



Especificación de requisitos de software

Proyecto: VetTrack

PRIMERA REVISION

Octubre

NOMBRES DE LOS INTEGRANTES

- Mateo González Alonso | NIF: 71473188K
- Víctor Acevedo Hernández | NIF: 02771814S

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
[Fecha]	V1.0	[Descripción]	[Firma o sello]

Documento validado por las partes en fecha: [Fecha]

Por el cliente	Por la empresa suministradora
Fdo. D./ Dña [Nombre]	Fdo. D./Dña [Nombre]



Contenido

FICHA DEL DOCUMENTO	2
CONTENIDO	3
1 INTRODUCCIÓN	5
1.1 Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	5
1.4 Definiciones, acrónimos y abreviaturas	6
1.5 Referencias	6
1.6 Resumen	6
2 DESCRIPCIÓN GENERAL	6
2.1 Perspectiva del producto	6
2.2 Funcionalidad del producto	6
2.3 Características de los usuarios	7
2.4 Restricciones	7
2.5 Suposiciones y dependencias	7
2.6 Evolución previsible del sistema	7
3 REQUISITOS ESPECÍFICOS	7
3.1 Requisitos comunes de los interfaces	9
3.1.1 Interfaces de usuario	9
3.1.2 Interfaces de hardware	9
3.1.3 Interfaces de software	9
3.1.4 Interfaces de comunicación	9
3.2 Requisitos funcionales	9
3.2.1 Requisito funcional 1	9
3.2.2 Requisito funcional 2	9
3.2.3 Requisito funcional 3	10
3.2.4 Requisito funcional n	10
3.3 Requisitos no funcionales	9
3.3.1 Requisitos de rendimiento	11
3.3.2 Seguridad	11
3.3.3 Fiabilidad	11



3.3.4	Disponibilidad	12
3.3.5	Mantenibilidad	12
3.3.6	Portabilidad	12
3.4	Otros requisitos	12
4	APÉNDICES	13



1 Introducción

Se necesita un programa para gestionar una clínica localizada en León que cuenta con (por ahora) una plantilla de 3 personas que son las encargadas de dirigir y gestionar la clínica. Se nos ha pedido que podamos realizar una aplicación de escritorio para poder hacer esto mucho más rápido.

Para ello implementaremos 3 interfaces distintas, una para iniciar sesión, otra para usuarios y otra para los empleados de la aplicación en la que gestionaremos no solo las mascotas de cada persona sino también implementaremos otras funciones como por ejemplo el manejo del inventario, un historial de cada persona en la que aparece todas las acciones en la aplicación y un horario.

1.1 Propósito

Poder gestionar tanto a las personas encargadas de la clínica como a los clientes para poder crear una mayor fluidez y organización del personal, inventario y de las instalaciones.

El propósito de este escrito sería poder dejar todo claro acerca de la aplicación para generar un único camino que tenemos que seguir a la hora de poder programar y entregar la aplicación a la fecha señalizada. Va dirigido a las personas encargadas de programar la aplicación como para los dueños de la clínica. Si cualquier cosa no es de agrado de los dueños siempre se podrá modificar.

1.2 Alcance

Hemos decidido usar el nombre VetTrack por ser un nombre corto y describe la actividad que tiene como objetivo nuestra aplicación. Esto hace que sea más fácilmente reconocible y por tanto que llegue a más audiencia.

El objetivo de esta aplicación es facilitar la organización y el trabajo del personal de la clínica veterinaria

1.3 Personal involucrado

Nombre	Víctor Acevedo Hernández
Rol	Administrador/Desarrollador
Categoría profesional	Ingeniero Informático y desarrollador Web
Responsabilidades	Encargado del back-end
Información de contacto	vacevh00@estudiantes.unileon.es
Aprobación	

Nombre	Mateo González Alonso
Rol	Administrador/Desarrollador
Categoría profesional	Ingeniero Informático y desarrollador Web
Responsabilidades	Encargado del front-end
Información de contacto	mgonza40@estudiantes.unileon.es
Aprobación	



1.4 Definiciones, acrónimos y abreviaturas

Algunos términos que manejamos podrían ser:

- back-end: La parte del servidor, no puede ser visto por el usuario normal
- front-end: La parte que pueden ver el usuario, interfaces, botones, cuadros de texto...
- interfaz: Parte del programa con la que el usuario puede interactuar
- backup: copia de seguridad para no perder progreso

1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
9ª edición	Software engineering	Ruta	2011	Ian Sommerville
2ª edición	Gestión de centros clínicos veterinarios	Ruta	2016	Ignacio Mérida Isla

1.6 Resumen

El contenido del resto del documento recoge las características de la aplicación VetTrack, tanto las características de implementación como la descripción, requisitos y funcionalidades del mismo.

2 Descripción general

2.1 Perspectiva del producto

El producto es un producto independiente, no sería la expansión de uno mayor ya que estaríamos gestionando la aplicación de la clínica entera. A partir de este producto se podrían ir añadiendo otras modificaciones en el futuro que complementen a este. Si tuviésemos que posicionarlo en un punto sería la base ya que hay funciones como la gestión de usuarios que es básico para el funcionamiento de este

2.2 Funcionalidad del producto

Las principales funcionalidades del producto como mencionamos en el anterior apartado serían el correcto funcionamiento de la gestión de usuarios, una correcta implementación de un historial de actividades de esa cuenta y la posible (pero aun por decidir) implementación de una gestión de fechas para poder pedir citas y que un administrador pueda aceptarlas y en caso de rechazarlas poder modificar para cuando podría ser la siguiente. Entre otras funcionalidades mas pequeñas podemos ver la implementación de un sistema de inicio de sesión, una interfaz para los administradores que sea distinta de los usuarios normales (clientes) y una funcionalidad de búsqueda en la que los administradores pueden buscar usuarios no tan solo por su nombre sino también por nombre de sus mascotas. La aplicación a priori podemos ver que manejamos animales de todo tipo (aun por decidir) y un sistema de implementación de inventario en el que aparecen productos con su respectivo precio, en la que no manejamos el patrimonio de cada cuenta, cada usuario tiene que pagar en físico



2.3 Características de los usuarios

Tipo de usuario	Administrador
Formación	Profesional veterinario con conocimiento del programa
Habilidades	Conocimiento de artículos de veterinaria, uso básico de un ordenador
Actividades	Capacidad para gestionar ventas, crear usuarios para los nuevos clientes, crear nuevos perfiles para las mascotas, editar las características de la mascota, generar nuevas citas

Tipo de usuario	Usuario
Formación	No es necesaria formación
Habilidades	Uso básico de un ordenador
Actividades	Ver sus compras, el perfil de su mascota y las citas asociadas a ella

2.4 Restricciones

No se manejarán saldos aun teniendo productos con precio, implementaremos todo en el sistema de lenguaje de java con (aún por decidir) una base de datos. Tendremos varios archivos con distintas clases entre las que se encuentran administrador, cliente, inicio de sesión (main) y llamadas a la base de datos

2.5 Suposiciones y dependencias

Se implementa en Windows, si este se ejecutase en Linux se ejecutaría de una manera completamente distinta

2.6 Evolución previsible del sistema

Un historial de búsqueda de productos, un historial de fechas venideras y pasadas

3 Requisitos específicos

Número de requisito	RE 1
Nombre de requisito	Las mascotas deben estar asociadas a un dueño
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesidad de los dueños de ver los datos de sus mascotas
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RE 2
Nombre de requisito	Los clientes pueden ver su historial de citas y compras
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Seguimiento del cliente
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional



Número de requisito	RE 3
Nombre de requisito	Los administradores pueden dar nuevas citas a los clientes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Organizar el horario laboral
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RE 4
Nombre de requisito	Tener un listado de productos a la venta
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Los clientes y administradores pueden ver los artículos disponibles
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RE 5
Nombre de requisito	Los productos tienen un stock y al gestionar ventas se actualiza el inventario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Gestión automática del inventario
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

Número de requisito	RE 6
Nombre de requisito	Se puede cambiar los datos de la mascota
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Posibles equivocaciones o actualizaciones
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

Número de requisito	RE 7
Nombre de requisito	Se puede eliminar a un cliente de la base de datos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	En caso de que se de de baja o fallezca
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

Número de requisito	RE 8
Nombre de requisito	Los administradores pueden registrar más de una mascota bajo el nombre de un propietario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Los clientes pueden tener más de una mascota
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional



3.1 Requisitos comunes de los interfaces

El administrador tiene la interfaz del cliente, pero con más añadidos

3.1.1 Interfaces de usuario

El usuario va a tener un total de 3 posibles interfaces.

- Una para administrador en el que tendrá botones a mayor de la interfaz de usuario con algunas funcionalidades a mayores para poder modificar/crear/borrar posibles eventos

- Una interfaz de usuario en la que tendrá un par de funcionalidades en la que podrá buscar información, solicitar eventos y (OPCIONAL) adquirir productos

- Una interfaz común de inicio de sesión que tendremos 2 cuadros de texto (uno para contraseñas y otro para el nombre de usuario), no manejaremos el cambio de contraseña por si se le ha olvidado al usuario ya que por ahora esto podría generar un par de problemas con la seguridad. (OPCIONAL) Creación de usuarios (nombre + username + contraseña + animales que tiene + información del animal)

3.1.2 Interfaces de hardware

No se necesita un hardware específico

3.1.3 Interfaces de software

No requiere de otros productos de software

3.1.4 Interfaces de comunicación

En caso de que se implemente la base de datos en la nube, se necesitará acceso a internet, en caso contrario, no es necesario

3.2 Requisitos funcionales

Estos son los requisitos que proponemos para nuestra aplicación:

3.2.1 Requisito funcional 1

Número de requisito	RF1
Nombre de requisito	Interfaz de inicio de sesión
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Se necesita para iniciar sesión en la aplicación y saber que usuario está intentando entrar
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.2.2 Requisito funcional 2

Número de requisito	RF2
Nombre de requisito	Interfaz distinta cliente/administrador
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	La interfaz de administrador tiene que tener más funcionalidades que la de cliente, a parte de las funcionalidades del cliente
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional



3.2.3 Requisito funcional 3

Número de requisito	RF3
Nombre de requisito	Conexión de la aplicación con la base de datos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesitaremos una conexión, ya sea a través de código de la propia aplicación o de terceros para poder manejar los usuarios
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.2.4 Requisito funcional 4

Número de requisito	RF4
Nombre de requisito	Una base de datos en la nube
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Una base de datos que sea externa al host en el que reside la aplicación
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

3.2.5 Requisito funcional 5

Número de requisito	RF5
Nombre de requisito	Lenguaje de programación elegido "Java"
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	El lenguaje que hemos elegido entre todos para la aplicación sería Java
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.2.6 Requisito funcional 6

Número de requisito	RF6
Nombre de requisito	Una conexión entre base de datos usando otro lenguaje de programación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	En caso de usar una base de datos en la nube usaremos otro lenguaje de programación para la conexión entre la aplicación en "Java" y la base de datos en MySQL
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

3.2.7 Requisito funcional 7

Número de requisito	RF7
Nombre de requisito	Botón para registrar clientes en la interfaz de inicio de sesión
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Habrà un botón para poder registrarse en la aplicación en la que se pondrán datos del usuario y datos de las mascotas que tienen
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional



3.2.8 Requisito funcional 8

Número de requisito	RF8
Nombre de requisito	No se podrá suplantar usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	No se podrá suplantar usuarios para poder entrar a su interfaz de usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.2.9 Requisito funcional 9

Número de requisito	RF9
Nombre de requisito	No se podrá crear un usuario sin mascotas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Un administrador no podrá crear un usuario sin mascotas
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

El sistema tendrá que soportar un número considerable de usuarios. La aplicación al ser de escritorio tendrá como límite las capacidades del propio ordenador. Ya que por aplicación iniciada se hará 1 usuario, si se abren 3 aplicaciones tendrá 3 usuarios. Nosotros al no manejar dinero (AUN POR DECIDIR) no tendremos ningún tipo de opción de transacción. Todo tipo de compras se pueden generar desde la aplicación, pero se tendrá que ir a entregar el dinero y recoger el producto desde la propia tienda

3.3.2 Seguridad

Tendremos en la primera interfaz (la interfaz de login) el saneamiento de los cuadros para meter las credenciales (no se permitirán “ , ‘ , paréntesis...) en que primero lo pasaremos el texto a texto plano y haremos un replace de todos estos caracteres

Hashearemos las contraseñas en la base de datos con un “salt” que funcionara de la siguiente manera: “username_password” y todo ira a un md5.

Tendremos un archivo en el que se registraran todas las actividades que se hagan en la aplicación

La aplicación al ser un .class no se podrá modificar el código por lo que no podremos cambiar el código fuente

Tendremos cuidado con el BufferOverflow y los caracteres nulos para evitar posibles ataques de este

(OPCIONAL) Para poder entrar en Administrador tendremos que poner un numero/cadena de caracteres a parte de la contraseña

3.3.3 Fiabilidad

A la hora de tener en cuenta los posibles fallos de aplicación o ataques, de encontrar alguno se avisaría a los desarrolladores para poder solucionarlo lo antes posible, para errores/fallos de la base de datos se crean un backup de usuarios en los que cada semana se haría un backup.

Siempre se va a tener una copia de la aplicación de backup que se ira modificando al mismo tiempo que la aplicación que se usaría de cara al publico



para tener un “respaldo” por si se llegase a realizar un ataque de deformación de la aplicación.

3.3.4 Disponibilidad

La aplicación estará disponible la gran mayoría del tiempo, excepto 1 hora entre las 4-5am para poder reiniciar el dispositivo que contiene la aplicación, para que podamos realizar las actualizaciones (si hubiese alguna del sistema) o posibles modificaciones de alguna parte de la aplicación. Exceptuando cuando se tengan que efectuar cambios en el sistema un poco mas grandes (arreglo de fallos de seguridad, que se quitaría la aplicación de cara al publico para no generar mas problemas a la empresa) y se trabajaría lo antes posible para poder arreglar estos fallos. Si son fallos de bajo y algunos de medio riesgo estos solo se arreglarían en las horas antes mencionadas (4-5 al dia) y si son algunos medios y graves se usaría el plan antes mencionado

3.3.5 Mantenibilidad

Las tareas de mantenimiento o ampliación de la aplicación deben ser llevadas a cabo por un desarrollador, ya que habría que modificar el código fuente de la aplicación.

Lo que si puede hacer el administrador es añadir nuevos clientes a la base de datos ya que se añadirá una función a la interfaz para facilitar este servicio. A priori, no se preeveen mantenimientos totalmente necesarios para la aplicación

3.3.6 Portabilidad

La portabilidad sería en cierto modo compleja ya que habría que desarrollar una nueva interfaz para dispositivos móviles, lo que conllevaría un nuevo desarrollo como mínimo de la interfaz, siendo probable que se tuvieran que usar otras herramientas de desarrollo.

El lenguaje que hemos decidido utilizar es Java, que facilita en cierta manera la portabilidad y además dispone de numerosas librerías y recursos para facilitar la implementación

El programa lo desarrollaremos orientado a Windows, aunque el cambio a Linux tampoco supondría demasiados problemas

3.4 Otros requisitos

Los requisitos podrían variar dependiendo de las leyes de la zona. Podrían estar influenciados a la hora de poner datos, por ejemplo el añadir un campo de datos a la hora de crear un usuario con los animales que sea por ejemplo un seguro específico para el animal, o si se ve influenciado por una cuota a la hora de tener un animal en específico. Otro ejemplo podría ser que si se aprobase una ley por ejemplo para que los perros rescatados de la calle se adoptan estos tendrían un descuento a la hora de comprar comida, poner vacunas...



4 Apéndices

Para empezar pensamos que el usuario no siempre seguirá las instrucciones indicadas o intentara crear problemas para el servidor, por lo que para empezar nos centraremos en crear una interfaz de inicio de sesión segura, en la que no se podrá saltar el inicio de sesión.

Utilizaremos un total de 2 hosts, uno en la nube en la que tendremos la base de datos con todos los usuarios (con contraseña pasada a hash con un salt para que no pongamos la contraseña en peligro en caso de ataque) y otro en la que residirá la aplicación de escritorio. Tendremos que tener conexión de internet para poder utilizar la aplicación.

La aplicación la estaremos modificando con la ayuda de aplicaciones de edición de código y luego la subiremos a aplicaciones de GitHub para poder descargar la aplicación y tener una especie de backup para no perder progreso (vacevh00 y mgonza40)