| DUOC UC - Escuela de informática y telecomunicaciones |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: PetShop* |
|  |
| **Revisión*: [02]*** |
| **24/06/2023** |

| Planificación y Especificación de Requisitos según estándares; IEEE 830, ISO9000 y PMI. |
| --- |

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_gjdgxs)

[**Ficha del documento 3**](#_30j0zll)

[**1. Introducción 3**](#_1fob9te)

[1.1.](#_3znysh7) Propósito 3

[1.2.](#_2et92p0) Ámbito del Sistema 4

[1.3.](#_tyjcwt) Definiciones, Acrónimos y Abreviaturas 4

[1.4.](#_3dy6vkm) Referencias 4

[1.5.](#_1t3h5sf) Visión General del Documento 4

[**2.**](#_4d34og8) **Descripción General 5**

[2.1.](#_2s8eyo1) Perspectiva del Producto 5

[2.2.](#_17dp8vu) Funciones del Producto 5

[2.3.](#_3rdcrjn) Características de los Usuarios 5

[2.4.](#_26in1rg) Restricciones 5

[2.5.](#_lnxbz9) Suposiciones y Dependencias 6

[2.6.](#_35nkun2) Requisitos Futuros 6

[**3.**](#_1ksv4uv) **Requisitos Específicos 7**

[3.1](#_44sinio) Requisitos comunes de las interfaces 8

[*3.1.1*](#_2jxsxqh) *Interfaces de usuario 8*

[*3.1.2*](#_z337ya) *Interfaces de hardware 8*

[*3.1.3*](#_3j2qqm3) *Interfaces de software 8*

[*3.1.4*](#_1y810tw) *Interfaces de comunicación 8*

[3.2](#_4i7ojhp) Requisitos funcionales 8

[3.3](#_2xcytpi) Requisitos no funcionales 9

[*3.3.1*](#_1ci93xb) *Requisitos de rendimiento 9*

[*3.3.2*](#_3whwml4) *Seguridad 9*

[*3.3.3*](#_2bn6wsx) *Fiabilidad 10*

[*3.3.4*](#_qsh70q) *Disponibilidad 10*

[*3.3.5*](#_3as4poj) *Mantenibilidad 10*

[*3.3.6*](#_49x2ik5) *Portabilidad 10*

[3.4](#_2p2csry) Otros Requisitos 10

[**4. Propuesta de Planificación 11**](#_147n2zr)

[4.1 Descripción general acerca de la Planificación 11](#_3o7alnk)

[*4.1.2 Definición del Equipo de Trabajo 11*](#_23ckvvd)

[*4.1.3 Definición de Actividades principales del Proyecto 11*](#_ihv636)

[4.2 Plan de Control de Cambio 11](#_32hioqz)

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *27/03/23* | *01* | *Diego Palmeiro*  *Javier Auspont* | *Comienzo y planeación de proyecto “PETSHOP”* |
| *04/04/23* | *02* | *Diego Palmeiro*  *Javier Auspont* | *Creación de vistas, carro de compras, administración de stock.* |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Javier Auspont* | *Jefe de Proyecto, Constructor* |
| *Diego Palmeiro* | *Diseñador, QA* |
|  |  |
|  |  |
|  |  |

# 1. Introducción

En este documento podrá encontrar las especificaciones del sistema de software a desarrollar llamado PetShop así como su propósito, vista general, etc

## 1.1. Propósito

Este documento está enfocado en mostrar todos los ámbitos del sistema, dirigido al equipo de desarrollo y a las personas quienes encargaron su realización.

## 1.2. Ámbito del Sistema

• El sistema llevará de nombre PetShop

• El sistema contará con una plataforma en internet para promover los productos, entregar un servicio de venta en línea, permitirá a los clientes monitorear sus compras, mejorará el acceso a la logística de los productos y generará descuentos o promociones en la página.

• Esto beneficiaria de gran manera a la empresa, debido a que ahora se podrán automatizar los procesos de venta y manejo de stock, siendo este nuevo sistema el que automáticamente hace las gestiones para que los usuarios puedan acceder a sus productos, se espera que en un futuro la gestión necesaria para mantener el negocio sea mínima.

## 1.3. Definiciones, Acrónimos y Abreviaturas

HTML: Lenguaje de marcado de hipertexto (HyperText Markup Language)

CSS: Hojas de estilo en cascada (Cascading Style Sheets)

JS: JavaScript, lenguaje de programación utilizado por los desarrolladores para hacer páginas web interactivas.

BS5: Bootstrap en su versión 5, es una biblioteca de componentes HTML, CSS y JS con la cual se diseña y organiza el sistema creado

Django: framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador.

Python: Lenguaje de programación ampliamente utilizado en aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning

## 1.4. Referencias

En la elaboración de este informe se han utilizado o se ha hecho referencia a los siguientes documentos:

-Documentación de Django: Documentación oficial del framework Django utilizado en el desarrollo del sistema.

- Documentación de Bootstrap: Documentación oficial de la biblioteca de componentes CSS y JS Bootstrap utilizada para el diseño y organización del sistema.

- Documentación de Python: Documentación oficial del lenguaje de programación Python utilizado en el desarrollo del sistema.

- Documentación de JavaScript Documentación de referencia para el lenguaje de programación JavaScript utilizado en el desarrollo de funcionalidades interactivas del sistema.

## 1.5. Visión General del Documento

El documento estará dividido a partir de aquí en 3 grandes secciones siendo estas la descripción general del proyecto, sus requisitos específicos y la propuesta de planificación del proyecto.

# 2. Descripción General

En esta sección se proporciona una visión general del proyecto, se describirán diferentes aspectos que afectan al producto y sus requisitos. A continuación se presentará una propuesta de solución considerando el contexto y las tecnologías solicitadas

## 2.1. Perspectiva del Producto

La aplicación web Petshop se desarrollará para ser implementada como una web de venda de productos de mascotas. El objetivo principal es mejorar la gestión de ventas, finanzas y logística. La aplicación se basará en tecnologías web como Bootstrap, CSS, JavaScript, JQuery, Django y HTML para ofrecer una experiencia interactiva y amigable para los usuarios.

## 2.2. Funciones del Producto

* La página web contará con un sistema de usuarios y permisos, los cuales podrán registrarse y hacer login, siendo lo que ven de la página dependiente de que tipo de usuario sean
* Se contará con un sistema de stock a tiempo real el cual irá acomodándose a la demanda de los usuarios.

## 2.3. Características de los Usuarios

Se espera que adultos dueños de mascotas, estos serán los clientes esperados, se busca encontrar que los clientes sean de un gran rango de edad, un aproximado de 18 a 60 años.

También estarán los usuarios administrativos, gestores de la tienda los cuales estarán encargados de agregar, editar y eliminar productos y agregar o eliminar categorías de estos.

## 2.4. Restricciones

Las restricciones a tener en cuenta en el desarrollo de PetShop podrían ser

* Restricciones de tiempo: Se debe cumplir con los plazos establecidos para el desarrollo y entrega del proyecto.
* Restricciones de presupuesto: Se debe trabajar dentro de los recursos financieros asignados para el proyecto.
* Restricciones técnicas: La aplicación debe ser compatible con los navegadores web modernos y adaptarse a diferentes dispositivos, como computadoras de escritorio y dispositivos móviles.

## 2.5. Suposiciones y Dependencias

Se asume que los usuarios contaran con un acceso a internet constante y dispositivos compatibles con los navegadores web modernos. Además la aplicación se basara en tecnologías como Bootstrap, CSS, JavaScript, JQuery, Django y HTML, por lo que se requerirá conocimiento y experiencia en estas tecnologías par el desarrollo del proyecto

## 2.6. Requisitos Futuros

Algunos requisitos futuros que podrían considerarse para el crecimiento y mejora del producto incluyen:

* Integración de un sistema de pagos en línea para facilitar las transacciones.
* Implementación de funciones de chat en vivo para brindar soporte y atención al cliente en tiempo real.
* Ampliación de la gama de productos ofrecidos, incluyendo nuevos artículos para mascotas.
* Mejoras en la interfaz de usuario y experiencia de navegación para garantizar una experiencia más atractiva y fluida.

# 3. Requisitos Específicos

En esta sección se detalla los requisitos específicos del proyecto PetShop, asegurando que sean legibles y siguiendo los principios mencionados.

## 3.1 Requisitos comunes de las interfaces

### 3.1.1 Interfaces de usuario

La interfaz debe ser fácil de entender, por lo que esta será compuesta por colores amigables y el texto intenta ser lo más claro posible, muchas veces optando por pictogramas para describir ciertas funcionalidades. Además las interfaces deberán ser capaces de adaptarse a diferentes tamaños de pantalla.

### 3.1.2 Interfaces de hardware

Al ser PetShop una aplicación web requerirá principalmente de un servidor, el cual será utilizado para almacenar no tan solo la página sino que también la base de datos.

El usuario tan solo necesitará un dispositivo lo suficientemente potente para poder ejecutar los navegadores web actuales y así poder conectarse con la página.

### 3.1.3 Interfaces de software

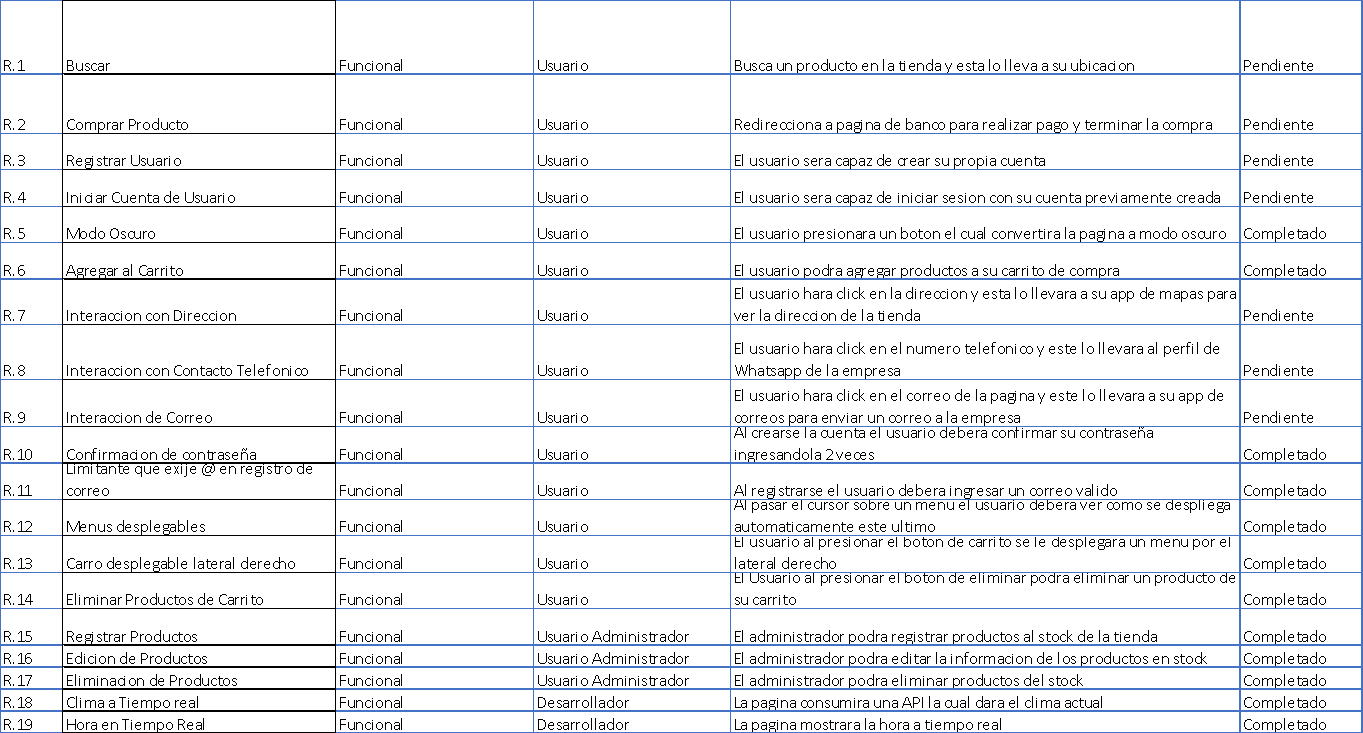
De momento no se a integrado el sistema a ningún otro software pero sí será necesario en un futuro integrarlo a un sistema de pago web para así poder concreta

### 3.1.4 Interfaces de comunicación

Al ser una Web de ventas se debe generar una comunicación con la base de datos de la tienda, esta ya viene incluida en el proyecto, además como ya se mencionó anteriormente, se podrá en un futuro comunicar con el sistema de pagos para así poder concretar finalmente las compras

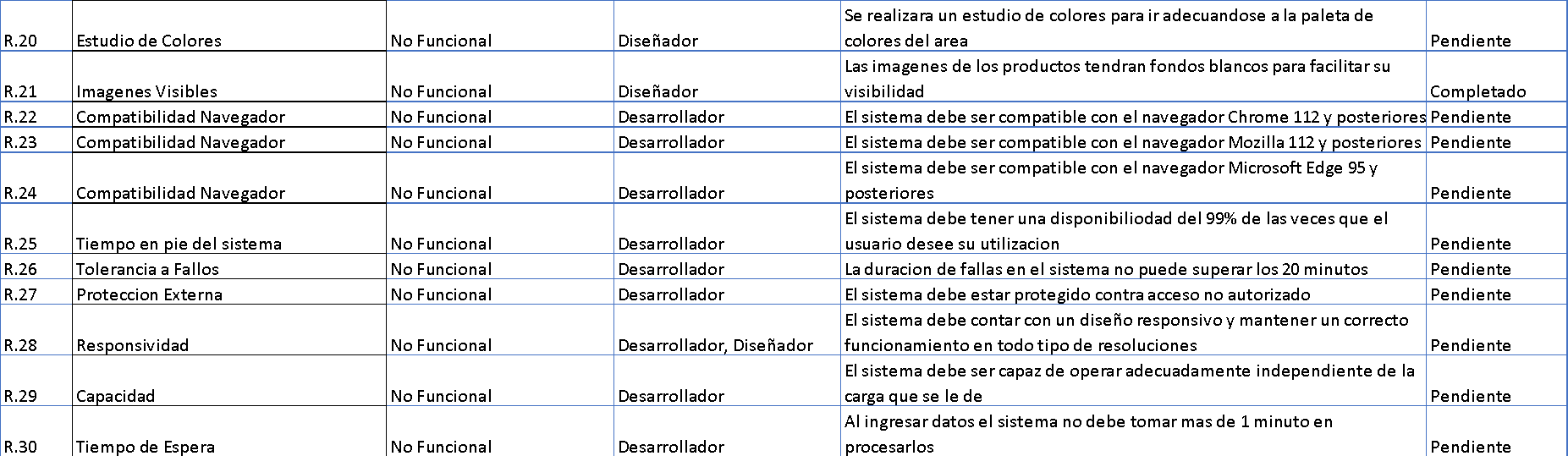
## 3.2 Requisitos funcionales

Los requisitos funcionales del sistema son:



## 3.3 Requisitos no funcionales

Los requisitos no funcionales del sistema son:



### 

### 3.3.1 Requisitos de rendimiento

Los requisitos de rendimiento del sistema son:

* Capacidad de manejar al menos 1000 usuarios simultáneos con un tiempo de respuesta promedio de 2 segundos.
* Procesamiento de al menos 500 transacciones por segundo, con el 95% completado en menos de 1 segundo.
* Gestión de un promedio de 5000 pedidos por día, con capacidad para manejar un aumento del 50% durante promociones especiales.
* Disponibilidad 24/7, con un tiempo de inactividad planificado inferior al 1% del tiempo total.

### 3.3.2 Seguridad

Los requisitos de seguridad del sistema son:

* Empleo de técnicas criptográficas para proteger la información sensible y las comunicaciones.
* Registro de “logs” de actividad para rastrear y auditar las acciones realizadas en el sistema.
* Restricción de comunicación entre módulos, garantizando que solo los componentes autorizados puedan interactuar.
* Comprobación regular de la integridad de la información crítica para detectar modificaciones no autorizadas

### 3.3.3 Fiabilidad

En cuanto a la fiabilidad del sistema, debe ser altamente confiable y estar disponible las 24h del día, los 7 días de la semana, con un tiempo de inactividad máximo permitido de menos de 1 hora al mes. Además se permite un máximo de 5 incidentes de fallos del sistema en un periodo de 6 meses y por último el sistema debe ser capaz de recuperarse automáticamente de fallos y errores, minimizando así el impacto en la disponibilidad y funcionalidad del sitio web de ventas.

### 3.3.4 Disponibilidad

En cuanto a la disponibilidad del sistema este debe estar disponible en una mínima del 99.9% del tiempo. El tiempo de inactividad programado para mantenimiento y actualizaciones no debe exceder el 1% del tiempo total de la disponibilidad requerida y el sistema debe ser capaz de manejar picos de carga sin afectar significativamente la disponibilidad del sitio web de ventas.

### 

### 3.3.5 Mantenibilidad

El mantenimiento del sistema será responsabilidad del equipo de desarrollo. Se deben realizar tareas de mantenimiento periódicamente para garantizar el rendimiento y la estabilidad del sistema. Las tareas de mantenimiento incluyen la corrección de errores, actualizaciones de seguridad, mejoras funcionales y optimización en el rendimiento de la aplicación. Las tareas de mantenimiento siempre deberán llevarse a cabo de manera planificada y coordinada para minimizar interrupciones en el funcionamiento del sistema. Se deben generar estadísticas de acceso de manera regular, como semanal y mensualmente, para evaluar el rendimiento y la usabilidad del sistema. Estas estadísticas servirán como base para realizar ajustes y mejoras en el sistema.

### 3.3.6 Portabilidad

En cuanto a la portabilidad del software, debe ser diseñado de manera que sea fácilmente adaptable a diferentes servidores y entornos de servidor, la dependencia del software a componentes específicos del servidor debe ser mínima. El código del software debe ser modular y estructurado de forma que pueda ser fácilmente portado a diferentes servidores. El lenguaje de programación usado debe ser ampliamente compatible y que permita la portabilidad del software a diferentes plataformas de servidor. Por último el software debe ser compatible con diferentes sistemas operativos comunes, lo que permite su ejecución en diferentes entornos sin requerir modificaciones significativas

## 3.4 Otros Requisitos

Tras un exhaustivo análisis de los requisitos del sistema y considerando el contexto del proyecto de desarrollo del servidor web de ventas, no se han identificado requisitos adicionales que no hayan sido abordados en las secciones anteriores. Todos los aspectos relevantes relacionados con el rendimiento, seguridad, fiabilidad, disponibilidad, mantenibilidad y portabilidad han sido adecuadamente especificados.

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

**Investigación de la demanda de artículos para animales domésticos.**

**Identificación de la competencia en línea y análisis de su oferta.**

**Determinar el público objetivo y sus necesidades.**

**Definición de la propuesta de valor:**

**Identificar los aspectos únicos y diferenciadores de nuestro negocio.**

**Establecer los beneficios que ofreceremos a nuestros clientes.**

**Determinar los valores y principios de la empresa.**

**Selección y adquisición de productos:**

**Realizar un inventario de los diferentes tipos de artículos para animales domésticos.**

**Evaluar la calidad y variedad de los productos disponibles en el mercado.**

**Establecer proveedores confiables y negociar acuerdos comerciales favorables.**

**Diseño y desarrollo del sitio web:**

Contratar a un equipo de diseño web o una agencia especializada.

Definir la estructura y la apariencia del sitio web.

Establecer una plataforma de comercio electrónico segura y fácil de usar.

**Logística y envío:**

Establecer una política de envío eficiente y confiable.

Implementar un sistema de seguimiento de pedidos para los clientes.

Investigar opciones de almacenamiento y gestión de inventario.

**Estrategia de marketing y promoción:**

Desarrollar una estrategia de marketing digital para aumentar la visibilidad del negocio.

Utilizar las redes sociales y el marketing de contenidos para llegar a nuestro público objetivo.

Considerar la posibilidad de colaboraciones con influenciadores y blogs especializados en mascotas.

**Plan financiero:**

Estimar los costos iniciales, incluyendo el diseño web, la adquisición de productos y los gastos de marketing.

Definir un presupuesto mensual para el funcionamiento del negocio.

Establecer metas financieras a corto y largo plazo.

### 4.1.2 Definición del Equipo de Trabajo

Tareas asignadas :

[Javier Auspont ]: Construcción de código, investigaciones de productos.

[Diego Palmeiro]: Investigaciones y consultas sobre mejoras de la página web.

### 4.1.3 Definición de Actividades principales del Proyecto

1- Levantamiento de requerimientos

2- Construcción (HTML, CSS, PYTHON, DJANGO Y APIS)

3- Test de compatibilidad, responsividad y retroalimentación.

4- Entrega final de producto

## 4.2 Plan de Control de Cambio

Identificación del cambio: Se debe establecer un proceso para identificar y documentar todos los cambios propuestos en el proyecto web. Esto puede incluir la creación de un formulario de solicitud de cambio o un sistema de seguimiento de problemas.

Evaluación del cambio: Cada cambio propuesto debe ser evaluado cuidadosamente para determinar su impacto en el proyecto. Esto implica analizar los recursos necesarios, los posibles riesgos y los beneficios esperados del cambio.

Aprobación del cambio: Una vez evaluado, el cambio debe ser aprobado o rechazado por el equipo responsable del proyecto. Esto puede implicar la revisión por parte de los desarrolladores, diseñadores, administradores del proyecto u otras partes interesadas relevantes.

Implementación del cambio: Una vez aprobado, el cambio debe ser implementado en el proyecto web. Esto puede requerir la modificación del código, la actualización de la base de datos, la incorporación de nuevos elementos de diseño, entre otros.

Pruebas y verificación: Después de la implementación, se deben realizar pruebas exhaustivas para verificar que el cambio se haya realizado correctamente y que no haya introducido errores o problemas en el proyecto web.

Documentación y seguimiento: Es importante documentar todos los cambios realizados, incluyendo detalles como la fecha de implementación, el motivo del cambio, las personas involucradas, etc. Además, se debe realizar un seguimiento de los cambios para evaluar su impacto a largo plazo y realizar ajustes si es necesario.

El plan de control de cambios en programación web es fundamental para mantener la integridad y la estabilidad del proyecto a lo largo del tiempo, al tiempo que permite la introducción de mejoras y actualizaciones de manera controlada y organizada