registrar Doctor
--	------------------

Versión	2.0	
Objetivo	Registrar a un nuevo usuario.	
Descripción	El administrador puede registrar un nuevo usuario para que este pueda loguearse en CASCO.	
Modificaciones	Como hemos añadido el atributo 'role' a los usuarios, es necesario que el administrador indique el rol a la hora de registrar el nuevo usuario. Podrá elegir entre uno de los tres roles disponible por el momento (Administrador, Doctor, Researcher).	
Actores	Administrador	
Secuencia	Paso	Acción
	1	Pulsar el botón Register user.
	2	<ul style="list-style-type: none"> - Añadir nombre valido. - Añadir contraseña valida. - Seleccionar rol. - Rellenar los campos adicionales para más información (opcional)
	3	Pulsar Register.
Excepciones		Acción
	1	Si existe algún error en el formulario que debe rellenar para registrarlo, se informa del error al administrador.
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como administrador.
	2	Rellenar correctamente el formulario.

Resultado	Registro en la BBDD de un nuevo usuario que tendrá acceso a CASCO.
-----------	--

	Modificar Doctor	
Versión	1.0	
Objetivo	Modificar los datos del doctor.	
Descripción	Permitir al administrador que pueda modificar cualquier dato de un usuario ya registrado en la aplicación.	
Modificaciones	-	
Actores	Administrador	
Secuencia	Paso	Acción
	1	Pulsar el botón Update user.
	2	Seleccionar el usuario y pulsar Modify.
	3	<ul style="list-style-type: none"> - Añadir contraseña valida. - Rellenar los campos adicionales para más información (opcional)
Excepciones		Acción
	1	Si los campos de la contraseña son indebidamente rellenos se le informa de este error.
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como administrador.

	2	Saber que usuario queremos modificar.
	3	Rellenar los campos del formulario correctamente.
Resultado	Registro en la BBDD de un nuevo usuario que tendrá acceso a CASCO.	

(UC-06 y UC-07) son acciones distintas, pero se llevan a cabo en el mismo contexto. Por esa razón se describen en la misma tabla.

UC-06 y UC-07	Borrar Doctor y transferir pacientes	
Versión	1.0	
Objetivo	Borrar un doctor de la BBDD y pasar sus pacientes/formularios a un doctor activo en la aplicación.	
Descripción	Permitir al administrador que pueda borrar un doctor para que no pueda identificarse en la aplicación CASCO. En este caso debemos transferir sus pacientes a otro doctor para que los tenga presentes por si el realizara su seguimiento o por las razones que el administrador crea oportunas.	
Modificaciones	-	
Actores	Administrador	
Secuencia	Paso	Acción
	1	Pulsar el botón Delete Doctor.
	2	Seleccionar el doctor a borrar.
	3	Seleccionar el doctor a transferir sus pacientes.
	4	Pulsar Delete and Transfer

Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como administrador.
	2	Saber que usuario queremos borrar.
	3	Saber a qué usuario transferir los pacientes.
Resultado	Borrado de un doctor de la BBDD y transferir sus pacientes al doctor seleccionado.	

	Visualización de usuarios	
Versión	1.0	
Objetivo	Visualizar los usuarios de CASCO	
Descripción	Permitir al administrador visualizar todos los usuarios que tienen acceso a la aplicación, así como sus datos más relevantes.	
Modificaciones	-	
Actores	Administrador	
Secuencia	Paso	Acción
	1	Pulsar el botón Table Doctors.
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.

Precondiciones	Paso	Condición
	1	Estar previamente logueado como administrador.
Resultado	Visualización de una tabla con los datos de usuarios.	

	Exportar CSV	
Versión	2.0	
Objetivo	Permitir la exportación de los formularios en un Excel.	
Descripción	Permitir al doctor exportar en un Excel el formulario del paciente que este modificando. En el caso del administrador puede seleccionar un paciente y posteriormente se exportarán todos sus formularios.	
Modificaciones	Al exportar en distintos sistemas operativos no realizaba correctamente el export y quedaban columnas con datos de otras. Se ha modificado y testeado para su correcto funcionamiento en Windows, MAC y Linux.	
Actores	Doctor y Administrador	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Compare.
	2	Seleccionar paciente
Excepciones		Acción

	Modificar doctor	
Versión	1.0	
Objetivo	Modificar información del propio doctor.	
Descripción	Permitir al doctor la modificación de su información personal.	
Modificaciones	-	
Actores	Doctor	
Secuencia	Paso	Acción
	1	Pulsar el botón Update Profile.
	2	Rellenar campos a modificar.
	3	Pulsar Update.
	4	Se informa al usuario de su modificación.
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Modificación en la BBDD del doctor que ha realizado la acción.	

	Dashboard pie	
Versión	1.0	
Objetivo	Visualizar datos en una gráfica tipo pie.	
Descripción	Visualizara en una gráfica pie cuantos formularios hechos tienen un valor de Severo, moderado, mediano o nada respecto a la puntuación del índice de CASCO.	
Modificaciones	-	
Actores	Doctor	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Pie.
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la gráfica pie con los datos comentados en la descripción.	

UC-13	Dashboard Tree (Modificando)	
Versión	1.0	
Objetivo		
Descripción		

Modificaciones	-	
Actores	Doctor	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Tree.
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la gráfica Tree con los datos comentados en la descripción.	

	Dashboard Bubble	
Versión	1.0	
Objetivo	Visualizar datos en una gráfica tipo bubble.	
Descripción	Visualizara datos en una gráfica bubble. Dicha grafica consiste en observar por cada año, cuantos formularios se han realizado, su fecha, su paciente y su score.	
Modificaciones	-	
Actores	Doctor	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Bubble.

Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la gráfica bubble con los datos comentados en la descripción.	

	Dashboard compare	
Versión	1.0	
Objetivo	Visualizar datos en una gráfica tipo barras y líneas.	
Descripción	Visualizar en una gráfica los diferentes formularios realizados por un mismo paciente ordenado por fecha para poder ver su evolución. La grafica consta de distintas barras que hacen referencia a los componentes de los cálculos (IMD, PHP, BWC, ANO y QOL) y una línea que proporciona la puntuación final.	
Modificaciones	-	
Actores	Doctor	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Compare.
	2	Seleccionar paciente
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.

Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la evolución del paciente en la gráfica comentada en la descripción.	

	Exportar Excel	
Versión	2.0	
Objetivo	Permitir la exportación de los formularios en un Excel.	
Descripción	Permitir al doctor exportar en un Excel el formulario del paciente que este modificando. En el caso del administrador puede seleccionar un paciente y posteriormente se exportarán todos sus formularios.	
Modificaciones	Al exportar en distintos sistemas operativos no realizaba correctamente el export y quedaban columnas con datos de otras. Se ha modificado y testeado para su correcto funcionamiento en Windows, MAC y Linux.	
Actores	Doctor y Administrador	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Compare.
	2	Seleccionar paciente
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.

Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la evolución del paciente en la gráfica comentada en la descripción.	

	Dashboard pie	
Versión	1.0	
Objetivo	Visualizar datos en una gráfica tipo pie.	
Descripción	Visualizara en una gráfica pie los porcentajes que tiene los componentes PHP, QOL, IMD, BWC y ANO de todos los formularios independientemente del doctor.	
Modificaciones	-	
Actores	Researcher	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Pie.
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la gráfica pie con los datos comentados en la descripción.	

	Dashboard Bubble	
Versión	1.0	
Objetivo	Visualizar datos en una gráfica tipo bubble.	
Descripción	Visualizara datos en una gráfica bubble del doctor que se seleccione. Dicha grafica consiste en observar por cada año, cuantos formularios se han realizado, su fecha, su paciente y su score.	
Modificaciones	-	
Actores	Researcher	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Bubble.
	2	Seleccionar doctor
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la gráfica bubble con los datos comentados en la descripción.	

	Dashboard compare	
Versión	1.0	
Objetivo	Visualizar datos en una gráfica tipo barras y líneas.	
Descripción	Visualizar en una gráfica los diferentes formularios realizados por un mismo paciente ordenado por fecha para poder ver su evolución. La grafica consta de distintas barras que hacen referencia a los componentes de los cálculos (IMD, PHP, BWC, ANO y QOL) y una línea que proporciona la puntuación final.	
Modificaciones	-	
Actores	Researcher	
Secuencia	Paso	Acción
	1	Pulsar el botón DashBoard Compare.
	2	Seleccionar paciente
Excepciones		Acción
	2	No hay conexión con la BBDD. Se informa al usuario con un mensaje.
Precondiciones	Paso	Condición
	1	Estar previamente logueado como doctor.
Resultado	Visualiza la evolución del paciente en la gráfica comentada en la descripción.	