



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Memoria

Herramienta para la gestión de datos, pacientes y agenda del grupo GICOREC

Trabajo de fin de grado

Autor: Marina León Hernández

Tutor: Francisco José Santana Pérez

(Director de la Escuela de Ingeniería informática)



06 de septiembre de 2013

Las Palmas de Gran Canaria

Índice

1. Estado actual y objetivos -----	2
1.1.1.Introducción -----	2
1.1.2.Estado actual -----	3
1.1.3.Objetivos -----	4
2. Competencias que hay que cubrir -----	5
3. Aportaciones -----	7
4. Normativa y legislación -----	8
5. Análisis -----	11
5.1. Casos de uso -----	13
5.2. Detalles de casos de uso -----	16
6. Diseño -----	19
7. Desarrollo -----	22
7.1.1.Base de datos -----	23
7.1.2.Clase de apoyo TemplatePower -----	26
7.1.3.Clase de apoyo TCPDF -----	29
7.1.4.Implementación -----	30
8. Limitaciones y mejoras futuras -----	32
9. Requisitos hardware y software -----	34
10. Modelo de negocio -----	35
11. Bibliografía -----	36
12. ANEXO: Manual de usuario --	37

1. Estado actual y objetivos

Introducción

Este trabajo de fin de grado consiste en el desarrollo de una aplicación web en PHP destinada al grupo GICOREC.

GICOREC es un grupo de la facultad de veterinaria que, entre otras actividades, se encargan de recibir y atender a mascotas en el hospital de dicha facultad, remitidas por otro hospital o clínica veterinaria. Todos los animales que les son remitidos deben ser registrados así como a sus propietarios, datos de contacto e información sobre el veterinario y la clínica remitente.

Cada mascota o paciente, tendrá un número de historia asignado además debe hacer referencia al número de DNI de su propietario. Hay que tener en cuenta de que para un mismo propietario, pueden existir varias mascotas registradas en el sistema, por lo que para cada propietario debemos poder consultar el listado de mascotas que tiene asignadas, pudiendo posteriormente consultar los datos de cada animal en su ficha correspondiente, así como los datos de contacto.

Por una parte se tendrán los datos básicos del paciente y por otra, los detalles referentes a las citas o exámenes que le han sido realizados en dicho hospital, estos deberán quedarse registrados en su historial y se podrán consultar en cualquier momento, para cada uno de ellos se podrán adjuntar documentos o imágenes. Además de los detalles de las citas, dado el caso, deberá llevarse un registro aparte para las cirugías que se realicen, para ello a la hora de crear una nueva cita, se deberá especificar si se trata de una consulta o de un proceso quirúrgico.

En cuanto a los productos y materiales utilizados en cada proceso, deben existir en el almacén del hospital, de donde serán descontados para, posteriormente añadir su importe a la facturación del proceso junto con los demás gastos que le serán cobrados al propietario del animal.

Los veterinarios responsables tendrán acceso al sistema y además podrán añadir nuevas citas desde la agenda, donde deberán especificar fecha, hora y el número de historia

del paciente para el que va destinada la cita. Posteriormente, una vez concluida la cita, los datos que esta aporte serán almacenados para su posterior consulta.

Estado actual

Actualmente no se dispone de ninguna aplicación que realice la gestión de todos los apartados anteriormente mencionados, todas las fichas de pacientes, control del almacén, citas y archivos, se realizan en papel son almacenadas en archivadores. La facturación también se realiza a mano y para las entradas y salidas de caja se utiliza una hoja de Microsoft Excel.

Automatizar estas tareas en una única aplicación hace posible ahorrar tiempo y recursos además de que resultarán más cómodas para el usuario.

En la actualidad existen numerosas aplicaciones destinadas al control de stock, ventas, facturación y contabilidad, como ejemplo tenemos el Openerp, pero dadas las necesidades de este caso en particular y el ámbito de trabajo, que será puramente hospitalario, resultaría más complicada de utilizar y configurar, necesitaría muchos más recursos que una sencilla aplicación web que además está hecha de forma específica ser utilizada en un hospital veterinario y al usuario le será sencilla de comprender.

Objetivos

El objetivo de este trabajo de fin de grado es desarrollar una aplicación web en PHP destinada al hospital veterinario de la Universidad de Las Palmas de Gran Canaria que facilitará las labores de registro de pacientes en general, propietarios, gestión de citas y consultas, gestión de los datos relacionados con las consultas así como de productos y facturas. En un principio, tanto la aplicación como su base de datos estarán alojadas en un servidor de la propia facultad, aunque se contempla la posibilidad de contratar un dominio web.

Podrá ser ejecutada desde el PC o una Tablet ubicada en la consulta o en el quirófano, tan solo será necesario acceder a la URL de esta e identificarse como usuario de la aplicación.

Todos los veterinarios usuarios de la aplicación podrán añadir propietarios, pacientes, crear citas para su posterior realización, gestionar el almacén y consultar los datos que necesiten.

Las citas se crearán para un determinado paciente que debe estar registrado en el sistema. Una vez creada una cita se guarda con estado “pendiente” y será el veterinario al terminar la misma quien la realice añadiendo los datos médicos pertinentes, los servicios ofrecidos, subir archivos si se da el caso y añadir productos (en caso de que sea necesario) que se han utilizado en esa cita, cambiando el estado de la cita a “realizada” generándose así el posterior informe médico y la correspondiente factura. Asimismo los datos médicos obtenidos de esta quedarán archivados en una sección llamada “Casos” pudiendo ser consultada con posterioridad con fines médicos.

2. Competencias que hay que cubrir

En este trabajo deben cumplirse una serie de competencias, asociadas a este, en concreto son: CII01, CII02, CII04, CII08, TFG01.

A continuación se detalla cada una de ellas con una breve explicación de cómo han sido cubiertas.

CII01: *capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.*

Este trabajo de fin de grado consiste en el diseño y desarrollo de una aplicación web destinada a un grupo de usuarios concreto para responder a una serie de necesidades de su entorno de trabajo. En todo momento se ha asegurado de que este sea fiable y cumpla las especificaciones éticas y la normativa vigente, velando por su seguridad y calidad.

CII02: *Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.*

Para la realización de este trabajo el alumno debió realizar su planificación y ejecución. También fue necesario que el alumno desplegara la aplicación en el entorno donde será utilizado, en concreto un servidor de la facultad de veterinaria de la ULPGC. Se han tenido en cuenta las posibles mejoras que se podrán aplicar para futuras versiones de la aplicación.

CII04: *Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.*

Se han tenido en cuenta los requisitos hardware y software necesarios para el desarrollo de la aplicación y su implantación, así mismo estos se han detallado en la presente memoria.

CII18: *Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.*

La normativa y legislación aplicable a este proyecto ha sido detallada en la presente memoria, siendo su conocimiento fundamental para el correcto desarrollo del mismo.

TFG01: *Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.*

El alumno realizará la correspondiente presentación y defensa del presente trabajo de fin de grado ante un tribunal. Para la elaboración del proyecto han sido necesarios conocimientos que abarcan diversas áreas del ámbito de la ingeniería informática.

3. Aportaciones

El presente trabajo de fin de grado ha sido desarrollado con la intención de poder ofrecer a un entorno científico-hospitalario, la posibilidad de mantener de forma óptima y segura un registro de sus principales actividades y usuarios.

Actualmente la informática se extiende por todo tipo de empresas y es importante acomodarse a las nuevas tecnologías, al tratarse de una aplicación web que correrá en un servidor, el usuario o veterinario tendrá total libertad de utilizarla desde cualquier parte, dentro de la ubicación del hospital, ya sea desde un ordenador de sobremesa o dada la implantación de las tecnologías móviles, desde una Tablet, así no solo se conseguirá implantar un sistema informático de registro y control, del que hasta ahora no disponían, sino que además podrán realizar las tareas desde cualquier zona que se necesite, con total comodidad.

Organizar los datos y tareas de este modo aporta mayor comodidad y eficiencia que la forma que hasta la actualidad se había estado utilizando que consistía que anotar todas las fichas de pacientes y citas en papel, pudiendo llevar así un control más detallado y ordenado de los historiales clínicos, así como hacer que las búsquedas de datos concretos, como pueden ser los de contacto de un propietario, sean mucho más rápidos.

4. Normativa y legislación

Este trabajo de fin de grado está amparado por la Ley Orgánica 15/1999 de 13 de diciembre de Protección de datos de carácter personal (LOPD), tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar. La presente ley en cuanto a su ámbito de aplicación narra lo siguiente:

Artículo 2 Ámbito de aplicación

1. La presente Ley Orgánica será de aplicación a los datos de carácter personal registrados en soporte físico, que los haga susceptibles de tratamiento, y a toda modalidad de uso posterior de estos datos por los sectores público y privado.

Se regirá por la presente Ley Orgánica todo tratamiento de datos de carácter personal:

a) Cuando el tratamiento sea efectuado en territorio español en el marco de las actividades de un establecimiento del responsable del tratamiento.

b) Cuando al responsable del tratamiento no establecido en territorio español, le sea de aplicación la legislación española en aplicación de normas de Derecho Internacional público.

c) Cuando el responsable del tratamiento no esté establecido en territorio de la Unión Europea y utilice en el tratamiento de datos medios situados en territorio español, salvo que tales medios se utilicen únicamente con fines de tránsito.

En cuanto a lo considerado por esta ley como datos de carácter personal:

A los efectos de la presente Ley Orgánica se entenderá por:

a) Datos de carácter personal: cualquier información concerniente a personas físicas identificadas o identificables.

b) Fichero: todo conjunto organizado de datos de carácter personal, cualquiera que fuere la forma o modalidad de su creación, almacenamiento, organización y acceso.

c) Tratamiento de datos: operaciones y procedimientos técnicos de carácter automatizado o no, que permitan la recogida, grabación, conservación, elaboración, modificación, bloqueo y cancelación, así como las cesiones de datos que resulten de comunicaciones, consultas, interconexiones y transferencias.

d) Responsable del fichero o tratamiento: persona física o jurídica, de naturaleza pública o privada u órgano administrativo, que decida sobre la finalidad, contenido y uso del tratamiento.

e) Afectado o interesado: persona física titular de los datos que sean objeto del tratamiento a que se refiere el apartado c) del presente artículo.

f) Procedimiento de disociación: todo tratamiento de datos personales de modo que la información que se obtenga no pueda asociarse a persona identificada o identificable.

g) Encargado del tratamiento: la persona física o jurídica, autoridad pública, servicio o cualquier otro organismo que, sólo o conjuntamente con otros, trate datos personales por cuenta del responsable del tratamiento.

h) Consentimiento del interesado: toda manifestación de voluntad, libre, inequívoca, específica e informada, mediante la que el interesado consienta el tratamiento de datos personales que le conciernen.

i) Cesión o comunicación de datos: toda revelación de datos realizada a una persona distinta del interesado.

j) Fuentes accesibles al público: aquellos ficheros cuya consulta puede ser realizada, por cualquier persona, no impedida por una norma limitativa o sin más exigencia que, en su caso, el abono de una contraprestación. Tienen la consideración de fuentes de acceso público, exclusivamente, el censo promocional, los repertorios telefónicos en los términos previstos por su normativa específica y las listas de personas pertenecientes a grupos de profesionales que contengan únicamente los datos de nombre, título, profesión, actividad, grado académico, dirección e indicación de su pertenencia al grupo. Asimismo, tienen el carácter de fuentes de acceso público los diarios y boletines oficiales y los medios de comunicación. Tienen la consideración de fuentes de acceso público, exclusivamente, el censo promocional, los repertorios

telefónicos en los términos previstos por su normativa específica y las listas de personas pertenecientes a grupos de profesionales que contengan únicamente los datos de nombre, título, profesión, actividad, grado académico, dirección e indicación de su pertenencia al grupo. Asimismo, tienen el carácter de fuentes de acceso público los diarios y boletines oficiales.

En la presente aplicación será necesario almacenar datos referentes a personas físicas o identificables tales como el número de su documento nacional de identidad, nombre, apellidos, dirección, correo electrónico y teléfono, siendo necesario en todo momento avisar al usuario de que estos datos serán almacenados en un registro interno que serán usados para fines concretos relacionados con el ámbito de la veterinaria, como identificación del propietario de las mascotas atendidas o para ponerse en contacto en caso de que sea necesario.

5. Análisis

El patrón de diseño seguido el desarrollo de esta aplicación PHP es el MVC (Modelo Vista Controlador). Este patrón separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar las comunicaciones con la base de datos, proponiendo para ello la construcción de tres componentes por separado, el modelo (componentes encargados del acceso a la información), la vista (interfaz de usuario) y el controlador (clases encargadas de interactuar con los datos y obtenerlos de forma dinámica generando el código HTML).

Flujo de control del MVC:

1. El usuario realiza una acción en la interfaz.
2. El controlador trata el evento de entrada.
3. El controlador notifica al modelo la acción del usuario, lo que puede implicar un cambio del estado del modelo (si no es una mera consulta).
4. Se genera una nueva vista. La vista toma los datos del modelo sin necesidad de que este tenga conocimiento directo de la vista.
5. La interfaz de usuario espera otra interacción del usuario, que comenzará otro nuevo ciclo.

Roles de usuario

Esta aplicación existe un único rol de usuario que será el de veterinario y el de administrador general de la aplicación.

Todos los veterinarios del grupo GICOREC tendrán acceso a la aplicación mediante su nombre de usuario y contraseña y podrán acceder a todo su contenido, ya sea añadir datos, eliminarlos, crear o eventos.

Funcionalidad

- **Almacén:** Mostrará un listado de productos y proveedores, siendo posible editar los campos de ambos y añadir nuevos registros.
- **Pacientes:** Permite ver un listado con todas las mascotas registradas, realizar ediciones

o eliminarlas, además desde aquí se podrá consultar su ficha donde podremos comprobar los datos de contacto de su propietario y su historial de citas.

- **Propietarios:** Al igual que en pacientes, desde aquí podremos añadir nuevos propietarios, editar o eliminarlos, así como consultar las mascotas pertenecientes a un usuario ya existente.
- **Casos:** Lleva un registro de todos los casos atendidos en el hospital permitiendo consultar de forma rápida todos sus detalles médicos y archivos en caso de que hayan sido agregados en el momento de realizar la cita.
- **Agenda:** Desde esta sección el veterinario podrá añadir nuevos eventos o citas y consultar las ya realizadas. Además será desde aquí donde se concluya la cita, seleccionando la que corresponda de la lista, pudieron hacer las anotaciones correspondientes en el transcurso de esta y posteriormente, finalizarla generando la correspondiente factura en PDF y un informe con los detalles de la cita.
- **Facturación:** Llevará un registro de todas las facturas generadas tras una cita o consulta permitiendo su posterior impresión.
- **Añadir nuevo usuario:** Desde aquí un veterinario ya registrado podrá añadir nuevos usuarios a la aplicación. El encargado de establecer una contraseña será el propio usuario la primera vez que se identifique en el sistema.

Casos de uso

Se consideran los siguientes casos de uso:

- A. Gestión de usuarios de la aplicación.
- B. Gestión de pacientes.
- C. Gestión de productos.
- D. Gestión de proveedores.
- E. Gestión de propietarios.
- F. Gestión de historial.
- G. Agenda y eventos.
- H. Facturación.

Para cada caso de uso se distinguen las siguientes categorías:

<u>CÓDIGO</u>	<u>NOMBRE</u>	<u>DESCRIPCIÓN</u>	<u>PRIORIDAD</u>	<u>ROL DE USUARIO</u>
A-1	Crear usuario	Solo podrá añadir usuarios al sistema el usuario administrador de la aplicación.	Muy alta	Administrador
A-2	Eliminar usuario	El administrador será capaz de eliminar usuarios del sistema.	Alta	Administrador
A-3	Recuperar contraseña	Cualquier usuario registrado en el sistema podrá solicitar su contraseña.	Alta	Usuario registrado
A-4	Iniciar sesión	Todo usuario registrado será capaz de iniciar sesión en el sistema mediante su identificador y contraseña.	Muy alta	Usuario registrado
B-1	Crear paciente	Los veterinarios serán capaces de añadir nuevos pacientes (mascotas) al sistema especificando los datos más relevantes.	Alta	Usuario registrado
B-2	Editar paciente	Todo usuario podrá editar todos los datos relacionados con un paciente dado, salvo el DNI de su propietario.	Media	Usuario registrado

B-3	Ver pacientes	Todo usuario registrado podrá acceder al listado completo de pacientes y sus detalles.	Media	Usuario registrado
B-4	Eliminar paciente	El usuario podrá eliminar pacientes registrados en el sistema.	Baja	Usuario registrado
C-1	Añadir productos	Se podrán añadir nuevos productos en almacén, para ello su proveedor debe haber sido registrado con anterioridad.	Alta	Usuario registrado
C-2	Editar productos	Se permitirá editar los datos relacionados con productos ya añadidos, salvo su proveedor.	Media	Usuario registrado
C-3	Ver productos	Todo usuario registrado podrá ver la lista de productos existentes en el almacén.	Alta	Usuario registrado
C-4	Eliminar productos	Eliminación de productos de la base de datos.	Media	Usuario registrado
D-1	Añadir proveedor	Crear nuevos proveedores con sus correspondientes datos que serán nombre, teléfono, correo electrónico y dirección.	Alta	Usuario registrado
D-2	Editar proveedor	Permite editar todos los datos de un proveedor dado.	Media	Usuario registrado
D-3	Ver proveedores	Se podrá consultar el listado de todos los proveedores disponibles así como sus detalles de contacto.	Alta	Usuario registrado
D-4	Eliminar proveedor	Eliminar de la base de datos un proveedor concreto.	Media	Usuario registrado
E-1	Añadir propietario	Añadir nuevos propietarios de mascotas a la base de datos.	Alta	Usuario registrado
E-2	Editar propietario	Editar determinados datos de un propietario.	Media	Usuario registrado
E-3	Ver propietarios	Ver listado de propietarios, datos y sus mascotas relacionadas.	Alta	Usuario registrado

E-4	Eliminar propietario	Eliminar un propietario de la base de datos.	Media	Usuario registrado
F-1	Ver detalles de un caso	Visualizar todos los detalles médicos de casos atendidos en citas	Alta	Usuario registrado
G-1	Crear cita	Crear una nueva cita para una mascota dada.	Muy alta	Usuario registrado
G-2	Realizar cita	Finalizar una cita en transcurso añadiendo los detalles observados y generando la correspondiente factura.	Muy alta	Usuario registrado
G-3	Editar cita	Editar detalles de una cita como el veterinario responsable, día u hora.	Media	Usuario registrado
G-4	Ver citas	Ver listado de citas pendientes y realizadas.	Alta	Usuario registrado
H-1	Ver facturas	Ver listado de facturas generadas con cada cita y su estado (pendiente o cobrada).	Alta	Usuario registrado
H-3	Cambiar estado de factura	Cambiar el estado de una factura de “pendiente” a “cobrada”.	Alta	Usuario registrado

Detalles de casos de uso más relevantes

Añadir usuario

Actores: Usuario registrado.

Precondición: Usuario autenticado.

Post-condición: Usuario nuevo añadido al sistema.

Camino normal:

1. El usuario se dirige al menú de “Nuevo usuario”.
2. Rellena los campos con los datos del nuevo usuario.
3. Completa el proceso accionando el botón “Crear usuario”.
4. La aplicación crea el nuevo usuario y vuelve al formulario por si se desea añadir otro, si no es el caso se acciona el botón “Menú principal”.

Camino alternativo:

- 1.1 Al rellenar los campos inserta un nombre de usuario o DNI ya existente y presiona el botón “Crear usuario”.
 - a) Se retorna al formulario y muestra un mensaje de error.
- 1.2 Deja los campos de Login de usuario y DNI en blanco y presiona el botón “Crear usuario”.
 - a) Se retorna al formulario y muestra un mensaje de error.
- 1.3 Presiona el botón de “Menú principal” sin haber creado un usuario.
 - a) Vuelve al menú principal de la aplicación.
- 1.3 Presiona el botón de “Cerrar sesión” sin haber creado un usuario.
 - a) Cierra la sesión actual en el sistema y redirige a la página de inicio de sesión.

Realizar cita

Actores: Usuario registrado.

Precondición: Cita creada.

Post-condición: Cita realizada y factura correspondiente generada.

Camino normal:

5. El usuario se dirige al menú de citas.
6. Selecciona la cita correspondiente y le da al botón de realizar.
7. Añade los detalles de la consulta (diagnostico, observaciones, etc.)
8. Finaliza la cita generando la correspondiente factura.

Camino alternativo:

1.3 No existen citas pendientes

- a) Se informa al usuario de que no hay citas pendientes

2.1 Añade productos a la consulta.

- a) El usuario introduce una parte del nombre o código del producto y sale una lista con los productos que encajan.
- b) El usuario selecciona el producto que buscaba
- c) El usuario especifica la cantidad que desea añadir de dicho producto
- d) Puede volver al punto 3.1 o seguir al punto 4

Ver detalles de mascotas

Actores: Usuario registrado.

Precondición: Ninguna.

Post-condición: Ninguna.

Camino normal:

1. Se dirige al menú de “Mascotas”.
2. Desde el registro correspondiente en la tabla accede a los datos de la mascota.

Cambiar estado de factura

Actores: Usuario registrado.

Precondición: Encontrarse en una factura pendiente de pago.

Post-condición: Estado de factura cambiado a “cobrado”.

Camino normal:

1. El usuario se dirige al menú de facturas.
2. Se sitúa en la lista de facturas pendientes de pago.
3. Selecciona una factura de la lista.
4. Le da al botón de “Pagar”.

6. Diseño

Las tareas básicas a realizar que se esperan de esta aplicación son la creación, edición y eliminación de los registros existentes para cada apartado. La estructura que se ha seguido para crear todas las secciones es bastante similar entre ellas. Cuando el usuario se dirige a un apartado en concreto como por ejemplo pacientes, se le mostrara el banner de la página, la sección de búsqueda, el botón de añadir que desplegará al pulsarlo el formulario destinado a tal fin y una tabla con un listado de todos los registros existentes, ordenado por número de historia, cada registro tiene asignados varios botones, los comunes son eliminar y editar, que lo redirigirá al formulario de edición de datos, en algunas secciones tendremos botones concretos para por ejemplo ver más detalles.

A continuación se explica cómo funciona cada uno de los apartados y la relación entre estos:

Almacén

En esta sección podremos encontrar un listado de todos los productos registrados en el almacén, el usuario podrá editar sus detalles o eliminar un producto de la lista. GICOREC dispone de proveedores concretos para cada tipo de producto, no es posible añadir un elemento que no disponga de un proveedor ya registrado en el sistema. El usuario podrá añadir un nuevo proveedor a la lista, cuando así lo precise, sin embargo, no podrá eliminar proveedores siempre que existan productos asociados a este en el almacén. Para un producto ya existente en el almacén, ya que este fue obtenido mediante un proveedor concreto, no será posible editar su campo *“Proveedor”*.

El usuario podrá consultar un listado de todos los productos relacionados a un proveedor concreto, para ello tan solo tendrá que introducir su nombre en el apartado *“buscar”*.

Pacientes

Cada paciente o mascota deberá tener asignado un propietario, para crear esta relación se utilizará su número de DNI que será solicitado en el momento de cumplimentar los datos del nuevo paciente, así pues será necesario registrar todos los datos del propietario de forma previa a los de su mascota.

Para cada entrada de paciente, tendremos acciones tales como eliminar y editar, además podremos consultar su ficha, de este modo el usuario visualizará una página con todos los datos referentes a la mascota y los datos de contacto de su propietario, así mismo se nos mostrará una sección donde se podrá consultar su historial de citas.

Propietarios

En esta sección los usuarios podrán agregar y gestionar todos los datos referentes a los propietarios de mascotas que acuden al hospital, así mismo se podrá consultar un listado de todas las mascotas pertenecientes a un mismo propietario, editarlos o eliminarlos. Si eliminamos a un propietario, se borrarán a su vez todas las mascotas que tenga relacionadas.

Casos médicos

En esta sección el usuario o veterinario podrá encontrar de forma más rápida y sintetizada la información referente a todos los casos médicos que habrán sido aportados en cada cita o consulta realizada en el hospital. Se podrá visualizar para cada caso las características más relevantes en cuanto a interés médico se refiere, como por ejemplo, el diagnóstico, los resultados de la exploración física, archivos e imágenes obtenidas de la misma, etc; así como breve información del paciente.

Agenda

El usuario podrá añadir nuevas citas, editarlas o eliminarlas.

Se mostrará un listado con todas las citas ya existentes en el sistema, pendientes o realizadas. En cuanto a las citas pendientes, será desde aquí donde el veterinario responsable deberá concluirla, accediendo al apartado *“realizar cita”* donde podrá añadir todos los detalles necesarios para el historial, archivos relacionados y además, donde podrá generar la factura correspondiente, en el momento de generar la factura, deberá seleccionar productos ya existentes en el almacén, estos se le mostrarán en una lista de productos relacionados en

cuanto comience a escribir su nombre, además se mostrará la cantidad que hay en stock el usuario no deberá añadir en la factura una cantidad superior a la dada. Después de seleccionar un producto y la cantidad, el precio correspondiente se actualizará de forma automática. Una vez finalizado todo el proceso de facturación, se registrarán todos los detalles en las tablas correspondientes y además, será generado un documento PDF con los detalles de la factura asociada a la cita.

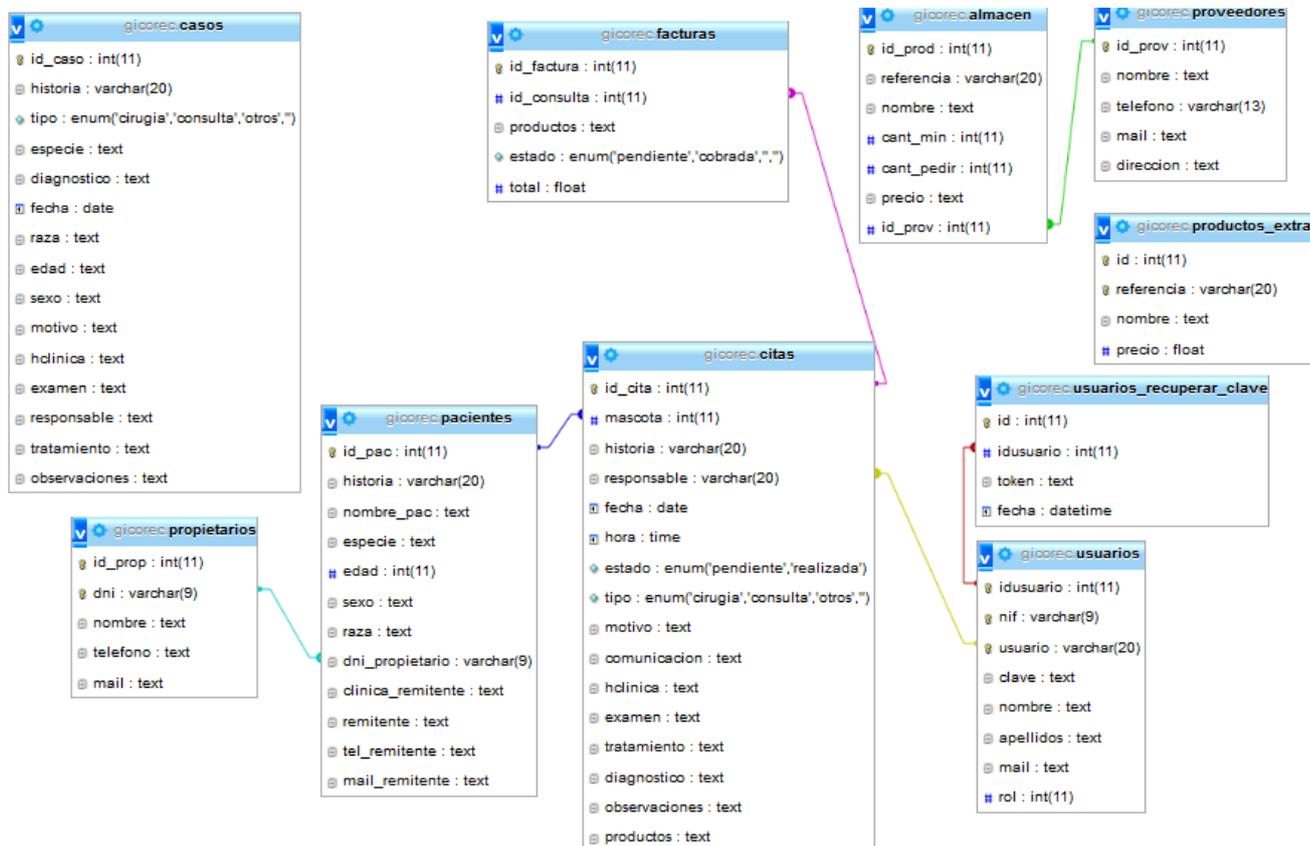
Facturación

Llevará un registro de todas las facturas generadas para los diferentes procesos realizados en el hospital, así como el estado de la factura, que puede ser *“pendiente”* o *“cobrada”* que además podrá ser modificado, en caso de encontrarse pendiente desde la propia lista. Aquí se reflejará el material utilizado para cada procedimiento así como su coste y demás cargos económicos que irán a cuenta del propietario de la mascota. También será posible generar el documento PDF correspondiente desde el botón *“Imprimir”*.

7. Desarrollo

a.) Base de datos

La base de datos de esta aplicación está compuesta por ocho tablas. El esquema entidad relación sería tal y como se muestra a continuación:



A continuación se explica brevemente la relación entre ellas:

- **Propietarios:** su clave primaria consiste en un ID autoincremental, que será única y no nula, tenemos también como clave su DNI, cuyo valor debe ser único y se relacionará con la tabla mascotas, siendo clave foránea de esta. Además se almacenarán datos personales como **nombre** y **apellidos**, **teléfono**, **mail** o correo electrónico.
- **Pacientes:** tendrá como clave primaria un ID autoincremental y como clave foránea **dni_propietario** que será el número del DNI de su propietario, también se usará como forma de identificar a un paciente su número de historial, campo **historia** el cual deberá de ser único. Se deberá almacenar también su nombre, en **nombre_pac**, **edad**, **especie**, **sexo**, **raza**, **clínica_remitente** ya que dado el carácter académico de este

hospital es necesario ya que la gran mayoría de los casos atendidos son remitidos por alguna clínica veterinaria oficial, **remite** nombre del veterinario remitente, **tel_remitente** para su teléfono y su correo electrónico, **mail_remitente**..

- **Almacén:** tendremos también un ID autoincremental que, al igual que en las demás tablas, ayudará a la identificación de registros en las operaciones internas entre la aplicación y la base de datos. El campo **referencia** deberá ser único para cada producto, el **nombre** del producto, **cant_min** o cantidad disponible en el almacén, **cant_pedir** para especificar la cantidad de producto pendiente de pedir a su proveedor, su **precio** y además tendremos también el ID del proveedor asociado a dicho producto, **id_prov**, este se obtendrá de forma automática al especificar su nombre cuando estemos añadiendo un nuevo producto al almacén, de esto se deduce que el proveedor debe existir ya en la base de datos.
- **Proveedores:** tendrá también un ID autoincremental nos permitirá llevar un registro de cada uno de ellos que nos serán necesarios a la hora de añadir un producto en el almacén, como ya se especificó en el apartado anterior. Se almacenará también su **nombre**, **teléfono**, correo electrónico o **mail** y dirección.
- **Productos extra:** Esta tabla contiene información relativa a servicios ofrecidos por el hospital que no tienen relación con los productos pero que igualmente deben de ser facturados como por ejemplo, el precio de una cita, de una analítica o el precio de una cirugía concreta. La información necesaria ya estará cargada por defecto y será mostrada en el momento de realizar la cita mediante un check-box pudiendo seleccionar las que sean necesarias. Los campos de la tabla son un **id** autoincremental y único que actuará como clave primaria, campo **referencia** que también deberá ser único y albergará el código con el que se hace referencia al servicio, el **nombre** del servicio y su **precio**.
- **Usuarios:** llevará el registro de todos los veterinarios o usuarios registrados que estarán destinados a utilizar la aplicación y por consiguiente, susceptibles de atender consultas, razón por la cual estará directamente relacionada con la tabla **citas**. Actualmente solo el administrador de la aplicación puede eliminar registros de usuarios o visualizar el contenido de esta tabla.

- **Usuarios recuperar clave:** tabla auxiliar que usaremos en el momento que un usuario registrado de la aplicación desee restablecer su contraseña, o si es la primera vez que se establece una para un usuario dado. Aquí se almacenará una ristra de caracteres o **token** generada haciendo uso de la función sha1. Está pensada para un sistemas de verificación en el que el usuario introduce su DNI o email, generando el sistema un código (token) que será válido durante veinticuatro horas, enviándose un email al usuario con dicho código para que este reestablezca su contraseña.
- **Citas:** nos permite llevar el registro de todos los eventos de la agenda, relacionando en esta al veterinario encargado o responsable de la misma que además debe estar registrado en el sistema, por lo que deberá estar en la tabla de usuarios, esta relación se hará efectiva mediante el campo **responsable** siendo además clave foránea, con la **mascota** o paciente a quien va dirigida la cita. Se especificará el **motivo** de la misma; aunque no será obligatorio, **fecha, hora**, el (pendiente o realizada), este se modificará una vez se realice la cita rellenando además los campos **comunicación** que tendrá toda la información intercambiada entre el dueño y el veterinario, **hclínica** donde se especificaran los detalles de la historia clínica de cada caso, **examen** con los resultados del examen físico, **tratamiento** a seguir dado el caso, **diagnóstico** observado, **observaciones** adicionales según crea conveniente el veterinario y **productos**, este último campo estará formado por un objeto JSON que contendrá todos los productos que se hayan usado durante el transcurso de la cita, así como su cantidad e importe.
- **Casos:** Esta tabla está destinada a almacenar datos de carácter médico obtenidos de la realización de una cita. Su clave primaria será un ID autoincremental y estará formado por los siguientes campos obtenidos de citas: **historia, tipo, diagnóstico, fecha, responsable, examen, hclinica, tratamiento, observaciones y motivo**; y por los siguientes casos obtenidos de pacientes: **raza, especie, edad y sexo**. Es necesaria ya que se quieren archivar todos los casos atendidos en el hospital.

- **Facturas:** estará relacionada de forma directa con citas, todos los registros de esta serán generados al culminar o realizar una cita, asociando a cada ID de factura generada (**id_factura**), que será autoincremental y clave primaria de la misma, el ID de la cita correspondiente con el campo **id_consulta** que será clave foránea. Tendremos además un campo **productos**, donde se almacenará en JSON la lista de elementos disponibles en el almacén utilizados durante el transcurso de la cita, el **estado** de la factura, campo de tipo *“enum”* cuyos valores pueden ser *“pendiente”* en el caso de que esta no se pague en el momento de ser generada o *“cobrada”* para el caso contrario o para el momento en cuestión en que se haga efectivo el pago de dicha factura.

b.) Clase de apoyo TemplatePower

Esta clase nos permite separar con gran facilidad el código PHP de las vistas o código HTML mediante la creación y uso de plantillas. Este sistema es gratuito y se distribuye bajo los términos de la Licencia pública **GNU**.

Esta clase nos permite extender la funcionalidad base de HTML convirtiendo los ficheros de este tipo en objetos sobre los cuales podemos realizar distintas funciones; utilizando para ello dos abstracciones propias de la clase, que son el concepto de bloque y de variable.

Un bloque es un conjunto de etiquetas o texto HTML encerradas entre dos “etiquetas” que se construyen utilizando comentarios del propio lenguaje, con la siguiente estructura.

```
<!-- START BLOCK : blockname -->  
<!-- END BLOCK : blockname -->
```

La clase hace uso de los bloques como elementos que se pueden crear y/o repetir de manera dinámica, dándonos libertad para construir las vistas mostrando u ocultando información y garantizando que se sigue el mismo formato en el caso, por ejemplo, de las repeticiones.

Las variables son expresiones regulares que siguen la forma **{nombre}** (la expresión regular que define la regla para componer la variable es la siguiente `^{\{.\}?\}$`). Este recurso se puede emplear en cualquier parte del código HTML y nos permite asignarle (utilizando el método apropiado de la clase) cualquier tipo de valor. La ventaja evidente es que nos permite simplificar y extender enormemente la funcionalidad de HTML ya que podemos, por ejemplo, construir modularmente una página definiendo las secciones de la página como variables en las cuales cargamos el tipo de contenido que se desee o nos facilita la construcción de etiquetas o elementos con valores dinámicos, por ejemplo un link o enlace que apunte a un identificador que especificamos de forma dinámica y variable según el momento.

Estas estructuras son muy útiles, entre otros casos, para la creación de tablas. En el caso de querer crear una tabla, tan solo debemos crear el bloque para la fila y situar dentro de la declaración de este las variables correspondientes a los valores que queremos mostrar en la fila, para posteriormente desde el archivo de control php, crear tantos bloques como filas deseemos que tenga la tabla y asignar los valores correspondientes a sus para cada iteración de la fila.

Para utilizar esta clase en primer lugar se debe instanciar un objeto de la misma indicándole el fichero de plantilla HTML a partir del cual vamos a construir el objeto (este fichero o plantilla debe incluir las declaraciones de bloques y variables que deseemos emplear posteriormente). Tras la instanciación debemos llamar al método **prepare()** que deja al objeto listo para operar sobre él. A partir de este punto ya podemos utilizar los métodos propios de la clase para la manipulación o manejo de datos, podemos crear bloques (mediante el método **newBlock('blockname')**) según los necesitemos y asignarle distintos valores a las variables (mediante el método **assign('varname', \$value)**)según el momento o la situación lo requiera. Para el correcto funcionamiento se debe entender que la clase solo permite asignar valores a variables que estén presentes en el bloque actualmente activo, salvo que se utilice el método **assignGlobal('varname', \$value)** que asigna el valor a todas las variables que se correspondan con ese nombre en todo el fichero. La clase nos ofrece el método **gotoBlock('blockname')** para establecer el bloque activo y poder operar sobre las variables ubicadas dentro de dicho bloque. Una vez se ha creado todo el contenido podemos proceder a realizar un volcado directo sobre la pantalla o el canal de salida haciendo uso del método **printToScreen()** o bien obtener en una string todo el contenido HTML del objeto mediante el método **getOutputContent()**, este método es especialmente útil para la construcción modular de la página ya que como se explicó antes, podemos tener toda la estructura, como por ejemplo, el menú principal en una plantilla individual para luego cargarla dinámicamente y con valores personalizados según la página que se esté visualizando.

Resumen de los métodos empleados por la clase:

- **assign:** asigna un valor a una variable.
- **assignGlobal:** asigna un valor a todas las variables con un nombre específico.
- **assignInclude:** dice que archivo o contenido se ha de incluir.
- **getOutputContent:** devuelve una string o cadena con todos los datos de la plantilla.
- **getVarValue:** devuelve el valor de la última instancia de una variable. Esto significa que si tenemos un bloque repetitivo solo se puede recuperar los valores de las variables del último bloque creado.
- **gotoBlock:** hace que un bloque especificado pase a ser el bloque actual.
- **newBlock:** crea un nuevo bloque.
- **prepare:** prepara la plantilla.
- **printToScreen:** muestra por pantalla todos los datos de salida.

- **serializedBase:** dice a la clase que el contenido del archivo es serializable.
- **showUnAssigned:** muestra u oculta variables no asignadas.
- **versión:** devuelve la versión de la plantilla.

La estructura de la aplicación se ha implementado de tal forma que la estructura HTML general (cabecera y contenido) se ha definido en el index, generándose las demás páginas a través de este, añadiendo al contenido, de forma dinámica, los bloques pertinentes, según el caso. Para ello, en cada página se utiliza una querystring cuya estructura determinará cual es, para especificar en el index.php que archivo de control cargar y además en algunos casos, si es necesario realizar una gestión concreta dentro de la misma página se especificará del mismo modo.

Por ejemplo para cargar la página de almacen, la estructura sería la siguiente **/gicorec/index.php?pagina=almacen** declarando en el archivo de control del index la variable **\$pagina** y especificando mediante un “switch” que archivo contiene los parámetros de control necesarios.

Para el caso de archivos de gestión, será declarada la variable **\$gestion** cuyo valor será el que especifique, al igual que para página, el archivo necesario, como por ejemplo, si queremos borrar dentro de facturas una entrada concreta, la estructura sería la siguiente **/gicorec/index.php?pagina=facturas&gestion=facturas_del&id={id_factura}**, donde **id_factura** es una variable en la que obtendremos el id que tiene en la base de datos la entrada en cuestión.

Finalmente, en el archivo de control de cada página, especificaremos que esta se ha de construir a través del index con la siguiente instrucción **\$tplIndex->assign("contenido", \$tplNombrePagina->getOutputContent());**

c.) Clase de apoyo TCPDF

Consiste en una clase de código abierto, destinada a ser usada en programación PHP, que permite la creación de archivos PDF de forma directa. Destaca por su simplicidad y por la capacidad de interpretar código XHTML.

Entre sus muchas características podemos destacar las siguientes:

- No necesita de bibliotecas externas para las funciones básicas.
- Soporte para UTF-8 Unicode.
- Interpretación de HTML.
- Incluye métodos para la creación de cabecera y pie de página.
- Soporta imágenes, colores, enlaces web y compresión de páginas.

El uso de esta clase es bastante sencillo, en primer lugar debemos instanciar el objeto invocando al constructor de la clase (**new TCPDF()**) indicando los parámetros básicos para la estructura del archivo que se va a generar, como son codificación de caracteres y orientación de la página, así pues creamos el documento:

```
$pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT, true, 'UTF-8', false)
```

A continuación debemos establecer la información referente al archivo pdf que vamos a crear, en primer lugar establecemos como lo vamos a crear (**SetCreator(PDF_CREATOR)**), luego especificamos la información relativa al archivo, como puede ser por ejemplo el autor (**SetAuthor('nombre_autor')**), el título (**SetTitle('titulo')**) o el asunto (**SetSubject('asunto')**).

El siguiente paso es definir el contenido de la cabecera (**SetHeaderData()**) así como su fuente (**setHeaderFont()**), los márgenes (**SetMargins()**), el salto de página automático (**SetAutoPageBreak()**), la fuente que se usará para el contenido del documento (**SetFont()**) y por último, añadir la página (**AddPage()**) para luego cerrar y dar salida al fichero PDF creado (**Output()**).

d.) Implementación

El proyecto completo se encuentra almacenado dentro de una carpeta a la que se ha llamado “gicorec” dentro de la misma podemos hallar varios archivos y subcarpetas donde hemos alojado el código y demás recursos necesarios para la aplicación, estructurándolas según sus funciones.

Los ficheros implementados en esta aplicación, siguiendo la estructura que nos facilita el framework del que se hace uso, son los siguientes:

- Doce ficheros html de vistas o plantillas (ubicados en la ruta gicorec/plantillas).
- Catorce clases de modelo (ubicados en gicorec/gestion).
- Catorce controladores (ubicados en la raíz del proyecto), uno de ellos como librería que alberga una serie funciones que han sido definidas, necesitadas por diferentes funcionalidades de la aplicación.
- Tres ficheros con funciones ajax auxiliares (ubicados en gicorec/ajax).
- Ocho ficheros javascript con funciones auxiliares (ubicados en gicorec/js). De entre otros podemos diferenciar dos desarrollados por el usuario que consisten en llamadas a funciones javascript (agenda.js y general.js). Estos nos permiten crear los mensajes de confirmación que son mostrados por ejemplo, al eliminar un registro, el widget de calendario al seleccionar una fecha, el autocompletar, paginar las tablas, etc.
- Cuatro ficheros de estilo CSS, así como una carpeta “media” donde se almacenan todas las imágenes utilizadas para el diseño de la aplicación (ubicados en gicorec/css y gicorec/css/images, respectivamente).

Además de los archivos que se han implementado propios de la aplicación, existen otros elementos de software externos, pertenecientes al framework o clases que hemos utilizado de apoyo como son:

- La clase “class.TemplatePower.inc” necesaria para el desarrollo por el sistema de plantillas (ubicada dentro de gicorec/clases).
- Los ficheros necesarios para la creación de archivos pdf, que se utilizará en la facturación (ubicados dentro de gicorec/clases/ tcpdf_min).
- Librerías específicas de javascript y jquery necesarias para el desarrollo de múltiples funciones que fueron necesarias para la aplicación, como son **jquery-ui-**

1.10.3.custom, necesaria para la creación de tablas y pestañas, **jquery-1.9.1.js** y **jquery.ui.datepicker-es.js**, que nos facilita la selección de fechas permitiéndonos mostrar un pequeño calendario (ubicados en la carpeta gicorec/js).

- Fichero de estilo específico para los elementos de jquery-ui-1.10.3.custom.js, llamado **jquery-ui-1.10.3.custom.min.css** (ubicado dentro de la carpeta gicorec/css).

8. Limitaciones y mejoras futuras

Entre las limitaciones o detalles de la aplicación que podrían ser mejorados, si bien no son muy relevantes con respecto al resto, se encuentran:

- A la hora de reestablecer una contraseña al usuario tan solo se le solicita verificar su número de DNI. Lo ideal, a efectos de aumentar la seguridad, sería disponer de un servidor de correo para en su lugar, enviar a la dirección de correo electrónico del usuario un enlace con un token, generado de forma previa, que le permita reestablecer la contraseña. Garantiza que el usuario de dicho email acceda a una página concreta, a la que solamente el tendrá acceso, para reestablecer su contraseña impidiendo que otro usuario pueda modificarla sin su consentimiento. Esto se ha dejado pendiente de implementar dado que en un principio, el servidor donde se alojará la aplicación no dispone de servidor de correo.
- En la actualidad ninguna cita y factura correspondiente pueden ser eliminadas dado que son necesarios para llevar un registro de casos y de contabilidad. Además de llevar un registro de facturas se debería crear una sección destinada a llevar la contabilidad general, aunque si bien esto sería bastante útil, no era necesaria su implementación inmediata. También se debería establecer un mecanismo para que por ejemplo, todas las citas ya realizadas y las facturas cobradas correspondientes, sean eliminadas de forma automática del sistema después de un cierto tiempo.
- Será de gran utilidad permitir que desde la realización de cada cita el usuario disponga de un gestor que le permita agregar varios archivos a la vez (tantos como desee) o imágenes y hacer referencia a estos desde la entrada correspondiente en la sección “Casos” para que sean visualizados, en este caso solo se ofrece la posibilidad de agregar un solo archivo o varios comprimidos en un .zip que se almacenaran ya descomprimidos dentro de la carpeta de destino para poder ser visualizados en casos.

- Tanto en citas como en facturas la fecha se visualiza en formato "dd-mm-yyyy" pero en la base de datos, dado el tipo de campo es almacenada en el formato "yyyy-mm-dd" teniendo el usuario que, a la hora de realizar una búsqueda, poner la fecha en este formato. Como mejora futura se pretende que, sea cual sea el formato de fecha introducido por el usuario, el sistema lo compruebe y lo transforme al usado por el campo fecha de la base de datos.
- En cuanto a mejoras de diseño se pretende hacer que el estilo sea mucho más flexible para su correcta visualización en pequeñas pantallas como puede ser la de un Smartphone.

9. Requisitos hardware y software

Entre los requisitos necesarios para esta aplicación tenemos por un lado, los del entorno donde se ha desarrollado y los necesarios para su puesta en funcionamiento y uso.

Para el desarrollo de la misma el software necesario fue:

- Sistema operativo, en este caso se usó Windows 7.
- Un servidor HTTP, como por ejemplo EasyPHP-12.1, que incluye MYSQL 5.5.27; necesario para la gestión de bases de datos, Apache 2.4.2 y PHP 5.4.6.
- Un entorno de desarrollo adecuado, como por ejemplo NetBeans.
- Navegador web, preferiblemente Mozilla firefox con el plugin firebug integrado.

Hardware necesario para su desarrollo:

- Un PC con teclado y ratón.

Para la explotación de la aplicación los requisitos software necesarios serán:

- Sistema operativo.
- Servidor HTTP.
- Versión reciente de PHP.
- Soporte para gestión de bases de datos como MYSQL.

El hardware necesario para su explotación será:

- Servidor Web donde alojar la aplicación.

En cuanto al software necesario para su uso se necesitará:

- Sistema operativo, en la Tablet por ejemplo puede ser Android.
- Un navegador web.

Y en cuanto al hardware:

- Un PC o Tablet.

10. Modelo de negocio

Esta aplicación web PHP, ofrece una solución simple y barata a las pequeñas clínicas veterinarias u hospitales.

Tras consultar a varios centros veterinarios sobre los sistemas que usan actualmente y sobre las dificultades que estos les puedan ocasionar, se llegó a la siguiente conclusión: muchos centros veterinarios no disponen de aplicaciones informáticas específicas para la gestión de los datos que les son relevantes o usan aplicaciones muy caras y difíciles de mantener, sobre todo para los usuarios con bajo conocimiento informático.

Así pues, esta aplicación ofrece una solución simple a estos problemas, siendo sencilla de manejar por lo que no es necesario un aprendizaje demasiado exhaustivo por parte del usuario final.

Permite una puesta en marcha prácticamente inmediata, puede ser alojada tanto en un servidor o dominio, como en el propio PC de la consulta. El mantenimiento necesario es mínimo posibilitando además de forma simple, el añadir nuevas funcionalidades según las necesidades del cliente.

El usuario podrá registrar y tener rápido acceso a todos los datos necesarios sin costosas búsquedas o procedimientos para ello.

El presupuesto estimado del proyecto sería:

<u>Elemento</u>	<u>Precio</u>
Desarrollo de la aplicación (300 horas)	200€
Cuota alojamiento en servidor web dedicado	25€/año (alojamiento básico)
Actualizaciones hasta versión definitiva	Gratuito
Soporte de la versión definitiva	Gratuito los tres primeros meses
Precio versión definitiva del producto	200€ (sobre el precio anterior de la aplicación)

11. Bibliografía

- Archivos y documentación de la clase TCPDF.
<http://www.tcpdf.org/docs.php>
- Archivos y documentación de la clase TemplatePower.
<http://templatepower.codocad.com/>
- Documentación de PHP.
<http://www.php.net/>
- Ayuda diversa sobre CSS, HTML, Javascript y PHP.
<http://www.w3schools.com/>
- Archivos y documentación de funciones JQueryui.
<http://api.jqueryui.com/>
- Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.
<https://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf>
- Manual de referencia de Mysql
<http://dev.mysql.com/doc/refman/5.0/es/>

12. ANEXO: Manual de usuario y software

Esta aplicación permite que el veterinario lleve un registro informático de todos sus clientes, tanto pacientes como propietarios asociando a ambos, además facilita la gestión del almacén, la creación de citas y facturas, así como el almacenamiento de datos relevantes a cada consulta.

En primer lugar, para la realización de cualquier actividad el veterinario debe de iniciar sesión mediante su nombre de usuario y contraseña. Si tiene algún problema para recordar esta última dispone de un enlace que le ayudara a reestablecerla.

Una vez identificado en el sistema tendrá acceso al menú principal, compuesto por una serie de botones o secciones, bastante intuitivos y que además permiten su fácil uso y visualización en Tablet. En este punto podrá acceder a la sección que desee o cerrar sesión.



Las secciones de Almacén, Pacientes, Dueños y Agenda, comparten un funcionamiento muy similar, en lo que se refiere a la visualización de la información existente, añadir nueva información, edición y eliminación.

Si entramos a una sección cualquiera de entre las mencionadas lo primero que observamos es el bloque de búsqueda referente a esa sección, el botón de añadir un nuevo

registro; que nos mostrará al pulsarlo el formulario para añadir datos, la lista o tabla con todos los valores existentes en el sistema, relacionados con el apartado que estamos visualizando y las correspondientes accesiones disponibles para cada registro de la tabla, como editar, eliminar o ver detalles según el caso.

Todas las tablas serán paginadas cuando el número de registros por página sea quince o superior.

Referencia	Nombre	Cantidad Mínima	Cantidad A pedir	Proveedor	Precio	Acciones
481555	SUERO RINGER LACTATO 500 ml 10UNID	14	3	BRAUN	1,73	Eliminar Editar

Algunas tablas están estructuradas en pestañas para una mejor organización de la información en los casos en los que, para registros de un mismo tipo, debemos distinguirlos o separarlos según una cualidad concreta, por ejemplo para el caso de facturas tenemos la siguiente vista que nos las separa en dos listas, según hayan sido cobradas o estén pendientes de pago:

Nº de historia	Fecha	Hora	Total	Acciones
113	31-08-2013	18:00	151.73	Detalles Imprimir
113	31-08-2013	18:00	166.93	Detalles Imprimir
115	30-08-2013	12:30	30	Detalles Imprimir

A continuación se detallará cada sección de la aplicación.

1. Inicio de sesión

En primer lugar, para la realización de cualquier actividad el veterinario debe de iniciar sesión mediante su nombre de usuario y contraseña. Si tiene algún problema para recordar esta última dispone de un enlace que le ayudara a reestablecerla.



Usuario

Contraseña

Acceder

[Recuperar la contraseña](#)

2. Añadir nuevo veterinario

Solo los veterinarios registrados en el sistema podrán añadir nuevos usuarios de la aplicación. Sin embargo, en un principio habrá un usuario administrador de la aplicación que tendrá acceso a la bases de datos y que será capaz de añadir al primer usuario.

Si se desea agregar un nuevo usuario, el veterinario que lo haga debe estar autenticado en el sistema, posteriormente desde el menú principal habrá un botón llamado “Añadir usuario” que nos mostrará un formulario solicitándonos el identificador del nuevo usuario, su nombre, sus apellidos y el número de su DNI.



Login de usuario

Nombre

Apellidos

DNI

Crear usuario

Una vez creado el usuario este deberá utilizar el enlace para "identificación de nuevo usuario" en la página principal y seguir los pasos que se describe en esa página para poder utilizar el nuevo usuario

Menú principal **Cerrar sesión**

Posteriormente será el usuario añadido el que, al iniciar sesión y dejando su contraseña en blanco, se encargará de establecer una clave de acceso, para ello, de forma previa el sistema, solicitará comprobar su DNI, si la comprobación ha sido correcta se le solicitará que introduzca el nombre de usuario con el que fue registrado y una clave de su elección.

Una vez completado este proceso el usuario será capaz de identificarse en la aplicación y de acceder a todas sus funciones.

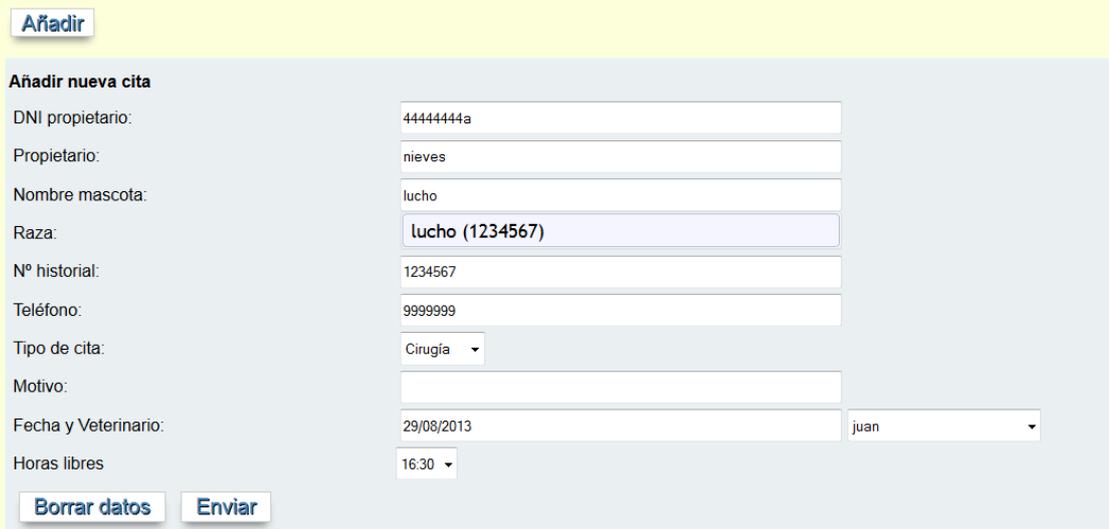
3. Agregar nuevos datos

Para agregar un nuevo registro, ya sea en almacén, pacientes, propietarios o citas, tan solo debemos pulsar el botón de Añadir correspondiente, completar el formulario y enviarlo.

The screenshot shows a web application interface for adding products to a warehouse. At the top, there are two buttons: "Añadir Productos" and "Añadir Proveedores". Below these is a section titled "Insertar productos en el almacén" which contains a form with the following fields: "Nombre:", "Referencia:", "Proveedor:", "Cantidad Mínima:", "Cantidad a pedir:", and "Precio unidad:". The "Precio unidad:" field has a placeholder "###" and a Euro symbol "€". A "Guardar" button is located at the bottom left of the form area.

La única diferencia que podemos encontrar en este caso es en el formulario destinado para añadir una nueva cita, esto se debe a que cada cita estará relacionada con un paciente ya registrado en el sistema y a su respectivo propietario. Así pues introduciendo el nombre del paciente en el campo pertinente, podremos visualizar una lista de todas las ocurrencias existentes para este así como su número de historial; dado que pueden haber varias mascotas con el mismo nombre, para así seleccionar el deseado, una vez hecho esto, se cargará de forma automática la información referente a este y a su propietario, como es la raza del paciente, su número de historia, el número de DNI del propietario, su nombre y su teléfono.

El veterinario tan solo tendrá que introducir de forma manual el tipo de cita (consulta, cirugía u otros), el motivo de la cita (que será opcional), el veterinario correspondiente, la hora y la fecha. Para un veterinario dado, solo se nos mostraran las horas que tenga libres en esa fecha, para evitar dar dos citas a la misma hora. Las franjas horarias entre citas serán de treinta minutos.



The screenshot shows a web form titled "Añadir nueva cita" (Add new appointment) with a yellow header bar containing a blue "Añadir" button. The form fields are as follows:

DNI propietario:	44444444a
Propietario:	nieves
Nombre mascota:	lucho
Raza:	lucho (1234567)
Nº historial:	1234567
Teléfono:	9999999
Tipo de cita:	Cirugía
Motivo:	
Fecha y Veterinario:	29/08/2013 juan
Horas libres:	16:30

At the bottom of the form are two buttons: "Borrar datos" (Clear data) and "Enviar" (Send).

4. Almacén

En la sección "Almacén" se realizarán todas las tareas referentes a productos y proveedores. Dispone de un formulario de búsqueda con el que podremos buscar datos referentes tanto a productos como a proveedores. Los datos susceptibles a una posible búsqueda serán: *referencias* de productos, *nombres*, *cantidades*, *precios* y nombre de un *proveedor* específico que nos mostrará los datos del mismo, así como todos los productos existentes en el almacén surtidos por dicho proveedor.

Bajo el formulario de búsqueda tenemos dos botones de añadir, uno para productos y otro para proveedores.

Se visualizaran dos tablas para ambos separadas por pestañas mostrando toda la información relevante a estos. Para productos tendremos su referencia, nombre proveedor asociado, cantidad existente, cantidad a pedir, precio y los botones de acciones que se podrán realizar para cada entrada.

Para proveedores tendremos su nombre, teléfono, dirección, correo electrónico y acciones.

En ambos casos, las acciones disponibles serán “Editar” un registro y “Eliminar”.

En cuanto a la edición de un elemento, se nos mostrará un formulario con todos los datos referentes a este precargados donde habrá que modificar lo que se desee, en el caso de productos se podrán modificar todos los campos salvo el nombre del proveedor, en caso de que sea necesario modificar este, se hará desde la edición del proveedor correspondiente, donde es posible editar todos sus campos.

5. Pacientes

Aquí encontramos todo lo referente a los datos de las mascotas registradas en el sistema como son su número de historial, nombre, especie, edad, sexo, raza, número de DNI de su propietario, veterinario remitente y acciones. Para este caso entre las acciones disponibles tenemos “Editar”, “Eliminar” y “Ver detalles”. También se podrán realizar búsquedas de registros. Todos los campos de pacientes son susceptibles de búsqueda (nombre, especie, raza, edad, veterinario remitente, etc...), para buscar una mascota perteneciente a un determinado dueño, será necesario introducir su número de DNI.

Los datos mostrados en esta sección estarán agrupados por especies.

6. Dueños

Aquí podremos ver una lista con todos los propietarios de mascotas registrados y añadir nuevos dueños.

Entre los campos mostrados tenemos el nombre del propietario, su número de DNI, el teléfono de contacto, el correo electrónico y una serie de acciones disponibles como, “Editar”; donde podremos editar cualquier dato del propietario desde el formulario de edición que se mostrará al pulsarlo, “Eliminar” y “Mascotas”, en este botón podremos visualizar una lista con todas las mascotas registradas en el sistema para ese propietario y eliminar desde aquí alguna en concreto si es necesario.

Es posible realizar búsquedas introduciendo cualquier dato de los campos mostrados en esta sección.

7. Casos

Sección donde podremos consultar todo lo referente a los detalles de todos los casos clínicos atendidos en el hospital, separados en pestañas según su tipo (cirugías, consultas, otros) mostrándose en detalles todos los datos relevantes de los casos atendidos, así como el listado de archivos agregado en el momento de realizar la cita (en caso de que el veterinario haya subido archivos).

Podemos realizar búsquedas de casos concretos introduciendo para ello valores como el número de historial, el diagnóstico, datos del examen físico y fechas.

NOTA: Aunque la fecha se muestra en formato “dd-mm-aaaa” a la hora de realizar una búsqueda, para que el sistema la reconozca, es necesario introducirla en el siguiente formato “aaaa-mm-dd”.

8. Agenda

En esta sección podremos visualizar todo lo referente a citas, teniendo por separado en dos tablas diferenciadas por pestañas las citas pendientes y las citas ya realizadas.

Para las citas pendientes se nos mostrará en la lista el número de historia del paciente para el cual va dirigida la cita, su nombre, la especie, el nombre del dueño, el teléfono de contacto, la fecha de la cita, la hora y las acciones disponibles.

Como acciones tenemos “Editar”, “Eliminar”, “Detalles” y “Realizar”. Podemos editar, para el caso de una cita que todavía está pendiente de realizar, la fecha, el veterinario responsable y la hora.

En detalles podemos ver de forma breve información adicional sobre la misma, por ejemplo, datos de contacto y en el caso de que haya sido realizada el diagnóstico y las observaciones. Para ver más información médica sobre todos los casos atendidos en el hospital tenemos la sección “Casos”.

Al pulsar el botón “Realizar” estaremos en el paso previo para hacer efectiva una cita, será aquí donde anotaremos todos los detalles que surjan en el transcurso de la misma, en los campos correspondientes, subir archivos (solo uno o varios comprimidos en un .zip). Podremos añadir productos del almacén en el caso de que su uso haya sido necesario, especificando su cantidad y visualizando el precio total al finalizar, posteriormente, al darle al botón de realizar, se generará la entrada en la tabla facturas y se generará una factura en PDF que podremos

descargar.

También podremos realizar búsquedas de citas. Los campos susceptibles a búsquedas son: número de historial, veterinario responsable, tipo de cita, nombre de paciente, nombre del propietario, DNI del propietario y fechas.

NOTA: Aunque la fecha se muestra en formato “dd-mm-aaaa” a la hora de realizar una búsqueda, para que el sistema la reconozca, es necesario introducirla en el siguiente formato “aaaa-mm-dd”.

IMPORTANTE: Los archivos que se suban a la aplicación no deben contener la letra ‘ñ’, tildes o caracteres especiales, es decir, el nombre de este debe de ser alfanumérico.

9. Facturas

Aquí podrán visualizarse todas las entradas de facturas generadas con cada cita finalizada. Tenemos dos tablas separadas por pestañas que diferencian las pendientes de pago de las cobradas. En la tabla se nos mostrará el número de historia del paciente sobre el que se generó la factura, la fecha, la hora, el importe total y las acciones relacionadas a cada registro, en este caso podemos “Imprimir” y “Detalles”. En detalles se mostrará una tabla con los productos adquiridos, la cantidad y el total, para el caso de las facturas pendientes se visualizará el botón “Pagar” para hacer efectivo el cobro de la misma.

En imprimir se generará la factura en formato PDF con los detalles de los productos adquiridos y su importe total.

Se podrán realizar búsquedas, los campos que permiten búsquedas en este caso son: número de historia del paciente para el cual se generó la factura, importe total, DNI del propietario, nombre del veterinario responsable y fecha en la que se generó la factura.

NOTA: Aunque la fecha se muestra en formato “dd-mm-aaaa” a la hora de realizar una búsqueda, para que el sistema la reconozca, es necesario introducirla en el siguiente formato “aaaa-mm-dd”.