| DUOC UC - Escuela de informática y telecomunicaciones |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: [DBZ\_STORE]* |
|  |
| **Revisión*: [02]*** |
| **[26-06-2023]** |

| 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**](#_30j0zll) 3

[**1. Introducción**](#_1fob9te) 4

[1.1.](#_3znysh7) Propósito 4

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

[1.3.](#_tyjcwt) Definiciones, Acrónimos y Abreviaturas 5

[1.4.](#_3dy6vkm) Referencias 5

[1.5.](#_1t3h5sf) Visión General del Documento 5

[**2.**](#_4d34og8) **Descripción General** 6

[2.1.](#_2s8eyo1) Perspectiva del Producto 6

[2.2.](#_17dp8vu) Funciones del Producto 6

[2.3.](#_3rdcrjn) Características de los Usuarios 7

[2.4.](#_26in1rg) Restricciones 7

[2.5.](#_lnxbz9) Suposiciones y Dependencias 7

[2.6.](#_35nkun2) Requisitos Futuros 8

[**3.**](#_1ksv4uv) **Requisitos Específicos** 9

[3.1](#_44sinio) Requisitos comunes de las interfaces 9

[*3.1.1*](#_2jxsxqh) *Interfaces de usuario* 9

[*3.1.2*](#_z337ya) *Interfaces de hardware* 10

[*3.1.3*](#_3j2qqm3) *Interfaces de software* 10

[*3.1.4*](#_1y810tw) *Interfaces de comunicación* 11

[3.2](#_4i7ojhp) Requisitos funcionales 11

[3.3](#_2xcytpi) Requisitos no funcionales 12

[*3.3.1*](#_1ci93xb) *Requisitos de rendimiento* 12

[*3.3.2*](#_3whwml4) *Seguridad* 13

[*3.3.3*](#_2bn6wsx) *Fiabilidad* 13

[*3.3.4*](#_qsh70q) *Disponibilidad* 14

[*3.3.5*](#_3as4poj) *Mantenibilidad* 14

[*3.3.6*](#_1pxezwc) *Portabilidad* 15

[3.4](#_49x2ik5) Otros Requisitos 15

[**4. Propuesta de Planificación**](#_2p2csry) 16

[4.1 Descripción general acerca de la Planificación](#_147n2zr) 17

[*4.1.2 Definición del Equipo de Trabajo*](#_3o7alnk) 18

[*4.1.3 Definición de Actividades principales del Proyecto*](#_23ckvvd) 18

[4.2 Plan de Control de Cambio](#_ihv636) 18

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *20-06-2023* | *Ver qué nuevas funciones agregar al proyecto, ver Crud* | *Isaias Acevedo* | *Agregar nuevas funciones al proyecto, tales como son las del CRUD, agregar, eliminar, obtener, editar.* |
| *23-06-2023* | *Ver la base de datos Django, buscar alertas.* | *Claudio Cornejo - Ariel Leiva* | *Pasar el proyecto a Django, agregar alertas de mensajes al proyecto, agregar carrito de compras conectado a la base de datos.* |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Claudio Cornejo* | *Gerente de Proyecto, Probador de Software.* |
| *Ariel Leiva* | *Desarrollador de Software, Probador de Software.* |
| *Isaias Acevedo* | *Diseñador de Software, Desarrollador de Software.* |
|  |  |
|  |  |

# 

# 

# 

# 1. Introducción

En este documento se darán las Especificaciones de Requisitos de Software (ERS) para el desarrollo del sistema de gestión de una tienda en línea. El objetivo del sistema es automatizar y simplificar las ventas de una tienda enfocado en productos japoneses, permitiendo a los empleados llevar un seguimiento más eficiente de los productos y reducir el riesgo de errores en el proceso de gestión de las ventas realizadas por la tienda en línea.

Este documento describe los requisitos funcionales y no funcionales del software y se basa en la información proporcionada por el cliente y los usuarios finales del sistema. También se incluyen detalles sobre la arquitectura y el diseño del sistema, así como los criterios de los diseñadores y las pruebas necesarias para asegurar que el software cumpla con los requisitos especificados.

## 1.1. Propósito

El **equipo de desarrollo** utilizará este documento como una guía para diseñar, construir y probar el sistema de gestión de la tienda en línea. Además, el documento será utilizado por el **equipo de desarrollo** para verificar que el software cumpla con los requisitos especificados y por el **cliente** para validar que el sistema cumpla con sus expectativas y necesidades de negocio.

## 1.2. Ámbito del Sistema

* el nombre del sistema será: ”DbzStore”.
* La función principal del sistema será vender todo tipo de productos para cualquier tipo de cliente, edad o nacionalidad, este sistema tendrá varias opciones de compra y será fácil de entender para cualquier usuario, el sistema no contará con información traficada, productos falsos o contenido fuera de lugar. El sistema no hará reportes, no responderá preguntas que no tengan que ver con el sistema, o dará dinero a usuarios, entre otros.
* Los beneficios del sistema será las ventas que realizará, la información que generará, además de todas las opciones que tendrá para el usuario, uno de los objetivos principales para este sistema será proporcionar los productos que el cliente necesite, además de darse a conocer por su calidad, buenos servicios y precios, una de las metas del sistema será registrar una de las mayores ventas realizadas a lo largo de la historia, también será ir actualizando el sistema dia a dia hasta que alcance un buen desarrollo profesional.

## 1.3. Definiciones, Acrónimos y Abreviaturas

1. Definiciones a Utilizar: **Algoritmo**(Un conjunto de instrucciones precisas y bien definidas que se utilizan para resolver un problema o realizar una tarea.), **Codificación**(La conversión de datos en un formato que una computadora o dispositivo pueda entender y procesar.), **Programación**(El proceso de escribir código para crear software, aplicaciones o sitios web.), **Base de Datos**(Una colección estructurada de información que se almacena en una computadora y se puede acceder, actualizar y gestionar de manera eficiente.), **Sistema operativo**(El software que controla los recursos de hardware y software de una computadora y proporciona servicios comunes para las aplicaciones del usuario.).
2. Acrónimos a Utilizar: **CPU**(Unidad central de procesamiento), **GPU**(Unidad de procesamiento de gráficos), **HTML**(Lenguaje de marcado de hipertexto), **CSS**(Hojas de estilo en cascada), **JS**(JavaScript), **URL**(Localizador Uniforme de Recursos),**LAN**(Red de Área Local), **WAN**(Red de Área Amplia), **VPN**(Red Privada Virtual), **RAM**(Memoria de acceso aleatorio), **HDD**(Disco Duro), **SSD**(Unidad De Estado Sólido), **IP**(Protocolo de Internet).
3. Abreviaturas a Utilizar: SQL, HTML, CSS, IP, LAN, HTTPS, loT, entre otros.

## 1.4. Referencias

* GitHub. (2023). "GitHub Pages".
* W3C. (2023). "HTML5 Specification".
* W3C (World Wide Web Consortium). (2023). "CSS Cascading and Inheritance Level 4 Specification".
* Ecma International. (2023). "ECMAScript® 2019 Language Specification".

## 1.5. Visión General del Documento

Este documento será utilizado para servir de guía en todo momento por el equipo de desarrollo además de ir viendo el avance del proyecto a lo largo del tiempo, servirá para anotar los requerimientos del usuario como también detallar cada una de las especificaciones y funciones del sistema.

Uno de los objetivos generales del proyecto es alcanzar el desarrollo sustentable por un periodo indefinido de tiempo, que el equipo de desarrollo logre lo que los usuarios esperan, además de entregar un buen producto final, las características claves del producto final serán la búsqueda de productos, la venta de productos, la filtración por tipo y cantidad, además de varias opciones dentro del sistema.

Por ahora el proyecto no presenta limitaciones, pero dentro de un periodo acotado de tiempo tendremos que ver lo recursos económicos que se necesitaran para el buen funcionamiento del sistema, también se verá el tiempo que presenta el equipo de desarrollo y todas las cosas que requiera el usuario.

Este proyecto va dirigido para todo público ya que se presentará a todo el mundo, el producto cuenta con varias opciones y será entendible para todo tipo de usuarios, la expectativa del proyecto es que alcance la mayoría de usuarios en un acotado tiempo, esperando que se tenga éxito en este.

Una lista resumida de los requisitos que presentará el sistema será: navegación rápida, interfaz de usuario, registro e inicio de sesión, búsqueda, filtros, contenidos variados, seguridad de información, formularios entre otros requisitos.

# 2. Descripción General

El sistema se enfocara en la gestión de venta de mangas, para los clientes será enfocado en los procesos de compra de mangas, también cabe recalcar que la página no tendrá interacciones o dependencia de cualquier tipo de programas ya que es un producto sustentable.

## 2.1. Perspectiva del Producto

El sistema de venta de mangas se llevará a cabo ocupando una base de datos, la cual será la encargada de manejar todo sobre los mangas, por ejemplo los precios, nombre, información sobre los productos y toda aquella información del usuario que pueda ser ocupada para beneficio de la página web.

Parte importante es la interacción que tendrá el usuario, la cual será mediante una interfaz gráfica en la cual el usuario deberá navegar, ya sea en formato web o móvil.

## 2.2. Funciones del Producto

• Registro usuario

• Registro de existencias

• Venta de productos

• Compra de productos

• Registro de proveedores

## 2.3. Características de los Usuarios

Habrá 2 tipos de usuarios, los cuales son dueños dé la página y los que son clientes de la página.

Enfocándonos en los clientes, estos tendrán que tener un conocimiento mínimo de navegar por el internet y saber ocupar teléfonos o computador, ya que al ser una página web necesitamos que lleguen clientes los cuales estén interesados, pudiendo como objetivo a todo aquel cliente que guste de los mangas o comics

## 2.4. Restricciones

Esta subsección describe aquellas limitaciones que se imponen sobre los desarrolladores del producto:

• Paleta de colores

• Limitaciones del hardware.

• La interfaz debe ser Responsive

• Debe ser trabajada mediante html, css, javaScript y django

• Todo aquel producto en venta deberá estar registrado en la base de datos

• Todo lo que tenga que ver con las consultas deberá ser trabajado con SQL

• compromiso con el equipo para presentarse a los horarios de trabajo.

## 2.5. Suposiciones y Dependencias

El sistema presentará funciones de búsqueda por filtro, se podrá buscar el producto en base a lo que el usuario requiera, el sistema presentará vistas atractivas para todo usuarios, además de funcionalidades de alto rendimiento, las dependencias del sistema serán, los lenguajes html, css, js, django, entre otros programas que el equipo de desarrollo utilizara para estar en continua actualización con el sistema, como por ejemplo: github, visual studio code e extensiones de este mismo.

## 2.6. Requisitos Futuros

Como todo sistema será susceptible en los hábitos de mejora y crecimiento debido a esto creemos que como equipos hay funcionalidades que podrán ir añadiendo con el tiempo,

en el ámbito de venta y compras afinar los procesos de tarjetas o depósitos con comprobantes.

Un sistema el cual pueda adaptarse a los procesamientos de la base de datos para así poder soportar más funcionalidades, ya sea aumento en terminales y cargas de existencias.

# 3.Requisitos Específicos

* Diseño atractivo y amigable: La página debe tener un diseño atractivo y fácil de usar para atraer a los usuarios y brindarles una experiencia agradable mientras navegan por el sitio.
* Catálogo de mangas: La página debe contar con un catálogo completo de mangas disponibles para la venta. Cada manga debe tener su propia página con información detallada, como título, autor, género, sinopsis y precio.
* Sistema de búsqueda y filtrado: Los usuarios deben poder buscar mangas por título, autor o género. Además, se deben proporcionar opciones de filtrado para ayudar a los usuarios a encontrar los mangas de su interés más fácilmente.

Carrito de compras: Los usuarios deben poder agregar mangas al carrito de compras y gestionar su contenido antes de finalizar la compra.

* Proceso de compra: Debe haber un flujo claro y sencillo para que los usuarios completen sus compras. Esto incluye ingresar la información de envío, seleccionar el método de pago y confirmar el pedido.
* Gestión de usuarios: Se requiere un sistema de registro de usuarios que permita a los clientes crear una cuenta, acceder a su historial de compras, guardar mangas en una lista de deseos y realizar un seguimiento de los pedidos.
* Sistema de pago seguro: La página debe implementar un sistema de pago seguro que proteja la información financiera de los usuarios durante las transacciones. Esto podría incluir la integración con servicios de pago confiables como PayPal o Stripe.
* Comentarios y valoraciones: Los usuarios deben poder dejar comentarios y valoraciones sobre los mangas que han comprado para ayudar a otros clientes en su elección.
* Gestión de inventario: Se necesita un sistema que controle el inventario de mangas, actualizando automáticamente la disponibilidad de los títulos cuando se realizan compras.
* Integración de redes sociales: Se debe permitir a los usuarios compartir mangas o enlaces a la página en redes sociales para aumentar la visibilidad y promoción del sitio.
* Soporte al cliente: Debe haber una sección de contacto o un sistema de tickets de soporte para que los usuarios puedan realizar consultas, resolver problemas o solicitar información adicional.
* Responsividad y adaptabilidad: La página debe ser totalmente responsiva y compatible con diferentes dispositivos, como computadoras de escritorio, tabletas y teléfonos móviles.

## 3.1 Requisitos comunes de las interfaces

Usabilidad: La interfaz debe ser fácil de usar, intuitiva y permitir a los usuarios realizar tareas de manera eficiente.

Navegación clara: Debe haber una estructura de navegación coherente y organizada que permita a los usuarios moverse sin problemas por la interfaz.

Diseño visual atractivo: La interfaz debe tener un diseño estético y agradable a la vista, utilizando colores, tipografías y elementos visuales coherentes y atractivos.

Consistencia: Los elementos de la interfaz, como botones, iconos y etiquetas, deben mantener una apariencia y comportamiento coherentes en todas las páginas y secciones.

Retroalimentación y respuesta: La interfaz debe proporcionar retroalimentación visual o auditiva para indicar que una acción se ha completado correctamente o que se ha producido un error.

Flexibilidad y personalización: Los usuarios deben poder adaptar la interfaz a sus preferencias y necesidades, por ejemplo, ajustando el tamaño del texto o configurando opciones de accesibilidad.

Eficiencia: La interfaz debe permitir a los usuarios realizar tareas de manera eficiente, minimizando la cantidad de pasos o acciones necesarios para completar una tarea.

Tamaño y legibilidad: Los elementos de la interfaz, como texto y botones, deben ser lo suficientemente grandes y legibles para que los usuarios puedan interactuar con ellos fácilmente, incluso en dispositivos móviles.

Accesibilidad: La interfaz debe ser accesible para usuarios con discapacidades, cumpliendo con pautas y estándares de accesibilidad, como el uso adecuado de etiquetas, contrastes de color y soporte para lectores de pantalla.

Feedback y ayuda: La interfaz debe proporcionar ayuda contextual y orientación a los usuarios en caso de que necesiten información adicional o se encuentren con problemas.

Eficiencia en la recuperación de errores: En caso de que ocurra un error, la interfaz debe informar claramente al usuario y proporcionar soluciones o sugerencias para resolverlo.

### 3.1.1 Interfaces de usuario

Interfaz de usuario gráfica (GUI, por sus siglas en inglés): Es una interfaz visual que permite a los usuarios interactuar con elementos gráficos como ventanas, botones, menús y cuadros de diálogo. Se utiliza en aplicaciones de escritorio y aplicaciones móviles.

Interfaz de línea de comandos (CLI, por sus siglas en inglés): Es una interfaz basada en texto que permite a los usuarios ingresar comandos y recibir respuestas en forma de texto. Se utiliza principalmente en sistemas operativos y herramientas de línea de comandos.

Interfaz web: Es una interfaz que se accede a través de un navegador web. Permite a los usuarios interactuar con elementos como enlaces, formularios, botones y menús desplegables. Se utiliza en sitios web, aplicaciones web y sistemas basados en la nube.

Interfaz móvil: Es una interfaz diseñada específicamente para dispositivos móviles, como teléfonos inteligentes y tabletas. Puede incluir gestos táctiles, pantallas deslizables, teclados virtuales y otras interacciones táctiles.

### 3.1.2 Interfaces de hardware

Interfaz de teclado: Permite a los usuarios ingresar datos y comandos mediante la pulsación de teclas. Se utiliza en computadoras, laptops, teléfonos con teclado físico y otros dispositivos que incorporan teclados.

Interfaz de ratón: Permite a los usuarios controlar el movimiento del cursor en una pantalla y realizar acciones mediante clics y movimientos del ratón. Se utiliza en computadoras de escritorio y laptops.

Interfaz de pantalla: Proporciona la salida visual del dispositivo al usuario. Puede ser una pantalla de cristal líquido (LCD), una pantalla de diodos emisores de luz (LED), una pantalla táctil o cualquier otra tecnología de visualización.

### 3.1.4 Interfaces de comunicación

Interfaz Ethernet: Permite la conexión y comunicación entre dispositivos a través de cables Ethernet. Se utiliza ampliamente para la conexión de redes locales (LAN) y conexión a Internet.

Interfaz Wi-Fi: Permite la conexión inalámbrica entre dispositivos a través de ondas de radio. Se utiliza para la conexión a redes locales y acceso a Internet sin cables.

## 3.2 Requisitos funcionales

Registro de usuarios: Permitir a los usuarios crear cuentas y gestionar sus datos de inicio de sesión.

Autenticación y autorización: Verificar la identidad de los usuarios y controlar su acceso a diferentes funciones y recursos del sistema.

Gestión de perfiles de usuario: Permitir a los usuarios personalizar su perfil y gestionar su información personal.

Búsqueda y filtrado: Proporcionar una funcionalidad de búsqueda para permitir a los usuarios encontrar información específica y aplicar filtros para refinar los resultados.

Creación, lectura, actualización y eliminación (CRUD): Permitir a los usuarios crear, leer, actualizar y eliminar datos en el sistema.

Procesamiento de transacciones: Permitir la realización de transacciones y operaciones comerciales, como compras, pagos y devoluciones.

Gestión de carrito de compras: Permitir a los usuarios agregar y gestionar elementos en un carrito de compras antes de finalizar una compra.

Gestión de pedidos: Permitir a los usuarios realizar un seguimiento de sus pedidos, ver el estado de entrega y recibir notificaciones sobre actualizaciones.

Integración de pasarelas de pago: Permitir pagos seguros en línea utilizando diferentes métodos de pago, como tarjetas de crédito, transferencias bancarias u otras pasarelas de pago.

Comentarios y valoraciones: Permitir a los usuarios dejar comentarios, reseñas y valoraciones sobre productos o servicios.

Gestión de inventario: Mantener un seguimiento preciso del stock de productos y actualizar automáticamente los niveles de inventario.

Generación de informes: Proporcionar informes y análisis para ayudar a los usuarios a comprender las tendencias, los datos de ventas y otros aspectos relevantes del sistema.

Funcionalidad multilingüe: Admitir múltiples idiomas y permitir a los usuarios seleccionar su preferencia de idioma.

Integración con redes sociales: Permitir a los usuarios compartir contenido del sistema en redes sociales y utilizar la autenticación de redes sociales para el acceso al sistema.

## 3.3 Requisitos no funcionales

1. -Rendimiento
2. -Usabilidad
3. -Seguridad
4. -Fiabilidad
5. -Escalabilidad
6. -Mantenibilidad
7. -Portabilidad

### 3.3.1 Requisitos de rendimiento

1. -Tiempo de Respuesta
2. -Capacidad de Carga
3. -Velocidad de procesamiento
4. -Eficiencia en los Recursos
5. -Tasa de T.D
6. -Tiempo de tareas Programadas

### 3.3.2 Seguridad

1. -Autenticación
2. -Encriptación
3. -Control de acceso
4. -Auditoría y registro
5. -Respaldo y recuperación
6. -Protección contra malware
7. -Protección de datos personales

### 3.3.3 Fiabilidad

* Disponibilidad
* Tolerancia a fallos
* Capacidad de recuperación
* Mantenimiento preventivo
* Integridad de datos
* Cumplimiento normativo
* Escalabilidad

### 3.3.4 Disponibilidad

1. -Tiempo de actividad
2. -Mantenimiento planificado
3. -Recuperación ante fallos
4. -Tolerancia a fallos
5. -Monitorización y alertas
6. -Plan de contingencia

### 3.3.5 Mantenibilidad

1. -Modularidad
2. -Documentación
3. -Estándares de codificación
4. -Separación de intereses
5. -Pruebas unitarias y de integración
6. -Control de versiones
7. -Facilidad de despliegue

### 3.3.6 Portabilidad

1. -Independencia de plataforma
2. -Interoperabilidad
3. -Requisitos de hardware mínimos
4. -Uso de estándares
5. -Adaptabilidad a diferentes resoluciones y pantallas

## 3.4 Otros Requisitos

1. -Interfaz de programación de aplicaciones (API)
2. -Usabilidad
3. -Integración
4. -Internacionalización y localización
5. -Confiabilidad

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

[Insertar una descripción de cómo se abordará el trabajo en cuanto a los días totales estimados y las personas involucradas en su ejecución, las buenas prácticas y condiciones necesarias a considerar para implementar para su buen término]

**Definición de requisitos y alcance del proyecto (2 semanas):**

* + Cliente y gerente de proyecto trabajan juntos para establecer los objetivos, requisitos y alcance del proyecto.
  + Diseñador y desarrollador de software participan en la discusión para comprender los requisitos del cliente y brindar su aporte.

**Diseño y planificación (2 semanas):**

* + Diseñador crea los bocetos y diseños de la interfaz de usuario.
  + Desarrollador de software revisa el diseño y propone una arquitectura técnica.
  + Gerente de proyecto coordina la planificación general del proyecto, establece hitos y plazos.

**Desarrollo del software (6 semanas):**

* + El desarrollador de software codifica y construye la aplicación siguiendo las mejores prácticas de programación.
  + El diseñador colabora con el desarrollador para asegurar la implementación adecuada de la interfaz de usuario.
  + El probador de software realiza pruebas funcionales y de rendimiento para garantizar la calidad del software.

**Pruebas y depuración (2 semanas):**

* + El probador de software identifica y reporta errores al desarrollador.
  + El desarrollador corrige los errores y realiza pruebas adicionales para verificar las correcciones.

**Implementación y despliegue (1 semana):**

* + El desarrollador realiza la integración final del software y se asegura de que esté listo para su implementación.
  + El gerente de proyecto coordina la implementación del software en el entorno de producción del cliente.

**Evaluación y seguimiento (2 semanas):**

* + Se realizan pruebas de aceptación por parte del cliente para asegurarse de que el software cumple con los requisitos acordados.
  + Se recopilan comentarios y se realizan ajustes finales si es necesario.

**Buenas prácticas para el proyecto:**

* Establecer una comunicación clara y constante entre todas las partes involucradas.
* Realizar reuniones regulares para realizar seguimiento del progreso y resolver problemas.
* Utilizar herramientas de gestión de proyectos para organizar tareas, plazos y asignaciones.
* Documentar adecuadamente los requisitos, el diseño, las pruebas y los cambios realizados durante el proyecto.
* Realizar pruebas exhaustivas para garantizar la calidad del software antes de su implementación.

**Condiciones necesarias para un buen término del proyecto:**

* Asignación adecuada de recursos humanos y técnicos.
* Compromiso y colaboración de todas las partes involucradas.
* Gestión eficiente del tiempo y los plazos establecidos.
* Flexibilidad para adaptarse a cambios o ajustes en los requisitos.
* Supervisión y control adecuados por parte del gerente de proyecto.

### 4.1.2 Definición del Equipo de Trabajo

| Equipo | Rol | Función Principal |
| --- | --- | --- |
| Claudio Cornejo | Gerente de Proyecto, Probador de Software. | Coordina y supervisa todo el proceso de desarrollo del software, desde la planificación hasta la entrega final, asegurando el cumplimiento de los plazos y objetivos. |
| Ariel Leiva | Desarrollador de Software, Probador de Software. | Escribe el código y programa el software, implementando las funcionalidades requeridas según las especificaciones del diseñador. |
| Isaias Acevedo | Diseñador de Software, Desarrollador de Software. | Crea la estructura y el diseño del software, centrándose en la arquitectura y lo pedido por el usuario. |

### 4.1.3 Definición de Actividades principales del Proyecto

[Descripción de las Principales fases y actividades que considera nuestra Programación de la Planificación argumentando bajo que estándares y buenas prácticas se basan (Gestión de la planificación PMI e Ingeniería de Software – es sólo enunciarlas]

| **Principales Fases** | **Descripción** | **Buenas Prácticas** |
| --- | --- | --- |
| Definición de requisitos y alcance del proyecto. | identificar y documentar las necesidades del proyecto y establecer los límites y objetivos del trabajo a realizar. | **-**Establecer una comunicación clara y constante entre todas las partes involucradas. |
| Diseño y planificación. | Crear la estructura y estrategia del proyecto, así como asignar los recursos necesarios. | **-**Realizar reuniones regulares para realizar seguimiento del progreso y resolver problemas. |
| Desarrollo del software. | Se programa y codifica el software, implementando las funcionalidades requeridas según los requisitos y diseños previamente establecidos. | **-**Utilizar herramientas de gestión de proyectos para organizar tareas, plazos y asignaciones. |
| Pruebas y depuración. | Evaluar el software para detectar errores y fallos, corrigiéndolos y asegurando su correcto funcionamiento. | **-**Documentar adecuadamente los requisitos, el diseño, las pruebas y los cambios realizados durante el proyecto. |
| Implementación y despliegue. | Instalación y puesta en marcha del software en el entorno de producción, asegurando su disponibilidad y funcionalidad para los usuarios finales. | **-**Realizar pruebas exhaustivas para garantizar la calidad del software antes de su implementación. |
| Evaluación y seguimiento. | Medir el rendimiento y la efectividad del software una vez implementado, así como realizar un seguimiento continuo para identificar posibles mejoras y garantizar la satisfacción de los usuarios. |  |

## 

## 4.2 Plan de Control de Cambio

[Se recomienda primero describir los tipos de cambio que se podrán resolver y sus alcances]

[Insertar Tabla de Control de Cambios]

| FECHA | AUTOR | DESCRIPCIÓN DEL CAMBIO | ESTADO DEL PROYECTO |
| --- | --- | --- | --- |
| 20-05-2023 | ISAIAS ACEVEDO | AGREGAR MODO OSCURO AL PROYECTO, Y BUSCAR API DEL RELOJ. | -EN PROCESO |
| 29-05-2023 | ARIEL LEIVA | AGREGAR VALIDACIONES DE FORMULARIOS | -EN PROCESO |
| 23-06-2023 | CLAUDIO CORNEJO | HACER LA CONEXIÓN CON LA BASE DE DATOS DJANGO, AGREGAR ALERTAS Y VALIDACIONES | -FINALIZADO |