| http://www.duoc.cl/sites/default/files/logo_summit_0.png |
| --- |
| Especificación de Requerimientos del Software |
| *Proyecto: Red Vecinal* |
|  |
| **Revisión*: 1.0*** |
| **23-10-2024** |

| **ISO/EIC/IEEE 29148 de Ingeniería de Requisitos** |
| --- |

**Tabla de Contenidos**

**Contenido**

[**Ficha del documento 4**](#_heading=h.gjdgxs)

[**1. Introducción 5**](#_heading=h.1ksv4uv)

[1.1. Propósito del Documento 5](#_heading=h.2jxsxqh)

[1.2. Ámbito del Sistema 5](#_heading=h.1y810tw)

[1.3. Definiciones, Acronimos y Abreviaturas 5](#_heading=h.4i7ojhp)

[*1.3.1. Perspectiva del producto 5*](#_heading=h.2xcytpi)

[*1.3.2. Funciones del producto 5*](#_heading=h.1ci93xb)

[*1.3.3. Características del usuario 5*](#_heading=h.3whwml4)

[*1.3.4. Limitaciones 5*](#_heading=h.2bn6wsx)

[**1.4. Referencias 5**](#_heading=h.qsh70q)

[**1.5. Visión General del Documento 5**](#_heading=h.3as4poj)

[**2. Descripción General 6**](#_heading=h.1pxezwc)

[2.1. Perspectiva del Producto 6](#_heading=h.49x2ik5)

[2.2. Funciones del Producto 6](#_heading=h.2p2csry)

[2.3. Características de los usuarios 6](#_heading=h.147n2zr)

[2.4. Restricciones 6](#_heading=h.3o7alnk)

[2.5. Suposiciones y Dependencias 6](#_heading=h.23ckvvd)

[2.6. Requisitos Futuros 7](#_heading=h.ihv636)

[**3. Requisitos Específicos 7**](#_heading=h.32hioqz)

[3.1. Requisitos de las Interfaces 8](#_heading=h.1hmsyys)

[3.1.1. Interfaces de Usuario 8](#_heading=h.41mghml)

[3.1.2. Interfaces de Hardware 8](#_heading=h.2grqrue)

[3.1.3. Interfaces de Software 8](#_heading=h.vx1227)

[*3.1.4. Interfaces de Comunicación 8*](#_heading=h.3fwokq0)

[3.2. Requisitos Funcionales 9](#_heading=h.1v1yuxt)

[3.3. Requisitos no Funcionales 9](#_heading=h.4f1mdlm)

[*3.3.1. Requisitos de Rendimiento 9*](#_heading=h.2u6wntf)

[*3.3.2. Seguridad 9*](#_heading=h.19c6y18)

[*3.3.3. Fiabilidad 9*](#_heading=h.3tbugp1)

[*3.3.4. Disponibilidad 10*](#_heading=h.igv2xye9695p)

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

[*3.3.6. Portabilidad 10*](#_heading=h.28h4qwu)

[3.4. Otros requisitos 10](#_heading=h.nmf14n)

[3.5. Requisitos de Base de Datos Lógica. 10](#_heading=h.37m2jsg)

[**4. Propuesta de Planificación 11**](#_heading=h.1mrcu09)

[4.1. Descripción general acerca de la planificación 11](#_heading=h.46r0co2)

[*4.1.1. Definición del Equipo de Trabajo 11*](#_heading=h.2lwamvv)

[*4.1.2. Definición de Actividades principales del Proyecto 11*](#_heading=h.111kx3o)

[*4.1.3. Diagrama EDT 12*](#_heading=h.3l18frh)

[*4.1.4. Carta Gantt 13*](#_heading=h.206ipza)

[*4.1.5. Resumen de Costos del Desarrollo del Proyecto 13*](#_heading=h.4k668n3)

[4.2. Plan de control de Cambio 13](#_heading=h.2zbgiuw)

[**5. Anexos 14**](#_heading=h.1egqt2p)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *15-10-2024* | *Primera Revisión* | *Victor Navarro* | *Inicio* |
| *23-10-2024* | *Ultima revisión* | *Victor Navarro* | *Ultimos detalles para entregar* |

1. **Introducción**

Este documento de **Especificación de Requerimientos de Software (ERS)** describe detalladamente los requisitos funcionales y no funcionales del sistema de gestión de la junta de vecinos. El propósito es proporcionar una guía clara para el desarrollo y la implementación de dicho sistema, asegurando que cumpla con las expectativas de los usuarios finales.

* 1. **Propósito del Documento**

El propósito de este documento es describir de manera exhaustiva los requisitos del sistema de gestión de la junta de vecinos. Esta especificación será utilizada por el equipo de desarrollo, los miembros del equipo de proyecto y los stakeholders. Servirá como referencia para asegurar que el sistema final cumpla con los objetivos y las necesidades identificadas.

Este documento está dirigido a:

* Desarrolladores e ingenieros de software involucrados en la implementación del sistema.
* Miembros del directorio de la junta de vecinos y usuarios finales que utilicen el sistema.
* Administradores del sistema encargados de su gestión y mantenimiento.
  1. **Ámbito del Sistema**

El sistema, denominado Sistema de Gestión Vecinal, tiene como objetivo mejorar la organización y comunicación dentro de una junta de vecinos, facilitando la inscripción de vecinos, la emisión de certificados, la postulación de proyectos vecinales, la reserva de espacios comunitarios y la publicación de noticias.

**Lo que el sistema hará**:

* Permitirá a los vecinos inscribirse en la junta, solicitar certificados de residencia y postular proyectos vecinales.
* Gestionará reservas de espacios comunitarios a través de un calendario.
* Enviará notificaciones a los vecinos vía email, WhatsApp, o tipo afiche.
* Publicará noticias y actividades para los miembros de la junta.

**Lo que el sistema no hará**:

* No manejará redes sociales directamente.
* No ofrecerá soporte técnico avanzado para usuarios externos al sistema.

**Beneficios esperados**:

* Mejora en la eficiencia de la gestión de actividades comunitarias.
* Ahorro de tiempo en la emisión de certificados y en la comunicación con los vecinos.
* Mayor participación vecinal a través de la inscripción digital y el acceso a proyectos y noticias.
  1. **Definiciones, Acronimos y Abreviaturas**
* ERS: Especificación de Requerimientos de Software
* API: Application Programming Interface (Interfaz de Programación de Aplicaciones)
* MVC: Modelo Vista Controlador
* HTTPS: Protocolo Seguro de Transferencia de Hipertexto
* UI/UX: User Interface/User Experience (Interfaz de Usuario/Experiencia de Usuario)
* SMTP: Simple Mail Transfer Protocol (Protocolo Simple de Transferencia de Correo)
* OAuth: Protocolo de autorización abierto
* WCAG: Web Content Accessibility Guidelines (Pautas de Accesibilidad para el Contenido Web)
  + 1. **Perspectiva del producto**

El Sistema de Gestión Vecinal es un sistema web diseñado para ser usado por juntas de vecinos en Chile. Está basado en la arquitectura MVC y es accesible desde dispositivos móviles y de escritorio.

* + 1. **Funciones del producto**
* Inscripción de vecinos en la junta.
* Solicitud y emisión de certificados de residencia.
* Postulación y gestión de proyectos vecinales.
* Reservas de espacios comunitarios.
* Publicación de noticias y envío de notificaciones.
  + 1. **Características del usuario**
* Vecinos: Usuarios que pueden registrarse, solicitar certificados, y postular proyectos.
* Administradores: Personal que gestiona solicitudes, aprueba proyectos y reservas.
  + 1. **Limitaciones**
* El sistema requiere conexión a Internet.
* El acceso a funciones administrativas está limitado a personal autorizado.
  1. **Referencias**
* ISO/IEC/IEEE 29148:2018 - Ingeniería de sistemas y software - Procesos del ciclo de vida - Ingeniería de requisitos
* Guía PMBOK 7ma Edición - Project Management Institute
* WCAG 2.1 - Pautas de Accesibilidad para el Contenido Web
* Documentación oficial de Django 4.2
* Normativa chilena sobre protección de datos personales (Ley 19.628)
  1. **Visión General del Documento**

Este documento de Especificación de Requerimientos de Software (ERS) está organizado de la siguiente manera:

Sección 1 - Introducción: Proporciona una visión general del proyecto, incluyendo el propósito, ámbito del sistema, definiciones importantes y referencias utilizadas.

Sección 2 - Descripción General: Describe el contexto del sistema, incluyendo la perspectiva del producto, funciones principales, características de los usuarios, restricciones del sistema y dependencias importantes.

Sección 3 - Requisitos Específicos: Detalla los requisitos del sistema a nivel técnico, incluyendo:

* Requisitos de interfaces (usuario, hardware, software y comunicaciones)
* Requisitos funcionales del sistema
* Requisitos no funcionales (rendimiento, seguridad, fiabilidad, etc.)
* Requisitos de base de datos

Sección 4 - Propuesta de Planificación: Presenta el plan de implementación del proyecto, incluyendo:

* Descripción general de la planificación
* Definición del equipo de trabajo
* Actividades principales
* Diagrama EDT y Carta Gantt
* Estimación de costos
* Plan de control de cambios

Sección 5 - Anexos: Contiene información complementaria y documentación adicional relevante para el proyecto.

Cada sección está diseñada para proporcionar información progresivamente más detallada, comenzando con una visión general y avanzando hacia especificaciones técnicas precisas y planes de implementación.

1. **Descripción General**

En esta sección se contextualizan los factores que afectan al sistema propuesto, proporcionando un marco general para definir los requisitos específicos en las secciones posteriores. Esto ayudará a comprender mejor las características y funciones del sistema.

* 1. **Perspectiva del Producto**

El Sistema de Gestión Vecinal será una plataforma web accesible desde dispositivos móviles y de escritorio, diseñada para facilitar la gestión interna de las juntas de vecinos y su interacción con los residentes. El sistema integrará módulos para la inscripción de vecinos, la solicitud de certificados de residencia, la postulación de proyectos vecinales, la reserva de espacios comunitarios, y la publicación de noticias y avisos.

El sistema está destinado a organizaciones comunitarias en Chile, como juntas de vecinos, y será utilizado principalmente por sus administradores y miembros.

* 1. **Funciones del Producto**

El sistema tendrá las siguientes funciones principales:

* Inscripción de vecinos: Los usuarios podrán registrarse en la junta de vecinos a través de un formulario en línea.
* Gestión de solicitudes de certificados: Los vecinos podrán solicitar certificados de residencia, que serán revisados y aprobados o rechazados por los administradores.
* Postulación de proyectos vecinales: Los vecinos registrados podrán postular proyectos comunitarios, que serán gestionados y evaluados por el directorio de la junta.
* Reserva de espacios comunitarios: A través de un calendario, los vecinos podrán reservar canchas, salas o plazas, seleccionando fechas y horarios disponibles.
* Publicación de noticias y avisos: Los administradores podrán publicar noticias y notificaciones que los vecinos podrán consultar desde la plataforma o recibir vía email o WhatsApp.
  1. **Características de los usuarios**

Los usuarios del sistema se dividen en dos grandes grupos:

* Vecinos: Usuarios que se inscriben en la junta de vecinos, realizan solicitudes de certificados, postulan proyectos y reservan espacios. Estos usuarios no requieren habilidades técnicas avanzadas, solo conocimientos básicos para usar un navegador web. Se asume que la mayoría de estos usuarios tiene un nivel educacional medio, con experiencia básica en el uso de aplicaciones web y móviles.
* Administradores/Directorio: Usuarios responsables de revisar y aprobar solicitudes, gestionar la plataforma y mantener el contenido actualizado. Se espera que estos usuarios tengan habilidades intermedias en el manejo de sistemas web, con capacidad para gestionar información sensible de forma segura.
  1. **Restricciones**

El desarrollo del sistema estará sujeto a las siguientes restricciones:

* Políticas de la empresa: El sistema debe cumplir con las regulaciones locales y políticas internas de las juntas de vecinos, especialmente en lo relacionado con el manejo de datos personales de los vecinos.
* Limitaciones del hardware: El sistema está diseñado para ser accesible desde dispositivos con capacidades básicas de procesamiento, tanto móviles como de escritorio. No se requiere hardware especializado.
* Interfaces con otras aplicaciones: El sistema deberá interactuar con plataformas de mensajería como WhatsApp para el envío de notificaciones y con servicios de correo electrónico para la comunicación con los vecinos.
* Operaciones paralelas: El sistema permitirá la gestión simultánea de múltiples solicitudes y reservas sin interferencias entre los usuarios.
* Lenguajes de programación: El desarrollo se realizará en Python con Django, utilizando HTML, CSS con Bootstrap para la interfaz de usuario, y JavaScript para las funciones interactivas.
* Seguridad: El sistema deberá garantizar la seguridad de los datos personales de los usuarios, implementando medidas de autenticación segura y almacenamiento encriptado de la información sensible.
  1. **Suposiciones y Dependencias**

El desarrollo del sistema dependerá de los siguientes factores:

* Se asume que el sistema correrá en servidores que soportan Django y base de datos PostgreSQL.
* Se requiere que los usuarios tengan acceso a dispositivos con conexión a Internet para interactuar con la plataforma.
* Se asume que los administradores están capacitados en el uso básico de la plataforma.

Cualquier cambio en estos factores, como un cambio en el entorno de servidor o en los requisitos de hardware, podría requerir una revisión de los requisitos del sistema.

* 1. **Requisitos Futuros**

Se identifican las siguientes posibles mejoras que podrían implementarse en futuras versiones del sistema:

* Integración de redes sociales: Permitirá compartir noticias y eventos de la junta directamente en plataformas como Facebook o Instagram.
* Sistema de votaciones: Para permitir que los vecinos participen en decisiones comunitarias a través de votaciones en línea.
* Notificaciones push: Enviar notificaciones directamente a los dispositivos móviles de los vecinos.
* Mejoras en la accesibilidad: Implementación de funciones para garantizar que el sistema sea accesible para personas con discapacidades visuales o motrices.

1. **Requisitos Específicos**

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describe comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

* El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
* Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
* Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
* Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:
  + **Corrección**: La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
  + **No ambiguos:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
  + **Completos:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
  + **Consistentes:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorios no es implementable.
  + **Clasificados:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, por no emplear excesivos recursos en implementar requisitos no esenciales.
  + **Verificables:** La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable.Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos.  
    Requisitos como “en caso de accidente la nube tóxica no se extenderá más allá de 25Km" no es verificable por el alto costo que conlleva.
  + **Modificables:** La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos facilitan enormemente esta tarea.
  + **Trazables:** La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica qué componentes del sistema son los que realizan el requisito R

* 1. **Requisitos de las Interfaces**

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

* + 1. **Interfaces de Usuario**

El sistema contará con una interfaz web responsiva, diseñada en HTML, CSS (Bootstrap) y JavaScript. Los requisitos clave incluyen:

* Estilo y colores: La interfaz debe seguir un esquema de colores institucionales definidos por cada junta de vecinos. Se utilizarán colores suaves (azules y grises) para evitar la fatiga visual de los usuarios.
* Pantallas principales:
  + Pantalla de inicio: Mostrar un resumen de noticias recientes, enlaces a módulos (solicitud de certificados, postulación de proyectos, reserva de espacios).
  + Formulario de registro de vecinos: Incluir campos de nombres, apellidos, dirección, teléfono, fecha de nacimiento, y los archivos requeridos (foto de cédula de identidad y comprobante de residencia).
  + Gestión de solicitudes: Los administradores deben poder ver un listado de solicitudes con la opción de aprobar o rechazar cada una.
  + Calendario de reserva de espacios: Un calendario interactivo que permita seleccionar fechas y ver las horas disponibles para la reserva.
    1. **Interfaces de Hardware**

El sistema no requiere hardware especializado. Está diseñado para ejecutarse en servidores web estándar y será accesible desde dispositivos móviles, tabletas y computadoras personales con capacidad básica de procesamiento.

* + 1. **Interfaces de Software**
* Correo electrónico: Para enviar notificaciones a los vecinos sobre el estado de sus solicitudes.
  + Propósito: Enviar correos automáticos en caso de aprobaciones, rechazos o actualizaciones de las solicitudes.
  + Interfaz: Uso de SMTP para enviar correos, compatible con servicios como Gmail o proveedores empresariales.
* WhatsApp API: Para enviar notificaciones rápidas y recordatorios de eventos a los vecinos.
  + Propósito: Notificar a los vecinos sobre nuevas noticias, fechas de eventos o actualizaciones de reservas.
  + Interfaz: Uso de la API de WhatsApp Business para enviar mensajes.
* Portal de Pagos: Para habilitar el arriendo de los espacios comunitarios.
  + Propósito: Tener un comprobante de pago de que el espacio está ocupado a través de transacciones en línea.
  + Interfaz: Uso de la API de MercadoPago para transacciones.
    1. **Interfaces de Comunicación**

El sistema debe usar HTTPS para asegurar la comunicación entre los usuarios y el servidor. Además, se implementará autenticación segura para los administradores, utilizando OAuth o un sistema de autenticación propio.

* 1. **Requisitos Funcionales**

Los requisitos funcionales describen las acciones que el sistema debe realizar para cumplir con sus objetivos.

**Requisito Funcional 1: Registro de Vecinos**

* Entrada: Los usuarios proporcionan sus datos personales, como nombres, apellidos, dirección y teléfono, y cargan los archivos requeridos.
* Proceso: Validación de los datos (formato de teléfono, fechas válidas) y verificación de que los archivos están completos.
* Salida: Confirmación de registro exitoso o mensajes de error si falta información o los datos son incorrectos.

**Requisito Funcional 2: Gestión de Solicitudes de Certificados**

* Entrada: Los vecinos solicitan un certificado de residencia cargando la información requerida.
* Proceso: Los administradores revisan la solicitud, verifican los documentos adjuntos y deciden si aprobar o rechazar la solicitud.
* Salida: Un correo electrónico y mensaje de WhatsApp se envían al vecino notificándole si su solicitud fue aprobada o rechazada.

**Requisito Funcional 3: Reserva de Espacios Comunitarios**

* Entrada: Los vecinos seleccionan una fecha en el calendario y eligen una hora para reservar un espacio (cancha, sala, etc.).
* Proceso: El sistema verifica la disponibilidad del espacio en la fecha seleccionada y confirma la reserva si está disponible.
* Salida: Confirmación de la reserva o mensaje de error si el espacio ya está reservado.

**Requisito Funcional 4: Postulación de Proyectos Vecinales**

* Entrada: Los vecinos completan un formulario con los detalles de su proyecto comunitario y adjuntan un archivo con la propuesta.
* Proceso: Los administradores revisan las postulaciones y deciden aprobar o rechazar el proyecto.
* Salida: Notificación de la resolución a través de correo y WhatsApp*.*
  1. **Requisitos no Funcionales**
     1. **Requisitos de Rendimiento**
* El sistema debe ser capaz de gestionar hasta 500 usuarios simultáneos sin pérdida significativa de rendimiento.
* Los tiempos de carga no deben superar los 3 segundos para cualquier página o funcionalidad del sistema.
  + 1. **Seguridad**
* Acceso: Solo los administradores tendrán acceso a la gestión de solicitudes y reservas.
* Protección de datos: Toda la información personal y los archivos subidos por los vecinos estarán encriptados en la base de datos.
* Autenticación: Los usuarios deberán autenticarse mediante nombre de usuario y contraseña seguros, con una longitud mínima de 8 caracteres.
  + 1. **Fiabilidad**
* El sistema debe estar disponible el 99.9% del tiempo, con solo un margen de 0.1% de tiempo de inactividad programado para mantenimiento.
  + 1. **Disponibilidad**
* El sistema debe estar disponible para los usuarios las 24 horas del día, 7 días a la semana con una disponibilidad mínima del 99.9% anual.
* Cualquier tiempo de inactividad planificado para mantenimiento debe ser notificado a los usuarios con al menos 48 horas de anticipación.
  + 1. **Mantenibilidad**
* Las tareas de mantenimiento, como la limpieza de registros antiguos y el monitoreo de la base de datos, se realizarán semanalmente.
  + 1. **Portabilidad**
* El sistema debe ser fácilmente desplegable en cualquier servidor web que soporte Django y Firebase Database, asegurando compatibilidad multiplataforma.
  1. **Otros requisitos**
* Internacionalización: El sistema debe ser adaptable para múltiples idiomas, con énfasis en español.
* Accesibilidad: El sistema debe ser accesible para usuarios con discapacidades visuales, cumpliendo con los estándares WCAG 2.1.
  1. **Requisitos de Base de Datos Lógica.**
* Vecinos: Almacenar nombres, apellidos, dirección, teléfono, fecha de nacimiento, y archivos adjuntos.
* Solicitudes: Almacenar las solicitudes de certificados y el estado de cada una.
* Reservas: Almacenar fechas y horas de reserva de espacios comunitarios.
* Noticias: Almacenar título, contenido, fecha de publicación e imagen

1. **Propuesta de Planificación**
   1. **Descripción general acerca de la planificación**

La planificación del proyecto abarcará un período de X días con la participación de un equipo multidisciplinario. Se estima que cada fase tendrá una duración aproximada de X semanas. Se aplicarán buenas prácticas de gestión de proyectos, como el uso de metodologías y reuniones de seguimiento semanales. Las condiciones necesarias para un buen término incluyen la asignación clara de roles y responsabilidades, la correcta gestión de riesgos y el cumplimiento de los plazos establecidos.

* + 1. **Definición del Equipo de Trabajo**

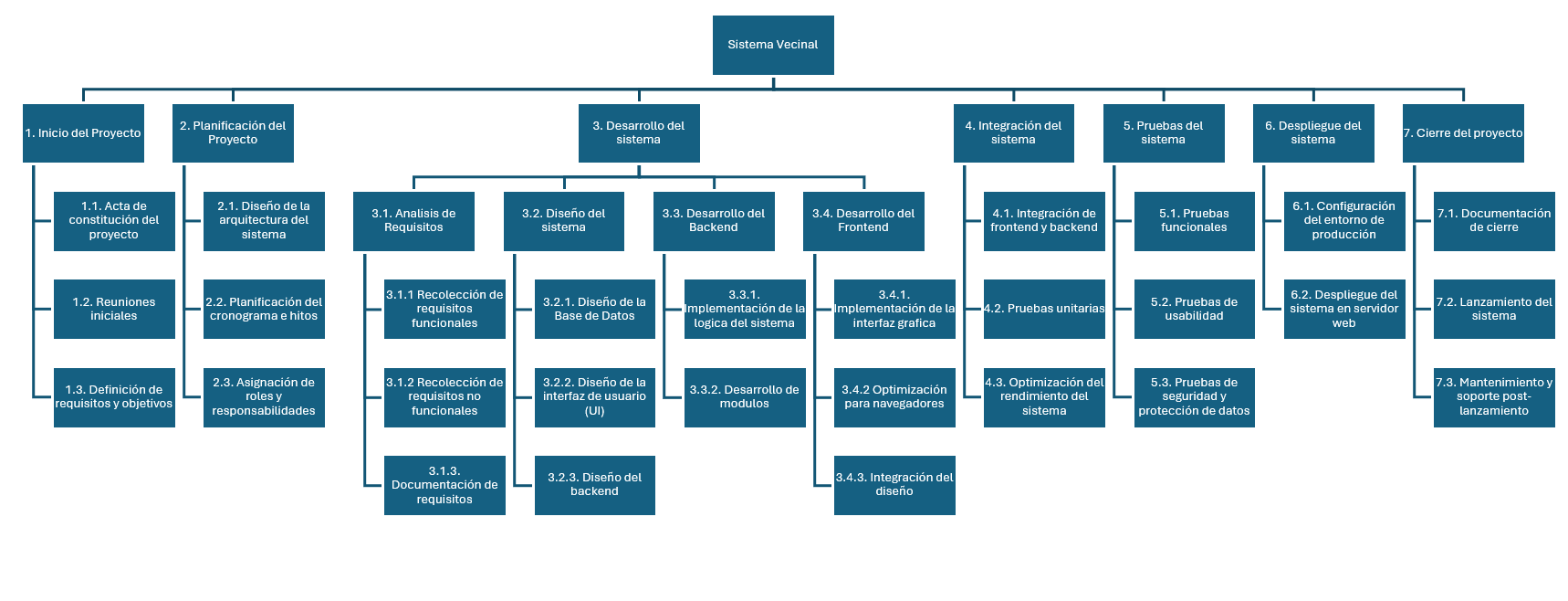
| **Recurso** | **Departamento / División** |
| --- | --- |
| **Lider de proyecto** | **Gestión del Proyecto** |
| **Desarrollador** | **Desarrollo del sistema** |
| **Tester** | **Control de Calidad** |
| **Analista** | **Analisis de negocio** |
| **Diseñador UI/UX** | **Diseño** |
| **Soporte Tecnico** | **Soporte y operaciones** |

* + 1. **Definición de Actividades principales del Proyecto**

Las principales actividades del proyecto son:

* Recolección de requisitos: Definir y documentar todos los requisitos funcionales y no funcionales.
* Diseño del Sistema: Esquematización de la arquitectura y la interfaz de usuario.
* Desarrollo: Implementación del sistema de acuerdo a los requisitos.
* Pruebas y Validación: Asegurarse de que el sistema cumple con los requisitos.
* Despliegue: Implementación en el entorno de producción.
* Mantenimiento: Resolución de incidencias y futuras mejoras.

* + 1. **Diagrama EDT**



* + 1. **Carta Gantt**

| **Actividad** | **Fase 1** | | | | **Fase 2** | | | | | | | | | | | | **Fase 3** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S 1** | **S 2** | **S 3** | **S 4** | **S 5** | **S 6** | **S 7** | **S 8** | **S 9** | **S 10** | **S 11** | **S 12** | **S 13** | **S 14** | **S 15** | **S 16** | | **S 17** | **S 18** |
| *Análisis de Requisitos* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Diseño de la arquitectura del sistema* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Diseño y prototipado de la Interfaz Grafica del Usuario* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Desarrollo de la lógica del sistema* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Implementación de la interfaz de usuario* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Integración de la interfaz de usuario con la lógica del sistema* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Ejecución de pruebas* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Despliegue de entorno de producción* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Planificación de mantenimiento y soporte* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |

* + 1. **Resumen de Costos del Desarrollo del Proyecto**

| **Costos**  *Definición de costos en recursos tecnológicos y recursos humanos.* |
| --- |
| El desarrollo del sistema web para la gestión de las juntas de vecinos implica los siguientes costos teniendo en cuenta tanto los recursos tecnológicos como los recursos humanos:  **Recursos tecnológicos:**   * Firebase y servicios en la nube: Teniendo un uso moderado de esta base de datos el costo de la utilización de esta base de datos para el proyecto sería de $2680/mes aproximadamente. * Hosting y Dominio: Los costos del dominio web serían aproximadamente de $10.000/año y en el hosting web usando un plan básico sería aproximadamente de unos $7.500/mes. * Computador: En caso de no contar con un computador para el desarrollo el costo de este consta desde los $500.000 hasta los $800.000.   **Recursos humanos:**  Desarrolladores: Teniendo en cuenta el tiempo de desarrollo y que se necesitan 3 desarrolladores para la realización de la página el costo sería de aproximadamente $2.650.000 por cada desarrollador.  Diseñador web: El coste para un diseñador web sería de aproximadamente $967.500  QA: El coste sería aproximadamente de $1.200.000.  Soporte y mantenimiento: Para el soporte y mantenimiento que se necesita dar a la página web sería de unos $150.000/mes. |

* 1. **Plan de control de Cambio**

Los cambios durante el desarrollo del proyecto podrán clasificarse en los siguientes tipos:

* Cambios Funcionales: Incluyen modificaciones a las funcionalidades previamente definidas.
* Cambios No Funcionales: Incluyen mejoras en rendimiento, seguridad o fiabilidad del sistema.
* Cambios Correctivos: Respuesta a errores o fallos detectados durante la fase de pruebas o post-despliegue.

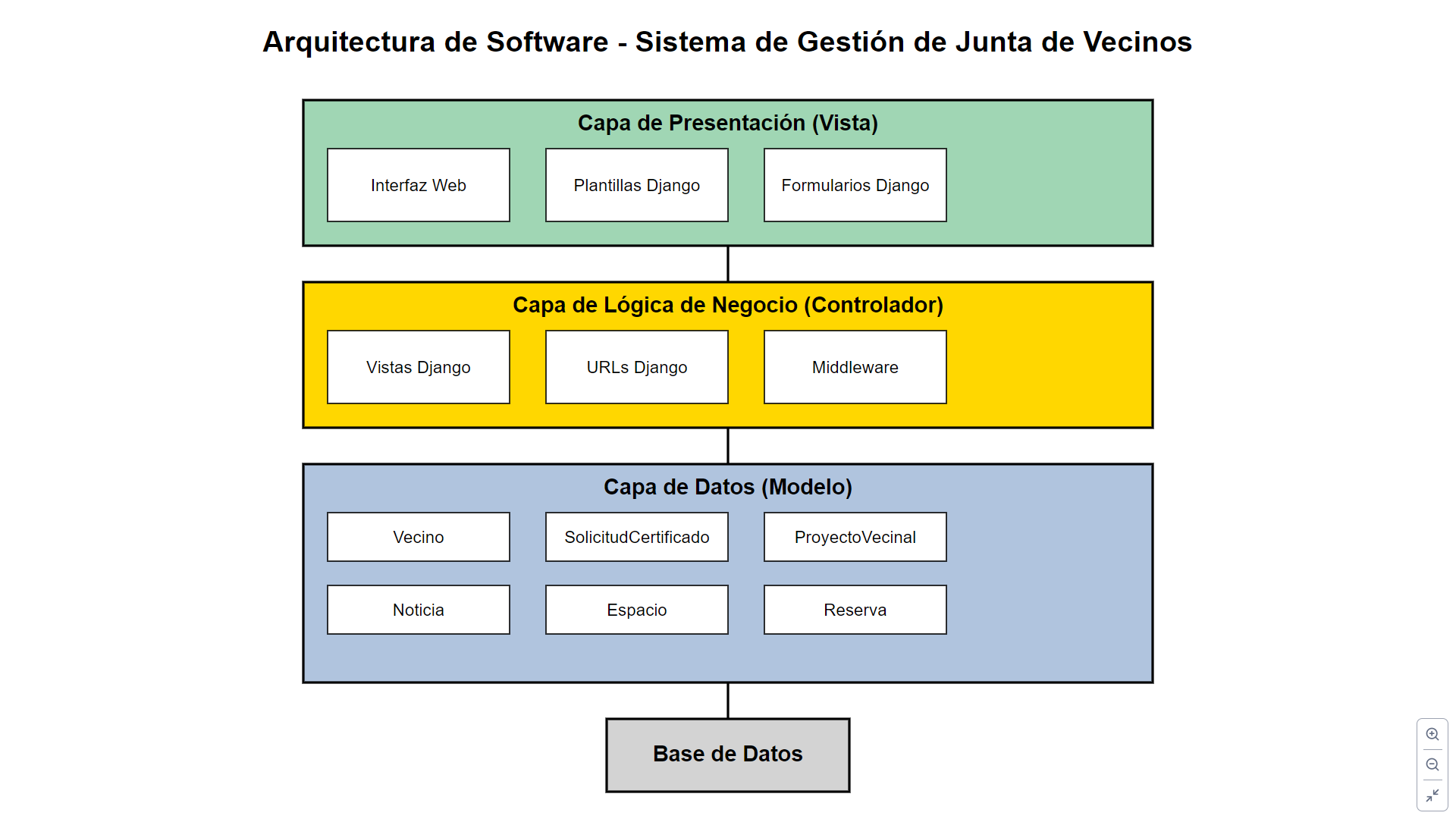
Tabla de control de cambios

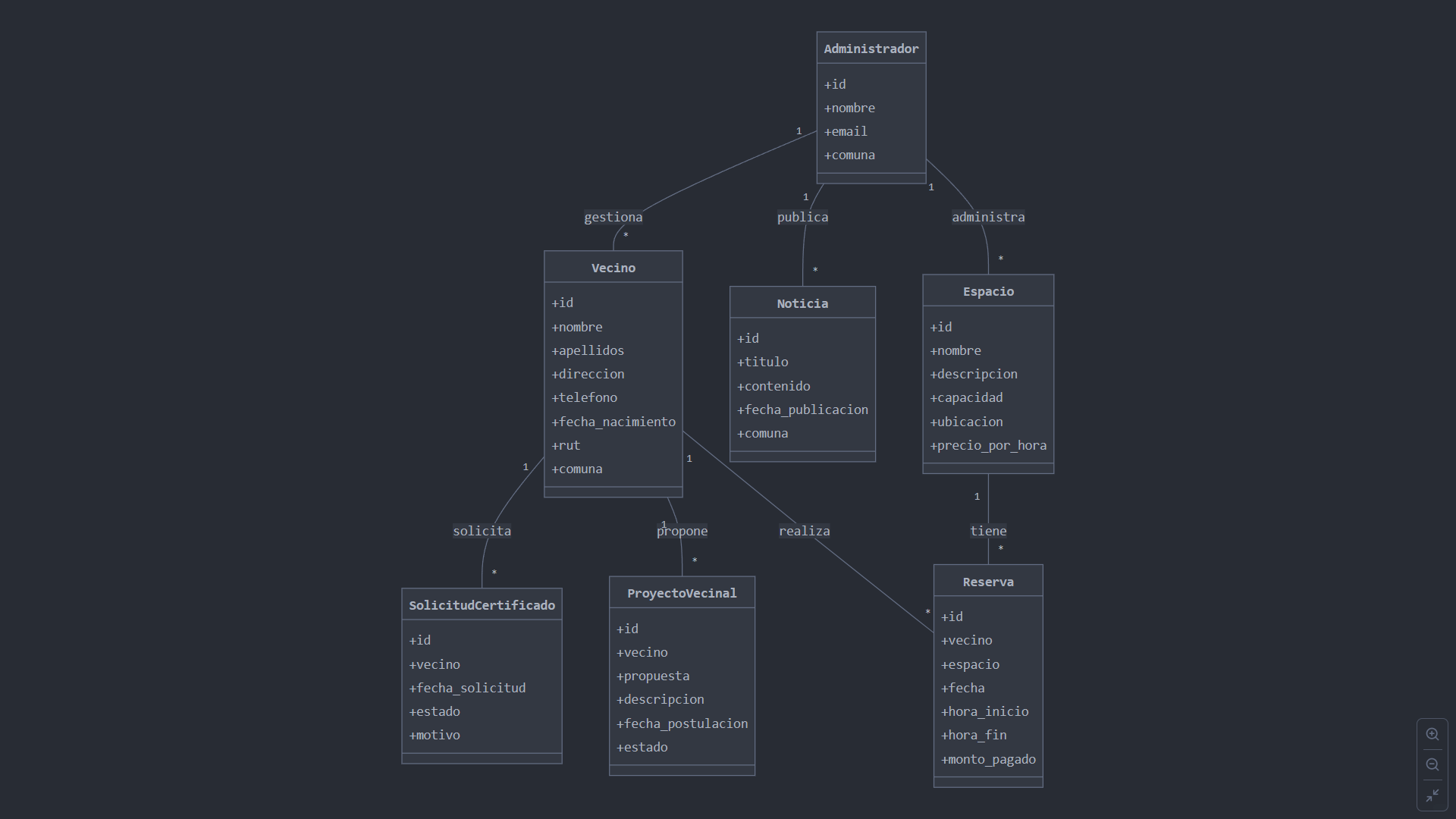
| Tipo de Cambio | Descripción | Alcance | Limitaciones | Aprobación |
| --- | --- | --- | --- | --- |
| Funcional | Modificación de una funcionalidad | Afecta a todo el sistema | Limitado a fases tempranas | Gerente de Proyecto |
| No Funcional | Mejora de rendimiento | Solo afecta al backend | Solo si no afecta al rendimiento de producción | Equipo de desarrollo |
| Correctivo | Corrección de errores | Impacto en un módulo | Limitado a fase de pruebas | Tester líder |

Se permitirán cambios en cualquier fase, siempre que cumplan con los criterios establecidos de necesidad y viabilidad. En caso de cambios significativos, se evaluará su impacto en el cronograma y el costo total del proyecto.

1. **Anexos**

**Diagrama MVC (Modelo - Vista - Controlador)**

****

**Modelo de la base de datos**

**Mockups**

****

****

****