| DUOC UC - Escuela de informática y telecomunicaciones |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: Purrfect Pet Shop* |
|  |
| **Revisión*: [01]*** |
| **[Seleccionar fecha]** |

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

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_heading=h.gjdgxs)

[**Ficha del documento 3**](#_heading=h.30j0zll)

[**1. Introducción 3**](#_heading=h.1fob9te)

[1.1.](#_heading=h.3znysh7) Propósito 3

[1.2.](#_heading=h.2et92p0) Ámbito del Sistema 4

[1.3.](#_heading=h.tyjcwt) Definiciones, Acrónimos y Abreviaturas 4

[1.4.](#_heading=h.3dy6vkm) Referencias 4

[1.5.](#_heading=h.1t3h5sf) Visión General del Documento 4

[**2.**](#_heading=h.4d34og8) **Descripción General 5**

[2.1.](#_heading=h.2s8eyo1) Perspectiva del Producto 5

[2.2.](#_heading=h.17dp8vu) Funciones del Producto 5

[2.3.](#_heading=h.3rdcrjn) Características de los Usuarios 5

[2.4.](#_heading=h.26in1rg) Restricciones 5

[2.5.](#_heading=h.lnxbz9) Suposiciones y Dependencias 6

[2.6.](#_heading=h.35nkun2) Requisitos Futuros 6

[**3.**](#_heading=h.1ksv4uv) **Requisitos Específicos 7**

[3.1](#_heading=h.44sinio) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.2jxsxqh) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.z337ya) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.3j2qqm3) *Interfaces de software 8*

[*3.1.4*](#_heading=h.1y810tw) *Interfaces de comunicación 8*

[3.2](#_heading=h.4i7ojhp) Requisitos funcionales 8

[3.3](#_heading=h.2xcytpi) Requisitos no funcionales 9

[*3.3.1*](#_heading=h.1ci93xb) *Requisitos de rendimiento 9*

[*3.3.2*](#_heading=h.3whwml4) *Seguridad 9*

[*3.3.3*](#_heading=h.2bn6wsx) *Fiabilidad 10*

[*3.3.4*](#_heading=h.qsh70q) *Disponibilidad 10*

[*3.3.5*](#_heading=h.3as4poj) *Mantenibilidad 10*

[*3.3.6*](#_heading=h.1pxezwc) *Portabilidad 10*

[3.4](#_heading=h.49x2ik5) Otros Requisitos 10

[**4. Propuesta de Planificación 11**](#_heading=h.2p2csry)

[4.1 Descripción general acerca de la Planificación 11](#_heading=h.147n2zr)

[*4.1.2 Definición del Equipo de Trabajo 11*](#_heading=h.3o7alnk)

[*4.1.3 Definición de Actividades principales del Proyecto 11*](#_heading=h.23ckvvd)

[4.2 Plan de Control de Cambio 11](#_heading=h.ihv636)

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
|  |  | *GianFranco Soto Munnioz* |  |
|  |  | *Jose Esteban Romero* |  |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Jose Esteban Romero Huentrepan* | *Developer* |
| *Gianfranco Soto Muñoz* | *Developer* |
|  | *Project Manager* |
|  |  |
|  |  |

# 1. Introducción

Este Proyecto Purrfect Pet Shop, Es una página web dedicada a la venta de productos para animales. El Propósito de este es definir de manera clara y precisa los requisitos funcionales y no funcionales del sistema. Se abordarán aspectos como la gestión de cuentas de usuarios, el catálogo de productos, el proceso de compra, la gestión de perfiles y el seguimiento de pedidos. A lo largo de este Informe, se proporcionarán definiciones, referencias y una visión general de la estructura. Esta ERS es esencial para el desarrollo exitoso de Purrfect Pet Shop y garantizar una experiencia de compra en línea satisfactoria y cómoda.

## 1.1. Propósito

El propósito de este informe de ERS es establecer de manera clara los requisitos funcionales y no funcionales del sistema Purrfect Pet Shop. Está dirigido a los equipos de desarrollo, partes interesadas y responsables de la toma de decisiones, brindando una guía para el diseño, desarrollo y pruebas del software. El informe asegura una comunicación efectiva, evaluación del progreso y cumplimiento de los requisitos establecidos. Su objetivo es garantizar el éxito en la implementación del sistema y la satisfacción de las expectativas de los usuarios y objetivos comerciales.

## 1.2. Ámbito del Sistema

El sistema se enfocará en la venta de productos para animales a través de una plataforma en línea. Proporcionará funcionalidades tanto para los clientes como para los administradores, con el objetivo de brindar una experiencia de compra satisfactoria y eficiente.

1- Nombre del sistema:

El sistema se denominará Purrfect Pet Shop.

2- Funcionalidades del sistema:

El sistema permitirá a los clientes explorar los catálogo de productos, agregar artículos al carrito, realizar transacciones seguras y realizar seguimiento de los pedidos. También les permitirá registrarse e iniciar sesión para ser redirigidos a una sección perfil, donde podrán modificar información como nombre, correo electrónico, dirección, y número de teléfono. Además, los clientes podrán suscribirse para recibir beneficios exclusivos, como descuentos y ofertas especiales.

3- Funcionalidades del administrador:

El sistema proporcionará un área de administración para los admin. Desde esta área, podrán gestionar los productos, incluyendo agregar nuevo producto, eliminar o modificar productos (marca, nombre, descripción, precio, stock, animal, categoría, fecha de vencimiento e imagen). También podrán gestionar los envíos, cancelando aquellos que sean necesarios. Los admin tendrán acceso a informes y estadísticas relacionadas con las ventas y los clientes, Lo que les permitirá tomar decisiones informadas para mejorar el rendimiento del sistema.

4- Beneficios, objetivos y metas:

El sistema Purrfect Pet Shop tiene como objetivo proporcionar una experiencia de compra en línea conveniente, segura, cómoda y satisfactoria para los amantes de las mascotas. Se espera lograr beneficios como el aumento de las ventas, la fidelización de los clientes y la mejora de la eficiencia en la gestión de productos y envíos. El sistema tiene como meta ofrecer productos de calidad, servicios personalizados y una plataforma intuitiva y fácil de usar tanto para los clientes como para los admin.

## 1.3. Definiciones, Acrónimos y Abreviaturas

En esta subsección se definirán todos los términos, acrónimos y abreviaturas utilizadas en la ERS.

## 1.4. Referencias

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS.

## 1.5. Visión General del Documento

En esta subsección se describe brevemente los contenidos y la organización del resto de la ERS.

# 2. Descripción General

En esta sección, se proporciona una descripción general del producto y su contexto, sin entrar en los requisitos específicos. Se abordan aspectos como la perspectiva del producto, las funciones del producto, las características de los usuarios, las restricciones, las suposiciones y dependencias, y los requisitos futuros.

## 2.1. Perspectiva del Producto

La perspectiva del producto describe la relación del sistema propuesto con otros productos o sistemas existentes. A continuación, se detallan algunos aspectos relevantes de la perspectiva del producto:

* **Producto independiente:** La aplicación web que se desarrollará es un producto autónomo y no depende directamente de otros sistemas existentes. No obstante, se debe asegurar la compatibilidad y la integración con componentes o servicios externos, como pasarelas de pago, servicios de envío y el servicio REST que simula la suscripción y consulta de la fundación sin fines de lucro.
* **Interfaz con otros sistemas:** Aunque la aplicación web es independiente, se prevé la necesidad de interactuar con otros sistemas o servicios para lograr ciertas funcionalidades. Por ejemplo, la integración con servicios de pago permitirá realizar transacciones seguras y la integración con servicios de envío facilitará el seguimiento del despacho de los productos.
* **Escalabilidad:** Se espera que el sistema pueda crecer y adaptarse a medida que aumente la demanda y se agreguen nuevas funcionalidades. Se debe considerar la capacidad de escalar tanto en términos de usuarios concurrentes como de volumen de transacciones, para asegurar un rendimiento óptimo y una experiencia fluida para los usuarios.
* **Mantenimiento y actualizaciones:** El sistema requerirá un mantenimiento continuo para corregir errores, implementar mejoras y adaptarse a los cambios tecnológicos y las necesidades de la organización. Se debe establecer un plan de mantenimiento y actualizaciones periódicas para garantizar la estabilidad y la funcionalidad a largo plazo.
* **Seguridad:** La aplicación web debe garantizar la seguridad de los datos de los usuarios, así como de las transacciones realizadas a través del sistema. Se deben implementar medidas de seguridad adecuadas, como cifrado de datos, autenticación de usuarios y protección contra posibles vulnerabilidades y ataques cibernéticos.
* **Ciclo de vida del producto:** Se espera que el sistema tenga un ciclo de vida prolongado, lo que implica la posibilidad de realizar mejoras, agregar nuevas funcionalidades y adaptarse a los cambios en el entorno tecnológico y las necesidades del negocio. El ciclo de vida del producto debe ser considerado desde la etapa de diseño y desarrollo inicial, garantizando la flexibilidad y la escalabilidad del sistema.

## 2.2. Funciones del Producto

En esta subsección se presentan de manera resumida las funciones principales que el sistema debe cumplir. Estas funciones representan las capacidades y acciones clave que los usuarios podrán llevar a cabo a través de la aplicación web. A continuación, se detallan las funciones del sistema:

* **Registro de clientes:** El sistema permitirá a los usuarios registrarse como clientes, proporcionando la información necesaria, como nombre, dirección y datos de contacto. Esta función garantizará que los usuarios puedan acceder a los servicios de ventas y realizar compras.
* **Gestión de stock:** El sistema mantendrá un registro actualizado del stock de productos disponibles para la venta. A medida que se realicen compras, se descontarán automáticamente los productos del stock, evitando la venta de productos no disponibles.
* **Realización de compras:** Los clientes podrán elaborar listas de compras, seleccionando los productos deseados y agregándolos al carrito de compras. El sistema calculará el total de la compra, considerando descuentos o promociones aplicables. Luego, los clientes podrán finalizar la compra, proporcionando los datos de envío y realizando el pago correspondiente.
* **Historial de ventas:** Los usuarios registrados tendrán acceso a un historial de ventas, donde podrán visualizar las compras previas realizadas. Esta función brindará a los clientes un seguimiento de sus transacciones y facilitará la gestión de presupuestos y finanzas personales.
* **Subscripción a la fundación sin fines de lucro:** El sistema permitirá a los clientes suscribirse a la donación a la fundación sin fines de lucro. Esta función motivará a los clientes a realizar donaciones recurrentes, ya que se les otorgará un descuento del 5% en el total de su venta. Los datos de suscripción se enviarán directamente a la fundación.
* **Mantenedores de información:** El sistema proporcionará mantenedores para la gestión de información relevante, como clientes, usuarios, productos y promociones o descuentos. Estos mantenedores permitirán la actualización y el mantenimiento de datos importantes para el funcionamiento del sistema.
* **Perfiles de usuario:** El sistema gestionará diferentes perfiles de usuario, con roles y privilegios específicos. Esto permitirá mostrar opciones y funcionalidades de acuerdo con los perfiles de cada usuario, garantizando una experiencia personalizada y adaptada a las necesidades individuales.

En conjunto, estas funciones del sistema permitirán a los usuarios registrados promover y comprar productos para mascotas, realizar un seguimiento de sus compras y presupuestos, mejorar la logística de los despachos, y participar en la suscripción a la fundación sin fines de lucro. Estas capacidades clave brindarán a las amigas la herramienta necesaria para ordenar sus finanzas, mejorar sus ventas y contribuir al trabajo solidario de la fundación.

## 2.3. Características de los Usuarios

En esta subsección se describen las características generales de los usuarios que interactuarán con la aplicación web. Es importante comprender el perfil de los usuarios para diseñar una interfaz y una experiencia de usuario adecuadas. A continuación, se detallan las características de los usuarios:

* **Nivel educacional:** Los usuarios pueden tener diferentes niveles educacionales, desde personas con educación básica hasta aquellas con educación superior. Se debe tener en cuenta esta diversidad al diseñar la interfaz y proporcionar instrucciones claras y comprensibles.
* **Experiencia en tecnología:** Los usuarios pueden tener diferentes niveles de experiencia en el uso de aplicaciones web y tecnología en general. Algunos pueden estar familiarizados con el uso de dispositivos móviles y compras en línea, mientras que otros pueden tener menos experiencia. La interfaz y las funcionalidades deben ser intuitivas y fáciles de usar para usuarios de diferentes niveles de experiencia.
* **Experiencia técnica:** No se requiere experiencia técnica específica para utilizar la aplicación web. Los usuarios no necesitarán conocimientos avanzados de programación o administración de sistemas para acceder a los servicios de ventas y realizar compras. La aplicación se diseñará para ser accesible y amigable para usuarios sin conocimientos técnicos especializados.
* **Intereses en productos** para mascotas: Los usuarios de la aplicación web estarán interesados en productos para mascotas, como bandanas, correas e identificaciones. Se espera que tengan una afinidad por las mascotas y estén dispuestos a realizar compras relacionadas con el cuidado y el bienestar de sus animales.
* **Motivación solidaria:** Los usuarios estarán interesados en contribuir a la fundación sin fines de lucro mediante donaciones. La motivación para suscribirse y participar en la donación será el deseo de ayudar a los animales y apoyar las actividades de la fundación.
* **Diversidad de edades:** Los usuarios pueden abarcar diferentes grupos de edad, desde jóvenes adultos hasta personas mayores. La interfaz y las funcionalidades deben ser accesibles y adecuadas para usuarios de diferentes edades, considerando factores como el tamaño de fuente, la legibilidad y la facilidad de navegación.
* **Preferencia por compras en línea:** Los usuarios estarán interesados en realizar compras en línea y aprovechar los beneficios de la comodidad y la accesibilidad. La aplicación web les proporcionará una plataforma conveniente para buscar productos, elaborar listas de compras y realizar transacciones seguras.

Al comprender estas características de los usuarios, se podrán diseñar interfaces y funcionalidades que se ajusten a sus necesidades y preferencias. La aplicación web se desarrollará teniendo en cuenta la diversidad de los usuarios y proporcionando una experiencia de usuario intuitiva y satisfactoria para todos.

## 2.4. Restricciones

El desarrollo del producto estará sujeto a ciertas restricciones, que incluyen:

* Cumplimiento de las políticas de la empresa.
* Limitaciones del hardware y la infraestructura tecnológica disponible.
* Interfaces con otras aplicaciones y servicios, como el servidor y los servicios REST de la fundación sin fines de lucro.
* Consideraciones de operaciones paralelas, auditoría y control.
* Selección de lenguaje(s) de programación y protocolos de comunicación adecuados, para este caso los lenguajes a utilizar son: HTML, Css, JavaScript y Python, en conjunto con los frameworks de Bootstrap y Django para el front-end y back-end respectivamente .
* Requisitos de habilidades técnicas por parte del equipo de desarrollo.
* Importancia crítica de la aplicación en términos de funcionalidad y seguridad.

## 2.5. Suposiciones y Dependencias

Estas suposiciones y dependencias pueden influir en el diseño y la implementación del sistema, así como en su funcionamiento y mantenimiento continuo. Es importante tener en cuenta estos factores, ya que cambios en estas suposiciones o dependencias pueden requerir modificaciones en los requisitos y en el sistema en general.

Algunas suposiciones y dependencias relevantes incluyen:

* **Organización interna:** Se supone que la estructura y organización interna de la empresa se mantiene estable durante el desarrollo y funcionamiento del sistema. Cualquier cambio en la organización podría afectar los roles y permisos de los usuarios, así como la integración con otros sistemas internos.
* **Sistema operativo y entorno tecnológico:** Se presupone que el sistema operativo y el entorno tecnológico en el que se ejecutará el producto se mantendrán consistentes con las especificaciones y requisitos iniciales. Cambios en el sistema operativo, actualizaciones de software o cambios en la infraestructura subyacente pueden requerir ajustes en la implementación y configuración del sistema.
* **Servicios REST de la fundación sin fines de lucro:** Se asume que los servicios REST proporcionados por la fundación para la validación de suscripciones y la aplicación de descuentos estarán disponibles y funcionando correctamente. Cualquier cambio en la disponibilidad, la estructura de los servicios o la forma en que se comunican podría afectar la funcionalidad del sistema.
* **Integración con otras aplicaciones y servicios:** El sistema puede depender de la integración con otras aplicaciones o servicios, como el servidor de alojamiento web o servicios externos de envío y logística. Se asume que estas integraciones estarán disponibles y seguirán las especificaciones requeridas.

## 2.6. Requisitos Futuros

En esta subsección se presentan posibles mejoras y requisitos adicionales que podrían implementarse en el sistema en futuras etapas de desarrollo. Estos requisitos futuros se identifican para permitir un crecimiento y mejora continua del producto, adaptándose a las necesidades cambiantes de la organización y de los usuarios. A continuación, se enumeran algunas posibles mejoras y requisitos futuros:

* **Integración con redes sociales:** Permitir a los usuarios compartir sus compras y experiencias en redes sociales populares, lo que puede aumentar la visibilidad de la empresa y generar un mayor alcance en la promoción de sus productos.
* **Sistema de calificaciones y comentarios:** Implementar un sistema de calificaciones y comentarios para que los usuarios puedan compartir sus opiniones y experiencias sobre los productos adquiridos. Esto puede ayudar a generar confianza y proporcionar retroalimentación valiosa para mejorar la calidad y la selección de los productos.
* **Sistema de recomendaciones:** Desarrollar un sistema de recomendaciones personalizadas basado en el historial de compras de los usuarios. Esto permitiría ofrecer sugerencias relevantes de productos, promociones y descuentos a los clientes, mejorando así la experiencia de compra y fomentando la fidelidad de los usuarios.
* **Gestión avanzada de inventario:** Mejorar la gestión del inventario mediante la implementación de técnicas avanzadas, como el uso de algoritmos de pronóstico de demanda y análisis de tendencias, con el objetivo de optimizar los niveles de stock y evitar la falta de productos o el exceso de inventario.
* **Aplicación móvil:** Desarrollar una aplicación móvil complementaria que permita a los usuarios acceder y utilizar todas las funcionalidades del sistema desde sus dispositivos móviles. Esto proporcionaría mayor comodidad y accesibilidad, ya que los usuarios podrían realizar compras y realizar un seguimiento de sus pedidos en cualquier momento y lugar.
* **Informes analíticos:** Implementar un módulo de informes y análisis que brinde a los administradores una visión detallada de las métricas de ventas, comportamiento de los clientes, productos más vendidos, entre otros datos relevantes. Esto ayudaría a tomar decisiones informadas y estratégicas para mejorar la gestión y el rendimiento del negocio.

# 3. Requisitos Específicos

La sección de Requisitos Específicos se centra en detallar los aspectos clave que deben considerarse para el desarrollo del proyecto de software. Estos requisitos abarcan desde la implementación de una correcta arquitectura de front-end y back-end, hasta la utilización de un framework de back-end, la generación de un modelo de datos adecuado y la implementación de un servicio REST. Además, se destaca la importancia de manejar y organizar el trabajo grupal utilizando repositorios. A través de estos requisitos, se busca garantizar el desarrollo de una aplicación web funcional y eficiente, que cumpla con los objetivos planteados por las amigas y contribuya al éxito de su emprendimiento solidario.

## 3.1 Requisitos comunes de las interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

### 3.1.1 Interfaces de usuario

La interfaz de usuario (UI) es una parte fundamental de la aplicación web que se desarrollará. A continuación, se detallan los requisitos del interfaz de usuario para el producto:

* **Adaptabilidad a diferentes tamaños de pantalla:** La interfaz debe ser responsive y adaptarse a tres tamaños de pantalla: móvil, tablet y escritorio. Se utilizará una grilla de 12 columnas para lograr una distribución y disposición adecuada de los elementos en la interfaz en cada tamaño de pantalla.
* **Estilo y diseño:** El estilo de la interfaz debe ser atractivo, moderno y acorde al tema de productos para mascotas. Se deben utilizar colores llamativos y amigables relacionados con el mundo de las mascotas, optando por una paleta de colores pasteles entre morado, azulado y rosa. Los elementos de la interfaz, como botones, campos de entrada y listas, deben ser intuitivos y fáciles de usar para los usuarios.
* **Página principal: L**a página principal debe mostrar de manera destacada los productos promocionados y las ofertas vigentes. Se debe incluir un formulario de suscripción a la fundación sin fines de lucro, con información sobre los beneficios de la suscripción.
* **Registro y autenticación de usuarios:** Debe existir un formulario de registro para que los usuarios puedan crear una cuenta. Los usuarios registrados deben tener la posibilidad de iniciar sesión en la aplicación para acceder a funcionalidades adicionales.
* **Catálogo de productos:** Los productos disponibles deben mostrarse en una lista o cuadrícula, con su nombre, imagen, descripción y precio. Se debe indicar claramente el stock real de productos disponibles, para evitar confusiones y permitir una gestión eficiente del inventario.
* **Proceso de compra:** Debe existir un carrito de compras donde los usuarios puedan agregar los productos deseados. Durante el proceso de compra, se deben mostrar los detalles del pedido, incluyendo la cantidad, el precio unitario y el subtotal de cada producto. Se deben aplicar descuentos o promociones automáticamente, dependiendo de las condiciones establecidas.
* **Historial de compras:** Los usuarios registrados deben poder acceder a un historial de sus compras anteriores, incluyendo detalles de cada transacción y estado del despacho.
* **Mantenedores:** Deben existir secciones de administración para gestionar la información de clientes, usuarios, productos y promociones/descuentos. Estos mantenedores deben ser accesibles sólo para usuarios con privilegios administrativos.

### 3.1.2 Interfaces de hardware

La aplicación web que se desarrollará no requiere de una interfaz de hardware específica, ya que se accederá a través de dispositivos como computadoras, tablets o teléfonos móviles. Sin embargo, se pueden mencionar algunas consideraciones relevantes en relación a las características lógicas de las interfaces con los componentes de hardware del sistema:

**Pantallas y dispositivos de visualización:** La aplicación web debe ser compatible con diferentes tamaños de pantalla, desde dispositivos móviles hasta monitores de escritorio. Debe asegurarse de que la interfaz se adapte correctamente a la resolución y proporciones de las pantallas utilizadas por los usuarios.

**Dispositivos de entrada:** La aplicación web debe ser compatible con diferentes dispositivos de entrada, como teclado y mouse para computadoras, así como pantallas táctiles para dispositivos móviles y tablets. Se deben tener en cuenta las interacciones de los usuarios con los elementos de la interfaz, como botones, campos de entrada y opciones de selección.

**Conexiones de red:** La aplicación web debe ser accesible a través de una conexión a internet estable y confiable. Se deben seguir los estándares y protocolos de comunicación web para garantizar una conexión segura y eficiente entre los usuarios y el servidor.

**Configuración del sistema:** Los usuarios deben poder acceder al producto desde los principales navegadores web actuales.

En resumen, la interfaz de hardware se basará en los dispositivos de visualización, dispositivos de entrada y conexiones de red utilizados por los usuarios para acceder a la aplicación web. La configuración del sistema deberá cumplir con los requisitos mínimos establecidos para garantizar un rendimiento adecuado y una experiencia de usuario fluida.

### 3.1.3 Interfaces de software

El producto de software a desarrollar requerirá integración con otros productos de software para su correcto funcionamiento. A continuación, se detallan los productos de software utilizados y los detalles de las interfaces correspondientes:

**Framework de back-end:**

* **Descripción del producto software utilizado:** Se utilizará el framework de back-end Django para facilitar el desarrollo de la lógica de negocio y la interacción con la base de datos.
* **Propósito del interfaz:** El framework de back-end permitirá la implementación de la funcionalidad del sistema, la gestión de datos y la interacción con los usuarios.
* **Definición de la interfaz:** La interfaz con el framework de back-end consistirá en la comunicación entre la aplicación web y las funcionalidades proporcionadas por el framework, como la creación de rutas, el manejo de solicitudes y respuestas HTTP, y la manipulación de la base de datos. El formato de intercambio de datos será JSON.

**Servicio REST de la fundación sin fines de lucro:**

* **Descripción del producto software utilizado:** Se creará un servicio REST que simule pertenecer a la fundación sin fines de lucro y proporcione los siguientes servicios: ingresar suscriptores, eliminar suscriptores y consultar la vigencia de los suscriptores.
* **Propósito del interfaz:** El interfaz con el servicio REST permitirá la comunicación entre la aplicación web y el servicio simulado de la fundación sin fines de lucro. Esto se utilizará para validar la suscripción de los clientes, aplicar descuentos en las ventas y realizar operaciones relacionadas con los suscriptores.
* **Definición del interfaz:** El interfaz con el servicio REST se realizará a través de solicitudes HTTP (GET, POST, DELETE) a las rutas específicas del servicio. Los datos se intercambiarán en formato JSON o XML, según las especificaciones del servicio simulado de la fundación.

### 3.1.4 Interfaces de comunicación

La aplicación web que se desarrollará requiere establecer comunicaciones con otros sistemas, como el servicio REST simulado de la fundación sin fines de lucro. A continuación, se describen los requisitos del interfaz de comunicación y los protocolos utilizados:

**Comunicación con el servicio REST de la fundación sin fines de lucro:**

* **Requisitos de la interfaz de comunicación:** La aplicación web debe poder enviar solicitudes al servicio REST de la fundación para validar la suscripción de los clientes, registrar y eliminar suscriptores, y consultar la vigencia de los suscriptores.
* **Protocolo de comunicación:** La comunicación con el servicio REST se realizará a través del protocolo HTTP (Hypertext Transfer Protocol).
* **Formato de intercambio de datos:** Las solicitudes y respuestas se realizarán en formato JSON (JavaScript Object Notation).

**Comunicación con el framework de back-end:**

* **Requisitos de la interfaz de comunicación:** La aplicación web debe establecer comunicación con el framework de back-end para enviar y recibir datos, gestionar las rutas de la aplicación y realizar operaciones en la base de datos.
* **Protocolo de comunicación:** La comunicación con el framework de back-end se realizará a través del protocolo HTTP.
* **Formato de intercambio de datos:** Las solicitudes y respuestas seguirán el formato JSON.

**Comunicación con la base de datos:**

* **Requisitos de la interfaz de comunicación:** La aplicación web debe interactuar con la base de datos para almacenar y recuperar información, como datos de clientes, productos y transacciones.
* **Protocolo de comunicación:** La comunicación con la base de datos se realizará utilizando el protocolo adecuado para la gestión de la base de datos seleccionada, en este caso, SQLite3.
* **Formato de intercambio de datos:** Los datos se almacenarán y recuperarán en un formato compatible con la base de datos utilizada, como registros en tablas o documentos en colecciones.

## 3.2 Requisitos funcionales

**3.2.1 Autenticar Usuario al iniciar Sesión**

* **Comprobación de validez de las entradas:** Verificar la autenticidad de las credenciales de inicio de sesión proporcionadas por el usuario.
* **Secuencia exacta de operaciones**: Validar las credenciales ingresadas, consultar la base de datos de usuarios, y permitir el acceso si las credenciales son válidas.
* **Respuesta a situaciones anormales:** Manejar situaciones como credenciales incorrectas, cuentas bloqueadas o errores de comunicación con la base de datos.
* **Parámetros:** Nombre de usuario y contraseña.
* **Generación de salidas:** Permitir el acceso al sistema o mostrar un mensaje de error en caso de autenticación fallida.
* **Relaciones entre entradas y salidas:** La entrada es el nombre de usuario y la contraseña, y la salida es el acceso al sistema o un mensaje de error según la validez de estos.

**3.2.2 Registrar Usuario al Sistema**

* **Comprobación de validez de las entradas:** Validar los datos ingresados por el usuario durante el registro, los cuales serán: nombre, dirección de correo electrónico, nombre de usuario, contraseña, dirección y teléfono.
* **Secuencia exacta de operaciones:** Recolectar los datos del usuario, crear un nuevo registro en la base de datos y generar un identificador único.
* **Respuesta a situaciones anormales:** Manejar situaciones como datos incompletos, dirección de correo electrónico inválida o fallas en la comunicación con la base de datos.
* **Parámetros:** Datos del usuario (nombre, dirección de correo electrónico, nombre de usuario, contraseña, dirección y teléfono).
* **Generación de salidas:** Confirmación del registro exitoso o mensaje de error en caso de registro fallido.
* **Relaciones entre entradas y salidas:** La entrada son los datos del usuario y la salida es la confirmación del registro.

**3.2.3 Búsqueda de Productos**

* **Comprobación de validez de las entradas:** Validar los criterios de búsqueda ingresados por el usuario, como nombre del producto, categoría, etc.
* **Secuencia exacta de operaciones:** Realizar una consulta en la base de datos utilizando los criterios de búsqueda proporcionados.
* **Respuesta a situaciones anormales:** Manejar situaciones como criterios de búsqueda no válidos o errores en la consulta a la base de datos.
* **Parámetros:** Criterios de búsqueda (nombre del producto, categoría, etc.).
* **Generación de salidas:** Mostrar los resultados de la búsqueda al usuario.
* **Relaciones entre entradas y salidas:** La entrada son los criterios de búsqueda y la salida son los resultados de la búsqueda.

**3.2.4 Catálogo de Productos**

* **Comprobación de validez de las entradas:** No aplica en este requisito.
* **Secuencia exacta de operaciones:** Obtener los productos almacenados en la base de datos y presentarlos en una lista o cuadrícula en la interfaz de usuario.
* **Respuesta a situaciones anormales:** Manejar situaciones como falta de productos en la base de datos o errores de comunicación.
* **Parámetros:** No aplica en este requisito.
* **Generación de salidas:** Mostrar el catálogo de productos al usuario.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.5 Carrito de Compras**

* **Comprobación de validez de las entradas:** Validar los productos seleccionados por el usuario y su disponibilidad en el inventario.
* **Secuencia exacta de operaciones:** Agregar los productos seleccionados al carrito y actualizar el stock en la base de datos.
* **Respuesta a situaciones anormales:** Manejar situaciones como productos agotados, cambios en el stock durante el proceso de compra, etc.
* **Parámetros:** Productos seleccionados por el usuario.
* **Generación de salidas:** Actualizar el contenido del carrito y mostrar el total de la compra al usuario.
* **Relaciones entre entradas y salidas:** La entrada son los productos seleccionados y la salida es el carrito actualizado y el total de la compra.

**3.2.6 Pago y Checkout**

* **Comprobación de validez de las entradas:** Validar la información proporcionada por el usuario para realizar el pago, como detalles de la tarjeta de crédito o cuenta bancaria.
* **Secuencia exacta de operaciones:** Procesar el pago utilizando la pasarela de pago seleccionada, registrar la transacción y generar un recibo o confirmación de la compra.
* **Respuesta a situaciones anormales:** Manejar situaciones como pagos rechazados, errores en la comunicación con la pasarela de pago, etc.
* **Parámetros:** Detalles de pago proporcionados por el usuario (tarjeta de crédito, cuenta bancaria, etc.).
* **Generación de salidas:** Generar un recibo o confirmación de la compra, actualizar el estado de la transacción en la base de datos.
* **Relaciones entre entradas y salidas:** La entrada son los detalles de pago y la salida es el recibo o confirmación de la compra.

**3.2.7 Historial de Pedidos**

* **Comprobación de validez de las entradas:** No aplica en este requisito.
* **Secuencia exacta de operaciones:** Consultar la base de datos para obtener el historial de pedidos asociado al usuario actual.
* **Respuesta a situaciones anormales:** Manejar situaciones como falta de historial de pedidos o errores en la consulta a la base de datos.
* **Parámetros:** No aplica en este requisito.
* **Generación de salidas:** Mostrar el historial de pedidos al usuario.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.8 Gestión de Inventario**

* **Comprobación de validez de las entradas:** No aplica en este requisito.
* **Secuencia exacta de operaciones:** Actualizar el stock de productos en la base de datos al agregar o finalizar una compra.
* **Respuesta a situaciones anormales:** Manejar situaciones como productos agotados, errores en la actualización del stock, etc.
* **Parámetros:** Cantidad de productos vendidos o agregados al inventario.
* **Generación de salidas:** Actualizar el stock de productos en la base de datos.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.9 Gestión de Envíos**

* **Comprobación de validez de las entradas:** No aplica en este requisito.
* **Secuencia exacta de operaciones:** Registrar y rastrear los envíos asociados a las compras realizadas.
* **Respuesta a situaciones anormales:** Manejar situaciones como errores en el registro de envíos o problemas en la comunicación con el servicio de envío.
* **Parámetros:** Información de envío (dirección de entrega, número de seguimiento, etc.).
* **Generación de salidas:** Mostrar el estado del envío al usuario, actualizar la información de envío en la base de datos.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.10 Validación de formularios**

* **Comprobación de validez de las entradas:** Verificar que los campos obligatorios estén completos y que los datos ingresados cumplan con el formato requerido.
* **Secuencia exacta de operaciones:** Validar los campos del formulario antes de procesar la información.
* **Respuesta a situaciones anormales:** Notificar al usuario los errores de validación y proporcionar mensajes descriptivos para corregir los campos incorrectos.
* **Parámetros:** Datos ingresados en el formulario.
* **Generación de salidas:** No aplica en este requisito.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.11 Reloj en vivo**

* **Comprobación de validez de las entradas:** No aplica en este requisito.
* **Secuencia exacta de operaciones:** Mostrar la hora actual en tiempo real.
* **Respuesta a situaciones anormales:** No aplica en este requisito.
* **Parámetros:** No aplica en este requisito.
* **Generación de salidas:** Mostrar la hora actual en la interfaz de usuario.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.12 API**

* **Comprobación de validez de las entradas:** No aplica en este requisito.
* **Secuencia exacta de operaciones:** Integrarse con la API proporcionada para obtener o enviar información específica.
* **Respuesta a situaciones anormales:** Manejar situaciones de error o falla en la comunicación con la API.
* **Parámetros:** Parámetros requeridos por la API para realizar las operaciones solicitadas.
* **Generación de salidas:** Obtener o enviar información a través de la API.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.13 Categorización**

* **Comprobación de validez de las entradas:** Verificar que se seleccionen categorías válidas.
* **Secuencia exacta de operaciones:** Asignar productos a categorías específicas según las selecciones realizadas por el usuario.
* **Respuesta a situaciones anormales:** Notificar al usuario si se produce un error al asignar categorías o si la categoría seleccionada no es válida.
* **Parámetros:** Categorías seleccionadas por el usuario.
* **Generación de salidas:** Asignar productos a categorías.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.14 Perfil de usuario**

* **Comprobación de validez de las entradas:** Verificar que los campos obligatorios estén completos y que los datos ingresados cumplan con el formato requerido.
* **Secuencia exacta de operaciones:** Permitir al usuario ingresar, modificar y guardar información personal en su perfil.
* **Respuesta a situaciones anormales:** Notificar al usuario si se produce un error al guardar los cambios en el perfil.
* **Parámetros:** Datos ingresados en el perfil de usuario.
* **Generación de salidas:** No aplica en este requisito.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.15 Suscripciones**

* **Comprobación de validez de las entradas:** Verificar que el usuario haya seleccionado una opción válida para la suscripción.
* **Secuencia exacta de operaciones:** Permitir al usuario suscribirse o cancelar la suscripción a la fundación sin fines de lucro.
* **Respuesta a situaciones anormales:** Notificar al usuario si se produce un error al realizar la suscripción o cancelarla.
* **Parámetros:** Opción seleccionada por el usuario para la suscripción.
* **Generación de salidas:** No aplica en este requisito.
* **Relaciones entre entradas y salidas:** No aplica en este requisito.

**3.2.16 Calcular total de la compra**

* **Comprobación de validez de las entradas:** Verificar que los productos y cantidades seleccionados sean válidos.
* **Secuencia exacta de operaciones:** Calcular el total de la compra en base a los precios de los productos y las cantidades seleccionadas.
* **Respuesta a situaciones anormales:** No aplica en este requisito.
* **Parámetros:** Productos seleccionados y cantidades.
* **Generación de salidas:** Mostrar el total de la compra al usuario.
* **Relaciones entre entradas y salidas:** El total es la suma de todos los productos seleccionados.

**3.2.17 Cambio de moneda**

* **Comprobación de validez de las entradas:** Verificar que la moneda seleccionada sea válida.
* **Secuencia exacta de operaciones:** Convertir el precio de los productos a la moneda seleccionada por el usuario.
* **Respuesta a situaciones anormales:** Notificar al usuario si se produce un error al realizar la conversión de moneda.
* **Parámetros:** Moneda seleccionada por el usuario.
* **Generación de salidas:** Mostrar los precios de los productos convertidos a la moneda seleccionada.
* **Relaciones entre entradas y salidas:** Los precios dependen de la moneda seleccionada.

## 3.3 Requisitos no funcionales

### 3.3.1 Requisitos de rendimiento

La subsección de Requisitos de rendimiento se refiere a la especificación de los requisitos relacionados con la carga que se espera que el sistema sea capaz de soportar. A continuación se detallan los requisitos de rendimiento para el sistema:

* **Capacidad de usuarios:** El sistema deberá ser capaz de manejar al menos 100 usuarios simultáneamente conectados. Se espera que el sistema pueda escalar para soportar un crecimiento futuro de usuarios.
* **Tiempo de respuesta:** El 95% de las transacciones realizadas por los usuarios deben completarse en menos de 2 segundos. El sistema debe responder a las solicitudes de los usuarios de manera rápida y eficiente.
* **Carga de transacciones:** El sistema debe ser capaz de manejar al menos 500 transacciones por segundo durante períodos de alta demanda. Se espera que el sistema pueda escalar su capacidad de transacciones para adaptarse a aumentos en la carga.
* **Disponibilidad del sistema:** El sistema debe estar disponible para su uso el 99% del tiempo. Se debe garantizar un tiempo mínimo de inactividad planificado para tareas de mantenimiento y actualizaciones.
* **Almacenamiento y acceso a la base de datos:** El sistema debe ser capaz de almacenar y gestionar un volumen mínimo de 10.000 registros de productos y usuarios. El acceso a la base de datos debe ser rápido y eficiente, permitiendo consultas y actualizaciones de manera ágil.
* **Tolerancia a fallos:** El sistema debe ser capaz de recuperarse automáticamente de fallos y errores sin pérdida de datos o interrupción del servicio. Se deben implementar medidas de seguridad y respaldo para proteger la integridad de los datos.

### 3.3.2 Seguridad

La subsección de Seguridad tiene como objetivo especificar los elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. A continuación se detallan los requisitos de seguridad para el sistema:

* **Empleo de técnicas criptográficas:** Se deben emplear técnicas criptográficas para proteger la confidencialidad y la integridad de la información sensible, como contraseñas, datos de usuarios y transacciones. Se utilizarán algoritmos de cifrado seguros y estándares reconocidos para proteger la comunicación y el almacenamiento de datos.
* **Registro de ficheros con "logs" de actividad:** El sistema debe llevar un registro detallado de la actividad realizada por los usuarios y el propio sistema. Los registros de actividad ("logs") deben incluir información relevante como fechas, horas, usuarios involucrados, acciones realizadas y resultados obtenidos. Los registros de actividad facilitarán la auditoría y la detección de posibles amenazas o actividades sospechosas.
* **Asignación de determinadas funcionalidades a determinados módulos:** Se deben asignar y restringir los permisos de acceso a determinadas funcionalidades del sistema según los roles y privilegios de los usuarios. Los usuarios solo deben tener acceso a las funcionalidades necesarias para realizar sus tareas, evitando así posibles riesgos de acceso no autorizado.
* **Restricciones de comunicación entre determinados módulos:** Se deben establecer restricciones de comunicación entre los diferentes módulos del sistema para prevenir accesos no autorizados o interacciones inapropiadas. Se deben implementar mecanismos de control de acceso y autenticación para garantizar que solo los módulos autorizados puedan comunicarse entre sí.
* **Comprobaciones de integridad de información crítica:** El sistema debe realizar comprobaciones de integridad de la información crítica para detectar cualquier intento de modificación no autorizada. Se deben implementar técnicas de detección de cambios no autorizados en los datos y sistemas, como firmas digitales o hash de archivos.

### 3.3.3 Fiabilidad

La subsección de Fiabilidad tiene como objetivo especificar los factores de fiabilidad necesarios del sistema, los cuales se refieren a la capacidad del software para realizar sus funciones de manera precisa y consistente a lo largo del tiempo. A continuación se detallan los requisitos de fiabilidad para el sistema:

* **Tiempo entre incidentes permisibles:** Se debe especificar el tiempo máximo permitido entre incidentes o fallas en el sistema. Esto se refiere al tiempo que el sistema puede funcionar sin experimentar interrupciones o errores significativos. Se establece un umbral aceptable para la frecuencia de incidentes, definiendo el tiempo máximo de inactividad o mal funcionamiento del sistema antes de que se considere una degradación de la fiabilidad.
* **Total de incidentes permisible:** Se establece un límite para el número total de incidentes o fallas permitidas dentro de un período determinado. Este requisito define el nivel de tolerancia a las fallas y establece un objetivo para mantener el sistema funcionando de manera confiable y sin interrupciones significativas.
* **Capacidad de recuperación y tolerancia a fallos:** El sistema debe contar con mecanismos de recuperación y tolerancia a fallos para minimizar el impacto de incidentes o interrupciones. Debe ser capaz de detectar y recuperarse de errores de manera autónoma, garantizando la continuidad del servicio y evitando la pérdida de datos o la degradación de la funcionalidad.
* **Mantenibilidad y facilidad de corrección:** El sistema debe ser diseñado de manera que sea fácil de mantener y corregir en caso de incidentes o errores. Se deben establecer procedimientos claros y eficientes para la identificación, diagnóstico y corrección de problemas, facilitando la rápida resolución de incidencias y minimizando el tiempo de inactividad.

### 3.3.4 Disponibilidad

La subsección de Disponibilidad tiene como objetivo especificar los factores de disponibilidad final exigidos al sistema, lo cual se refiere al porcentaje de tiempo en el que el software debe estar disponible y en pleno funcionamiento. A continuación, se detallan los requisitos de disponibilidad para el sistema:

* **Porcentaje de disponibilidad:** Se establece el nivel de disponibilidad requerido para el sistema, expresado como un porcentaje del tiempo total en el que el software debe estar operativo. Por ejemplo, si se especifica una disponibilidad del 99%, significa que el sistema debe estar disponible al menos el 99% del tiempo.
* **Tiempo máximo de inactividad permitido:** Se define el tiempo máximo de inactividad o interrupción permitido para el sistema en un período determinado. Esto implica que cualquier período de inactividad programada o no planificada no debe exceder el límite establecido, asegurando que el software esté disponible para su uso en la mayoría de las situaciones.
* **Mantenimiento planificado:** Se deben establecer períodos de mantenimiento planificado durante los cuales el sistema puede estar temporalmente no disponible. Estos períodos deben ser programados de antemano y comunicados a los usuarios para minimizar el impacto en la disponibilidad del sistema.
* **Respaldo y recuperación de datos:** Se deben implementar medidas de respaldo y recuperación de datos para garantizar la disponibilidad continua del sistema. Esto implica realizar copias de seguridad periódicas de la información crítica y establecer procesos para recuperar rápidamente los datos en caso de pérdida o corrupción.

### 3.3.5 Mantenibilidad

La subsección de Mantenibilidad tiene como objetivo identificar el tipo de mantenimiento necesario para el sistema y especificar quién debe realizar las tareas de mantenimiento, así como cuándo deben realizarse. A continuación, se detallan estos aspectos:

* **Tipo de mantenimiento:** Se debe identificar el tipo de mantenimiento necesario para el sistema, como mantenimiento correctivo, preventivo o evolutivo. El mantenimiento correctivo se refiere a la corrección de errores o fallas en el sistema. El mantenimiento preventivo implica la realización de tareas periódicas para evitar posibles problemas o mejorar el rendimiento del sistema. El mantenimiento evolutivo se enfoca en la incorporación de nuevas funcionalidades o mejoras al sistema.
* **Responsable del mantenimiento:** Se especifica quién será responsable de realizar las tareas de mantenimiento. Esto puede incluir a los usuarios del sistema, personal de soporte técnico o desarrolladores del software.
* **Programación de tareas de mantenimiento:** Se establece cuándo deben realizarse las tareas de mantenimiento de acuerdo a un cronograma predefinido. Por ejemplo, la generación de estadísticas de acceso semanales y mensuales puede ser programada para evaluar el rendimiento del sistema y realizar ajustes si es necesario.

### 3.3.6 Portabilidad

La subsección de Portabilidad tiene como objetivo especificar los atributos que el software debe presentar para facilitar su traslado a otras plataformas o entornos. A continuación, se detallan algunos elementos que pueden incluirse:

* **Porcentaje de componentes dependientes del servidor:** El porcentaje de componentes del software que dependen del servidor es de un 90%. Esto se refiere a las funcionalidades o características que requieren una conexión y comunicación constante con el servidor para su correcto funcionamiento.
* **Porcentaje de código dependiente del servidor:** El porcentaje de código del software que está específicamente diseñado para funcionar en el servidor es de aproximadamente un 70%. Esto implica aquellas partes del software que no pueden ejecutarse de manera autónoma en un entorno diferente al servidor.
* **Uso de un determinado lenguaje por su portabilidad:** Se menciona el uso de HTML, Python y JavaScript se elige por su capacidad de portabilidad, ya que son conocidos por ser altamente portables y permitir la ejecución del software en diferentes plataformas.
* **Uso de un determinado compilador o plataforma de desarrollo:** Se utilizará la plataforma de desarrollo Visual Code ya que facilita la portabilidad del software.
* **Uso de un determinado sistema operativo:** Al ser una plataforma web, es compatible con cualquier sistema operativo.

## 3.4 Otros Requisitos

**3.4.1 Usabilidad**

La usabilidad es un requisito fundamental para garantizar una buena experiencia de usuario. Se debe enfocar en la facilidad de uso y la satisfacción del usuario al interactuar con el software. Esto implica que la interfaz de usuario debe ser intuitiva, con una navegación clara y coherente. Se deben proporcionar instrucciones claras y ayuda contextual para guiar a los usuarios en el uso del software. Además, se puede requerir la disponibilidad de documentación detallada, tutoriales o material de capacitación para facilitar el aprendizaje y la adopción del sistema.

**3.4.2 Escalabilidad**

La escalabilidad se refiere a la capacidad del software para manejar un crecimiento en términos de carga de usuarios, volumen de datos o transacciones sin comprometer su rendimiento y funcionalidad. Los requisitos de escalabilidad pueden incluir la capacidad de procesamiento para manejar un aumento en la demanda, la capacidad de almacenamiento para manejar un aumento en los datos y la capacidad de escalado horizontal o vertical para adaptarse a un crecimiento sustancial.

**3.4.3 Componentes de Bootstrap**

La inclusión de componentes de Bootstrap implica el uso de un marco de trabajo de desarrollo web que proporciona elementos de diseño predefinidos y estilos para acelerar el proceso de desarrollo y garantizar la coherencia visual del software.

**3.4.4 Diseño responsivo**

El requisito de diseño responsivo implica que el software debe ser capaz de adaptarse y mostrarse correctamente en diferentes dispositivos y tamaños de pantalla. Esto implica que la interfaz de usuario debe ser flexible y ajustarse de manera fluida a los cambios de tamaño de pantalla, ya sea en dispositivos móviles, tablets o escritorios. El diseño responsivo garantiza que los elementos y contenido se presenten de forma legible y estéticamente agradable en cualquier dispositivo, ofreciendo una experiencia de usuario consistente y satisfactoria.

**3.4.5 Modo Oscuro**

El requisito del modo oscuro implica proporcionar una opción para que los usuarios puedan cambiar entre una apariencia clara y una apariencia oscura en el software. Esto implica que se debe desarrollar una interfaz que pueda adaptarse a ambos modos de visualización, asegurando que los elementos y el contenido sean legibles y atractivos en ambos casos. El modo oscuro no solo es una cuestión estética, sino que también puede ayudar a reducir la fatiga visual y mejorar la experiencia de uso en entornos de poca iluminación.

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

En esto abordamos la migración del front-end existente a Django, lo cual se seguirá una planificación detallada que considere los días totales estimados y las personas involucradas en su ejecución.

Se aplicaran las buenas prácticas y se establecerán las condiciones necesarias para un buen término de este proyecto purrfect pet shop.

Este proceso de migración requerirá un análisis exhaustivo de la estructura y funcionalidades existente del front-end, con el objetivo de identificar los componentes y la lógica que se deben implementar en Django.

Se establecen hitos y entregables intermedios para así medir el progreso y asegurar la calidad del trabajo realizado. Se asignaran recursos necesarios, incluyendo desarrolladores con experiencia en Django y conocimiento del front-end existente.

El equipo trabajará de manera colaborativa, siguiendo las metodologías ágiles que permitan una comunicación eficiente y de adaptación flexible a los cambios y requerimientos que pudieran surgir. como también se implementarán las mejores prácticas de desarrollo en django asegurando que la estructura del codigo modular, legible y mantenible como también se mantendrá especial atención a la seguridad, rendimiento y escalabilidad del sistema considerando aspectos de manejo de sesiones, autenticacion, validacion de datos y consultas eficientes a la base de datos.

La planificación también contemplara pruebas para así garantizar una integridad del sistema y compatibilidad entre el front-end y el back-end como también se realizarán pruebas unitarias de integración y de aceptación con un objetivo de identificar y corregir cualquier problema o error antes de la implementación final.

### 4.1.2 Definición del Equipo de Trabajo

El equipo de trabajo definido para el proyecto de migración del front-end a Django está compuesto por dos integrantes: GianFranco Soto Muñoz. Cada miembro del equipo del equipo desempeñará un rol específico y tendrá funciones definidas para asegurar el éxito del proyecto. A continuación se presenta la tabla de definición de roles y funciones:

| **integrante** | **Rol Definido** | **funciones** |
| --- | --- | --- |
| *-* | *Project manager* | *-Supervisar y coordinar el proyecto en su conjunto.* |
| *-* |  | *-Establecer la planificación y los hitos del proyecto.* |
| *-* |  | *-Gestionar los recursos y el presupuesto del proyecto.* |
| *-* |  | *-Facilitar la comunicación entre los miembros del equipo.* |
| *-* |  | *-realizar el seguimiento del progreso y los resultados.* |
| *-* |  | *-Tomar decisiones estratégicas y resolver problemas.* |
| *GianFranco Soto Munnioz* | *Developer* | *-Desarrollar el back-end utilizando Django.* |
|  |  | *-Implementar la seguridad y la gestión de usuarios.* |
|  |  | *-Realizar pruebas unitarias, de integración y de aceptación.* |
|  |  | *-Identificar y resolver errores o problemas en el sistema.* |
|  |  | *-Asegurar la calidad y la estabilidad del sistema.* |
| *Jose Esteban Romero* | *Developer* | *-Desarrollar el back-end utilizando django.* |
|  |  | *-Migrar y adaptar la lógica y los componentes del front-end.* |
|  |  | *-Implementar la seguridad y la gestión de usuarios.* |
|  |  | *-Integrar y probar las funcionalidades del sistema.* |
|  |  | *-Optimizar el rendimiento y la escalabilidad del back-end.* |

### 4.1.3 Definición de Actividades principales del Proyecto

El proyecto se basará en estándares y buenas prácticas de gestión de la planificación del PMI(PROJECT MANAGEMENT INSTITUTE) y la ingeniería de software. A continuación, se describen las principales fases y actividades que se consideran en la programación de la planificación:

1.Inicio del proyecto.  
2.Análisis de Requisitos.

3.Diseño del sistema.

4.Implementación y desarrollo.

5.Pruebas y Validación.

6.Despliegue y puesta en Producción.

7.Mantenimiento y Mejora Continua.

## 4.2 Plan de Control de Cambio

El plan de Control de cambio se encarga de gestionar los cambios que puedan surgir durante el desarrollo del software, ya sea por requerimientos del usuario, mejoras o correcciones detectadas por el equipo. A continuación, se describen los tipos de cambios que se podrán resolver y sus alcances, seguido de la tabla de control de cambios:

Tipos de cambio y alcances:

1.- Cambios de Funcionalidades:

\*Se podrán realizar modificaciones en la lógica y las funcionalidades del sistema.

\*Se permitirá la adición de nuevas características o procesos.

\*Será posible eliminar funcionalidades existentes.

2.- Cambios en Requisitos no funcionales:

\*Se podrán realizar modificaciones en aspectos de rendimiento, escalabilidad y seguridad.

\*Se permitirán cambios en la interfaz de usuario y la usabilidad.

\*Será posible actualizar las Tecnologías y herramientas utilizadas.

3.- Cambios en la arquitectura del sistema:

\*Se podrán realizar modificaciones en la estructura y la organización del sistema.

\*Se permitirá la incorporación de nuevos componentes o servicios.

\*Serán posibles cambios en la integración con sistemas externos.

4.- Corrección y mejoras:

\*Se podrán realizar correcciones y solución de errores identificados.

\*Se permitirá la optimización de código y mejora del rendimiento.

\*Será posible actualizar bibliotecas y dependencias.

El control de cambios se aplicará en diferentes instancias del proyecto, considerando la etapa del desarrollo y las fases posteriores. Los cambios serán evaluados y validados en función de su impacto en el proyecto y su alineación con los objetivos establecidos.

Pero también es importante destacar que existen casos en los que no será posible aplicar cambios, como en situaciones en las que los cambios requeridos no están alineados con los objetivos del proyecto, exceden los recursos disponibles o comprometen la estabilidad del sistema.