

Especificación de requisitos de software

Proyecto:

My Vet
Sistema de gestión de Veterinaria

Instrucciones para el uso de este formato

Este formato es una plantilla tipo para documentos de requisitos del software.

Está basado y es conforme con el estándar IEEE Std 830-1998.

Los textos en color azul son indicaciones que deben eliminarse y, en su caso, sustituirse por los contenidos descritos en cada apartado.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad.
25/06	01	Primera Versión del sistema, para ser presentado al cliente Se definen historias de usuarios.	Guadalupe Padin Rojas
06/07	02	Modificación de Historias de usuario Se establecen los criterios de diseño de las distintas interfaces	Guadalupe Padin Rojas
03/08	03	Se actualiza la documentación respaldatoria del sistema (ERS)	Guadalupe Padin Rojas

Contenido

Ficha del documento	3
Contenido	4
1. Introducción	5
1.1. Propósito	5
1.2. Alcance	5
1.3. Personal Involucrado	5
1.4. Definiciones, acrónimos y abreviaturas	5
1.5. Referencias	6
1.6. Resumen	6
2. Descripción general	6
2.1. Perspectiva del producto	7
2.2. Características de los usuarios	7
2.3. Restricciones	8
3. Requisitos específicos	8
3.1. Requerimientos	8
3.2. Product Backlog	9
3.3. Sprints	13
3.3.1. Sprint 0	13
3.3.2. Sprint 01	13
3.3.3. Sprint 02	16

1. Introducción

El presente documento define las Especificación de Requerimientos de Software (ERS) para el Sistema de información My Vet, destinado a la gestión de procesos de veterinaria.

Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

1.1. Propósito

1.2. Alcance

Este documento tiene como propósito brindar una descripción detallada de los requisitos y especificaciones funcionales para la implementación de un sistema de información WEB. Refleja la funcionalidad del producto a desarrollar y los procesos, herramientas, etapas y metodología de trabajo para llevar a cabo el Sistema de información y Tienda Virtual My Vet.

1.3. Personal Involucrado

Nombre	Padin Rojas Guadalupe
Rol	Desarrollador
Categoría	Profesional Estudiante
Responsabilidad	Análisis, Diseño, Testing, Implementación.
Información de contacto	guadapadinrojas@gmail.com

1.4. Definiciones, acrónimos y abreviaturas

<i>Nombre</i>	<i>Descripción</i>
Usuario	Persona que usará el sistema para gestionar procesos
Cliente	Persona que utiliza los servicios de la veterinaria.

HC	Historia Clínica
Paciente	Mascota
ERS	Especificación de Requerimientos de Software

1.5. Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.6. Resumen

Este documento consta de tres secciones.

En la primera sección se efectúa una introducción al mismo y se provee una visión general del alcance.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las primordiales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en grandes detalles.

Por último, la tercera sección del documento es en la que se definen detalladamente los requerimientos que debe satisfacer el sistema.

2. Descripción general

La problemática que existe en las veterinarias al momento de llevar un orden en sus registros de clientes-mascotas y los trabajadores, y más aún, cuando hablamos de llevar las historias clínicas de los pacientes en fichas de forma manual lo que genera un alto volumen de documentación almacenada de forma insegura y sin un rápido acceso a la información. Además de tener en una misma plataforma, una tienda virtual donde el cliente pueda conseguir alimentos, accesorios etc. que pueda requerir su mascota.

Ante esta situación, como solución, surge el sistema My Vet de Gestión Integral para Veterinarias.

Es un sistema desarrollado para centralizar la información de servicios ofrecidos en veterinarias tales como tratamientos médicos, gestión de historias clínicas. Permitirá al cliente acceder a un registro completo con información acerca de vacunación y datos de sus mascotas.

De esta manera, el sistema web, le permite al usuario trabajar de forma rápida, administrando el flujo de información que se origina a la hora de realizar las actividades. Este sistema se diseñó tomando en cuenta las necesidades de los usuarios permitiendo así iniciar sesión y realizar las tareas o procesos que le correspondan.

El programa se desarrollará bajo una plataforma web, que permitirá el acceso a las funciones del Sistema a través del internet mediante un navegador web. El sistema se desarrollará en tres módulos. Durante el primero, el sistema permitirá hacer un login, gestión de usuarios, clientes y mascotas, como así también, gestionar su vacunación, tienda virtual vinculada a Mercado Pago, lo que le permitirá al proyecto alcanzar su producto viable mínimo. El módulo dos, será desarrollado durante los años siguientes de la carrera Tecnicatura en Desarrollo Web y Aplicaciones Digitales.

2.1. Perspectiva del producto

My Vet es un sistema para la gestión de historias clínicas para veterinarias, manteniendo un registro actualizado de los clientes y sus mascotas. Permite llevar un registro de vacunas aplicadas. La automatización de las historias clínicas permitirá al médico veterinario llevar un mayor control y seguimiento de las consultas y evolución de sus pacientes.

Cuenta con una tienda virtual

Será un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma rápida teniendo acceso a la carga de datos en forma inmediata desde cualquier dispositivo mientras el usuario se encuentre fuera de su lugar de trabajo evitando la pérdida de información o la acumulación de trabajo en caso que los datos sean registrados en papel. Además de tener acceso a la información existente en el momento que lo necesite.

2.2. Características de los usuarios

Tipo de usuario	Administrador / Veterinario
Formación	Manejo de herramientas informáticas
Actividades	Control y mantenimiento del sistema. Administración de Usuario, Pacientes, Vacunación y Tienda Virtual. Carga de stock.

Tipo de usuario	Cliente
------------------------	---------

Formación	Manejo de herramientas informáticas, Manejo de entornos web
Actividades	Registrar la información del paciente por primera y única vez. Visualizar la vacunación ya aplicada. Agregar productos al carrito de compras y abonarla a través de Mercado Pago

2.3. Restricciones

El sistema, en principio, podrá ser utilizado solamente por Internet.

- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.
- Lenguajes y tecnologías en uso:

HTML: lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas.

JavaScript: lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

CSS: «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

NextJs:

Bootstrap:

Django:

PostGres/Supabase: sistema de gestión de bases de datos.

3. Requisitos específicos

3.1. Requerimientos

Se dividirán los requerimientos en funcionales, que es lo que el sistema realmente hará,
y no funcionales, que son las restricciones del sistema.

Requerimientos Funcionales:

En esta sección se detallarán las funcionalidades del nuevo sistema, los que serán divididos por los usuarios que lo utilizarán.

Veterinarios

- El sistema debe permitir cargar, listar, eliminar, y modificar los datos de un cliente.
- Debe permitir cargar, listar, eliminar, y modificar los datos del paciente.
- Debe permitir crear, buscar, ver, modificar y eliminar un registro del paciente.

- Debe permitir crear, buscar, ver, modificar y eliminar una aplicación de vacunas.
- El sistema permitirá crear, ver, eliminar y modificar registro de productos
- Debe permitir cargar, listar, buscar, ver, modificar y eliminar usuarios del sistema.
- Debe permitir crear y otorgar permisos a los distintos usuarios según su funcionalidad.

Cientes

- El sistema deberá permitir acceder a la información de sus mascotas.
- El sistema deberá permitir consultar y comprar productos de la tienda.

Requerimientos No Funcionales:

- Debe ser capaz de ser visualizado en los exploradores más importantes: Internet Explorer, Mozilla Firefox y Google Chrome.
- El sistema debe permitir inicio de sesión segura, de tal manera que exista una correcta autenticación.
- Debe indicar el formato correcto de la información que debe ingresar el usuario al sistema en caso de que no corresponda a lo solicitado.

Práctica Profesionalizante

Especificación de requisitos de software

- Cuando exista problema de conectividad el sistema deberá indicarlo mediante un mensaje.

METODOLOGÍA DE DESARROLLO

Se determinó para el desarrollo del proyecto la implementación de un marco de trabajo ágil de desarrollo iterativo e incremental orientado a la gestión de procesos como lo es Scrum, Esta metodología permite la colaboración permanente entre los miembros del equipo. Se divide el proyecto en Sprint de tal manera que se pueden realizar entregas frecuentes y continuas de los módulos terminados, de forma que el cliente pueda disponer de una funcionalidad básica en un tiempo mínimo y a partir de ahí, ir logrando mejoras continuas e incrementales del sistema, revalorizando así, el trabajo en equipo y obteniendo una mayor dinámica en el cumplimiento de los objetivos planteados.

3.2. Product Backlog

El Product Backlog es un listado que contiene todas las tareas que se necesitan realizar para el desarrollo del producto, son las historias de usuarios que conformarán el producto final.

Contiene todos los requerimientos que necesitamos implementar en el producto. Esta lista es el resultado del trabajo del Product Owner con el cliente.

#1 SPA responsive con componentes y servicios, incorporando el sistema de ruteo para la navegación

11 tasks

Páginas:

Home
Registro / Login
Producto / Servicio (PDP)
Listas de Productos / Servicios (PLP)
Perfil / Dashboard Cliente
Información básica
Compras / Turnos
Perfil Dashboard Admin
Información básica
Compras / Turnos
Manejo de usuarios de rol "cliente"

#2 Formularios con su respectiva validación y mensajes personalizados

5 tasks

Front-end
Creación Usuario y Total Clientes
Creación/Edit/Eliminar Mascota y Total Mascotas
Vista Aplicación Vacunas y Total Aplicaciones
Creación/Edit/ Eliminar Aplicaciones de vacunas
Creación/Edit/ Eliminar Producto y Total Productos
Creación/Edit/ Eliminar Tipo Mascota
Creación/Edit/ Eliminar Tipo Producto
Creación/Edit/ Eliminar de Tipo de Vacuna
Contactanos
Login

#3 Data binding

#4 Servicios para separar la lógica de conexión a API Rest (backend) de los componentes

2 tasks

Observables
Manejo de Excepciones

#5 Tablero de control (dashboard) para usuarios cliente

Vista/Edición datos Usuarios

Vista Mascota y sus vacunas aplicadas

#6 Tablero de control administrador (dashboard)

Creación Usuario y Total Clientes
Creación/Edit/Eliminar Mascota y Total Mascotas
Vista Aplicación Vacunas y Total Aplicaciones
Creación/Edit/ Eliminar Aplicaciones de vacunas
Creación/Edit/ Eliminar Producto y Total Productos
Creación/Edit/ Eliminar Tipo Mascota

Creación/Edit/ Eliminar Tipo Producto
Creación/Edit/ Eliminar de Tipo de Vacuna

#7 Script SQL para la creación de la base de datos con los productos/servicios/suscripción iniciales en stock

#8 Usuario del tipo administrador para poder usar Django Admin

#9 Modelos Django necesarios para la aplicación web

5 tasks

Back-end (Modelos)

Pet

Usuario

Producto

Vaccine

Pet Type

Usuario Type

Producto Type

Vaccine Type

#10 Dashboard de Django Admin

4 tasks

Create

Read

Update

Delete

#11 REST API

2 tasks

Endpoints

Servicios

Usuarios

#12 Autenticación y Roles

5 tasks

Autenticación

Login

Logout

Register

Roles

Cliente

Veterinario/Admin

#13 Tienda y Carrito de Compras

Crear el front-end de

Tienda PLP

PDP

Implementar Sandbox de Mercado Pago

#14 Branching

Usaremos un derivado de Conventional Commits que se explica a continuación:

feat: agrega funcionalidades o implementaciones

fix: arregla pequeñas porciones de código de funcionalidades e interfaces

chore: ordena dependencias y/o archivos que no corresponden a una funcionalidad y/o interfaz específica

test: agrega pruebas al código

docs: ordena archivos de documentación

refactor: reorganiza funcionalidades para una optimización u ordenamiento del código

styles: modifica el formateo del código

La estructura será la siguiente:

<scope>/<issue-number><short-description>

Ejemplo:

feat/09-custom-admin-django

#15 Commits

Usaremos un derivado de Conventional Commits que se explica a continuación:

feat: agrega funcionalidades o implementaciones

fix: arregla pequeñas porciones de código de funcionalidades e interfaces

chore: ordena dependencias y/o archivos que no corresponden a una funcionalidad y/o interfaz específica

test: agrega pruebas al código

docs: ordena archivos de documentación

refactor: reorganiza funcionalidades para una optimización u ordenamiento del código

styles: modifica el formateo del código

Notaremos también que vamos a utilizar lenguaje imperativo para darle más carácter al mensaje y, para seguir un estándar, usaremos el idioma inglés.

La estructura será la siguiente:

<scope>: <imperative-short-message>

Ejemplo:

feat: add custom methods for Pet model

#16 Releases

Usaremos semver para los releases. Detallamos más abajo el formato:

SemVer tiene un formato del tipo vX.Y.Z en donde se corresponde lo siguiente:

major (X): agrega funcionalidades que no tienen retrocompatibilidad y cambian la lógica de la aplicación

minor (Y): agrega funcionalidades que tienen retrocompatibilidad y cambian la lógica de la aplicación

patch (Z): agrega fixes a la lógica de negocio que no conllevan ningún agregado de funcionalidad

#17 Documentación

Wiki

IEEE830

PR

DER

Modelo de Datos Relacional

UML de Clases

3.3. Sprints

3.3.1. Sprint 0

N° de sprint Sprint Backlog	Determinar las funciones del sistema. Plantear diseños de base de datos. Plantear páginas. Recuperación de archivos del proyecto de 2021. Elegir librerías y frameworks. Instalar, actualizar etc. Comenzar con la redacción del Informe ieee-830. Crear la empresa, el proyecto y el repository en github Creación de User Stories, Branching y Commits
Responsabilidades	Las actividades las desarrolla la dev Guadalupe Padin Rojas
Calendario	25/06/2024 al 06/07/2024

3.3.2. Sprint 01

El objetivo del segundo sprint es diseñar la interfaz principal como la de inicio que gestiona la distribución de las tareas que realizará el usuario y la de Login para permitir el ingreso al sistema.

El desarrollo del mismo fue planteado teniendo en cuenta los requerimientos planteados para esta segunda etapa.

Tomando como base el desempeño en el sprint 0 y evaluando los puntos a tener en cuenta para corregir errores se determina lo siguiente:

- Redefinir algunas historias de usuarios que permitan la evolución del sprint.
- Plantear proyecto: front y back.
- Crear carpetas, archivos.
- Refactor del código a NextJS y Django.
- Modelos de Django.
- Creación de base de datos en supabase
- Primeras páginas y componentes para el front.
- Formularios.
- Conexión con base de datos.
- APIRest de Django.
- Autenticación Django.

N° de sprint	01
--------------	----

Sprint Backlog

REST API

#11 by GPRNomade was closed yesterday

2 tasks done

@GPRNomade

Script SQL para la creación de la base de datos con los productos/servicios/suscripción iniciales en stock

#7 by GPRNomade was closed yesterday

@GPRNomade

Servicios para separar la lógica de conexión a API Rest (backend) de los componentes

#4 by GPRNomade was closed last week

2 tasks done

@GPRNomade

Data binding

#3 by GPRNomade was closed last week

@GPRNomade

SPA responsive con componentes y servicios, incorporando el sistema de ruteo para la navegación

#1 by GPRNomade was closed last week

11 tasks done

@GPRNomade

Modelos Django necesarios para la aplicación web

#9 by GPRNomade was closed last week

5 tasks done

@GPRNomade

	<p>Dashboard de Django Admin</p> <p>#10 by GPRNomade was closed 3 weeks ago</p> <p>4 tasks</p> <p>@GPRNomade</p> <p>Usuario del tipo administrador para poder usar Django Admin</p> <p>#8 by GPRNomade was closed 3 weeks ago</p> <p>Autenticación y Roles</p> <p>#12 opened 3 weeks ago by GPRNomade</p> <p>5 tasks</p>
Responsabilidades	Las actividades las desarrolla la dev Guadalupe Padin Rojas
Calendario	06/07/2024 al 20/07/2024

3.3.3. Sprint 02

N° de sprint	02
Sprint Backlog	<p>Tablero de control administrador (dashboard)</p> <p>#6 by GPRNomade was closed now</p> <p>4 of 5 tasks</p> <p>@GPRNomade</p> <p>Tablero de control (dashboard) para usuarios cliente</p> <p>#5 by GPRNomade was closed now</p> <p>3 tasks done</p> <p>@GPRNomade</p> <p>Tienda y Carrito de Compras</p> <p>#13 by GPRNomade was closed 15 minutes ago</p> <p>4 tasks</p>

	<p>Formularios con su respectiva validación y mensajes personalizados</p> <p>#2 opened 3 weeks ago by GPRNomade</p> <p>9 of 11 tasks</p>
Responsabilidades	Las actividades las desarrolla la dev Guadalupe Padin Rojas
Calendario	Desde el 20/07/2024 hasta el 06/08/2024