**[Sistema Vecinal]**

**(SAD) Software Architecture Document**

**Versión 1.0**

* **Tabla de Contenidos**

Contenido

[**● Tabla de Contenidos 0**](#_heading=h.gkp737h9219k)

[**● Identificación de Documento 2**](#_heading=h.kvyhq8ffddmh)

[**● Historia de cambios 2**](#_heading=h.42sj4ucaubjt)

[**● Índice de figuras 3**](#_heading=h.3znysh7)

[**● Introducción 4**](#_heading=h.2s8eyo1)

[**● Alcance 4**](#_heading=h.zet794er9cuo)

[**● Referencias 4**](#_heading=h.3rdcrjn)

[**● Arquitectura de Software 4**](#_heading=h.26in1rg)

[**● Objetivos y Restricciones de la Arquitectura 4**](#_heading=h.lnxbz9)

[**● Diagramas 4**](#_heading=h.35nkun2)

[**● Tamaño y desempeño 4**](#_heading=h.xh4j8fds8gvf)

* Identificación de Documento

| **Identificación** | DAS |
| --- | --- |
| **Proyecto** | Junta de Vecinos |
| **Versión** | 2.0 |

| **Documento mantenido por** | Victor Manuel Navarro Fermín |
| --- | --- |
| **Fecha de última revisión** | 15-09-2024 |
| **Fecha de próxima revisión** | 16-09-2024 |

| **Documento aprobado por** |  |
| --- | --- |
| **Fecha de última aprobación** |  |

* **Historia de cambios**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 15-09-2024 | 1.0 | Primera Revisión | Victor Navarro |
| 09-10-2024 | 2.0 | Revisión Final | Victor Navarro |
|  |  |  |  |
|  |  |  |  |

* **Índice de figuras**
* Diagrama de Arquitectura MVC (Modelo-Vista-Controlador)
* Diagrama de Secuencia
* Diagrama de Despliegue

* **Introducción**

Este Documento de Arquitectura de Software (DAS) describe la arquitectura elegida para el desarrollo del sistema web de gestión de una junta de vecinos. El propósito del documento es proporcionar una visión detallada de los componentes arquitectónicos clave, las decisiones de diseño, y cómo se implementará la arquitectura MVC en el proyecto para garantizar su éxito.

* **Alcance**

El alcance de este documento se centra en describir la arquitectura del sistema de gestión para una junta de vecinos, utilizando el framework Django y basado en la arquitectura MVC. Se detallan los componentes que soportan las funcionalidades de inscripción de vecinos, manejo de certificados, gestión de proyectos vecinales, reservas de espacios comunitarios, y publicación de noticias. No se incluyen integraciones con sistemas externos o aplicaciones móviles específicas en este alcance inicial.

* **Referencias**
* **Arquitectura de Software**

La arquitectura seleccionada para el proyecto es el patrón Modelo-Vista-Controlador (MVC). El modelo se encargará de gestionar los datos y la lógica de negocio, que incluyen los vecinos, solicitudes de certificados, postulaciones a proyectos, noticias y reservas de espacios. Las vistas serán la interfaz gráfica, implementada en HTML, CSS y Bootstrap, presentando al usuario la información de manera clara y accesible. El controlador gestionará la interacción entre el usuario y el sistema, recibiendo las solicitudes y actualizando el modelo según corresponda.

* **Objetivos y Restricciones de la Arquitectura**

**Objetivos:**

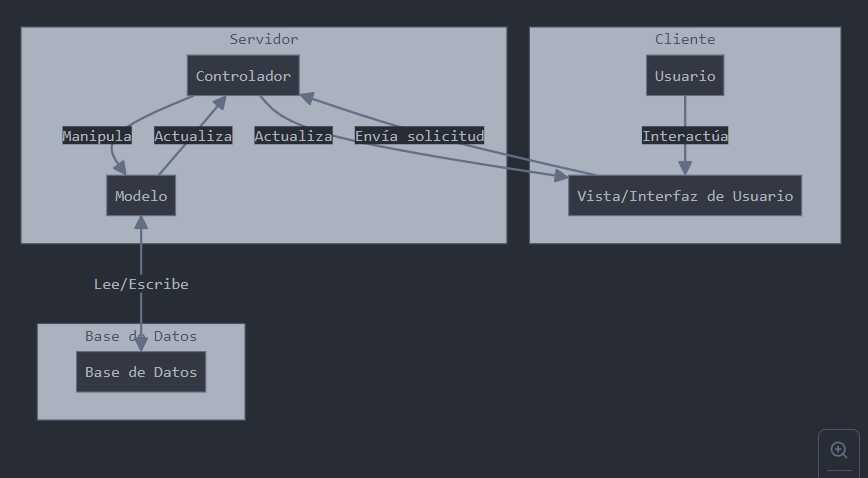
* Implementar una arquitectura escalable y mantenible.
* Permitir la fácil integración de nuevas funcionalidades, como notificaciones por WhatsApp.
* Asegurar la seguridad de los datos de los usuarios y la integridad del sistema.

**Restricciones:**

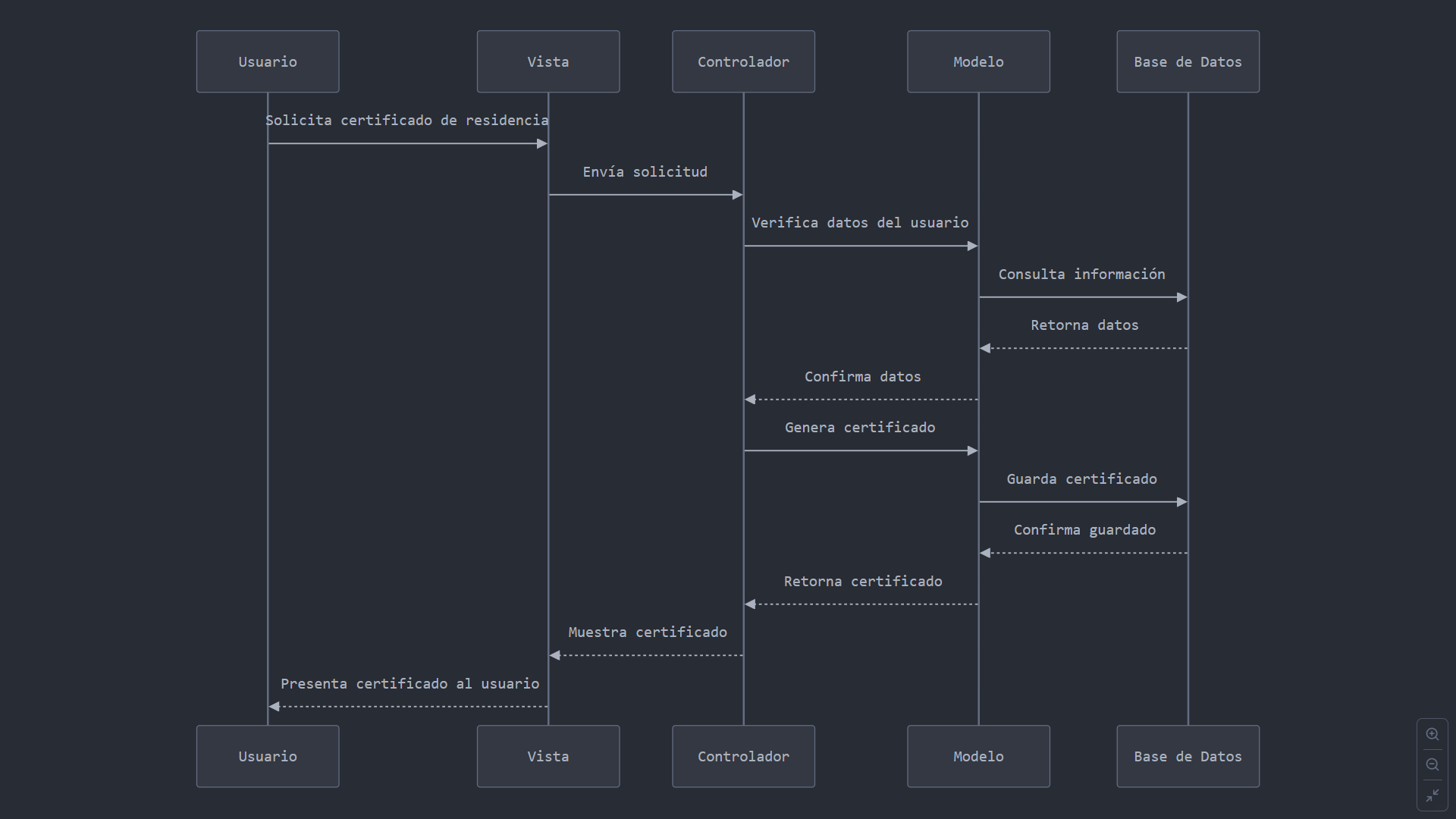
* El sistema debe funcionar en entornos web responsivos.
* Limitación en el tiempo de desarrollo (aproximadamente 4 meses).
* Uso de tecnologías predefinidas: Python, Django, HTML, CSS, Bootstrap.

* **Diagramas**

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

****

Este diagrama muestra la estructura básica de la arquitectura MVC de tu sistema. El usuario interactúa con la Vista, que envía solicitudes al Controlador. El Controlador manipula el Modelo, que a su vez interactúa con la Base de Datos. El Controlador luego actualiza la Vista según sea necesario.

* Diagrama de Secuencia (Solicitud de certificado)

Este diagrama de secuencia muestra el proceso de solicitud y emisión de un certificado de residencia, que es una de las funcionalidades clave de tu sistema. Ilustra cómo interactúan los diferentes componentes del sistema para procesar esta solicitud.

* Diagrama de Despliegue

Este diagrama de despliegue muestra cómo se distribuyen los diferentes componentes de tu sistema en la infraestructura física. Incluye los dispositivos de usuario (navegador web y aplicación móvil), el servidor web (que contiene el servidor de aplicaciones y el servidor de base de datos), y los servicios externos para correo y WhatsApp.

* **Tamaño y desempeño**

El sistema está diseñado para soportar la inscripción de hasta 10,000 vecinos, con la capacidad de gestionar simultáneamente hasta 500 solicitudes de certificados o proyectos vecinales. Se espera que el sistema maneje una carga moderada, con un promedio de 100 a 200 usuarios simultáneos. Django permitirá escalar el sistema si se requiere en un futuro, ya sea incrementando los recursos del servidor o distribuyendo la carga.