

# **PROGRAMA DE SOLICITUDES DE COMUNICACIÓN BIDIRECCIONAL**

---

Julián Alexis Irigoyen Valles

Proyecto Final Thincrs

**15 de agosto de 2023**

## **1. \*\*Resumen Ejecutivo\*\***

- Breve descripción del proyecto y sus objetivos.

El proyecto en cuestión es una página de solicitudes de la ciudadanía, simulando a una página de gobierno. Los ciudadanos son capaces de crear solicitudes sobre quejas, problemas y cosas que les conciernen dentro de su comunidad, y tienen la posibilidad que los servidores públicos respondan sus cuestiones.

- Justificación del proyecto y su importancia.

Este proyecto es importante debido a que no siempre es posible para un ciudadano, comunicarse con alguien que pueda hacer algo al respecto de ciertas cuestiones de la vida cotidiana. Lo que este proyecto promueve, es la comunicación entre ciudadanos y servidores públicos

- Alcance general del proyecto.

El alcance del proyecto abarca el diseño, desarrollo e implementación de una plataforma en línea denominada "Solicitud Ciudadana".

## **2. \*\*Introducción\*\***

- Contexto del proyecto: Problema a resolver o necesidad a cubrir.

La falta de un canal de comunicación eficaz entre los ciudadanos y los servidores públicos ha llevado a situaciones en las que los problemas y quejas de la comunidad a menudo quedan sin respuesta o se abordan de manera ineficiente. Esto resulta en una desconexión entre los ciudadanos y sus representantes, lo que puede dar lugar a la insatisfacción generalizada y a la falta de resolución de problemas fundamentales en la comunidad.

- Beneficios esperados del software.

La implementación de este proyecto ofrecerá a los ciudadanos plantear sus inquietudes de manera rápida y directa, lo que fomentará la participación y la

colaboración en la mejora de la comunidad. Los servidores públicos, por su parte, contarán con una plataforma centralizada para recibir, gestionar y responder a las solicitudes, lo que resultará en una administración más eficiente y una mayor transparencia en sus acciones.

- Audiencia principal del software.

La audiencia principal del proyecto incluye a dos grupos clave: los ciudadanos y los servidores públicos.

### **3. \*\*Objetivos del Proyecto\*\***

- Enumeración de los objetivos específicos y medibles del proyecto.

1. Facilitar la creación de solicitudes por parte de los ciudadanos.
2. Agilizar la respuesta de los servidores públicos a las solicitudes.
3. Mejorar la transparencia gubernamental mediante el seguimiento de solicitudes.
4. Centralizar la gestión de solicitudes para evitar duplicación de esfuerzos.
5. Impulsar la participación ciudadana en la resolución de problemas locales.
6. Generar informes para obtener información valiosa sobre problemas recurrentes.

- Cómo los objetivos se alinean con las necesidades del cliente y del negocio.

Los objetivos del proyecto están directamente alineados con las necesidades tanto de los ciudadanos como del negocio gubernamental. Desde la

perspectiva de los ciudadanos, el proyecto aborda la necesidad de tener una forma efectiva y accesible de comunicarse con las autoridades gubernamentales para resolver problemas y mejorar su calidad de vida.

Desde la perspectiva del negocio gubernamental, el proyecto responde a la necesidad de mejorar la eficiencia en la gestión de solicitudes, lo que a su vez aumenta la satisfacción de los ciudadanos y fortalece la relación entre el gobierno y la comunidad. Además, la plataforma proporcionará datos valiosos que pueden ser utilizados para tomar decisiones informadas y mejorar los servicios públicos en general.

#### **4. \*\*Alcance del Proyecto\*\***

- Descripción detallada de las funcionalidades y características que el software incluirá.

##### **1. Registro de Usuarios:**

- Los ciudadanos podrán crear cuentas personales en la plataforma utilizando información básica y segura.
- Se implementará un proceso de verificación para garantizar la autenticidad de los usuarios.

##### **2. Creación de Solicitudes:**

- Los usuarios podrán redactar y enviar solicitudes detalladas sobre problemas y preocupaciones en su comunidad.
- Las solicitudes contendrán información específica, como la descripción del problema y la ubicación.

##### **3. Seguimiento en Tiempo Real:**

- Los ciudadanos podrán rastrear el estado de sus solicitudes en tiempo real, desde su envío hasta su resolución.
- Los servidores públicos proporcionarán actualizaciones a medida que avance la resolución de las solicitudes.

#### 4. Asignación y Gestión de Solicitudes:

- Los servidores públicos podrán ver y gestionar las solicitudes asignadas a ellos.

#### 5. Comunicación Bidireccional:

- Los ciudadanos podrán recibir respuestas y comentarios de los servidores públicos en relación con sus solicitudes.
- Los servidores públicos podrán solicitar información adicional o proporcionar actualizaciones directamente a los ciudadanos.

#### 6. Generación de Informes:

- Se generará un registro histórico de solicitudes, incluyendo detalles sobre su estado y resolución.
- Los informes periódicos permitirán identificar patrones, problemas recurrentes y áreas de mejora.

#### - Límites y restricciones del proyecto.

1. No es una Plataforma Gubernamental Oficial: Este proyecto es una simulación y no reemplaza los canales oficiales de comunicación gubernamental.
2. No Incluye Aspectos Legales: El proyecto no abarcará cuestiones legales o procesos judiciales relacionados con las solicitudes.

3. No se Garantiza la Resolución de Problemas: Aunque el proyecto busca mejorar la comunicación, no garantiza la resolución inmediata de todos los problemas planteados.
4. Requisitos Técnicos Limitados: El software estará diseñado para ser accesible, pero no puede garantizarse en todos los dispositivos o conexiones de red.

## **5. \*\*Arquitectura y Diseño\*\***

- Diseño de alto nivel de la arquitectura del software.

El diseño de la arquitectura del software del proyecto se basa en una estructura modular y escalable para garantizar la eficiencia, seguridad y mantenibilidad del sistema. Además, las vistas son simples, fáciles de entender para el público general y accesibles.

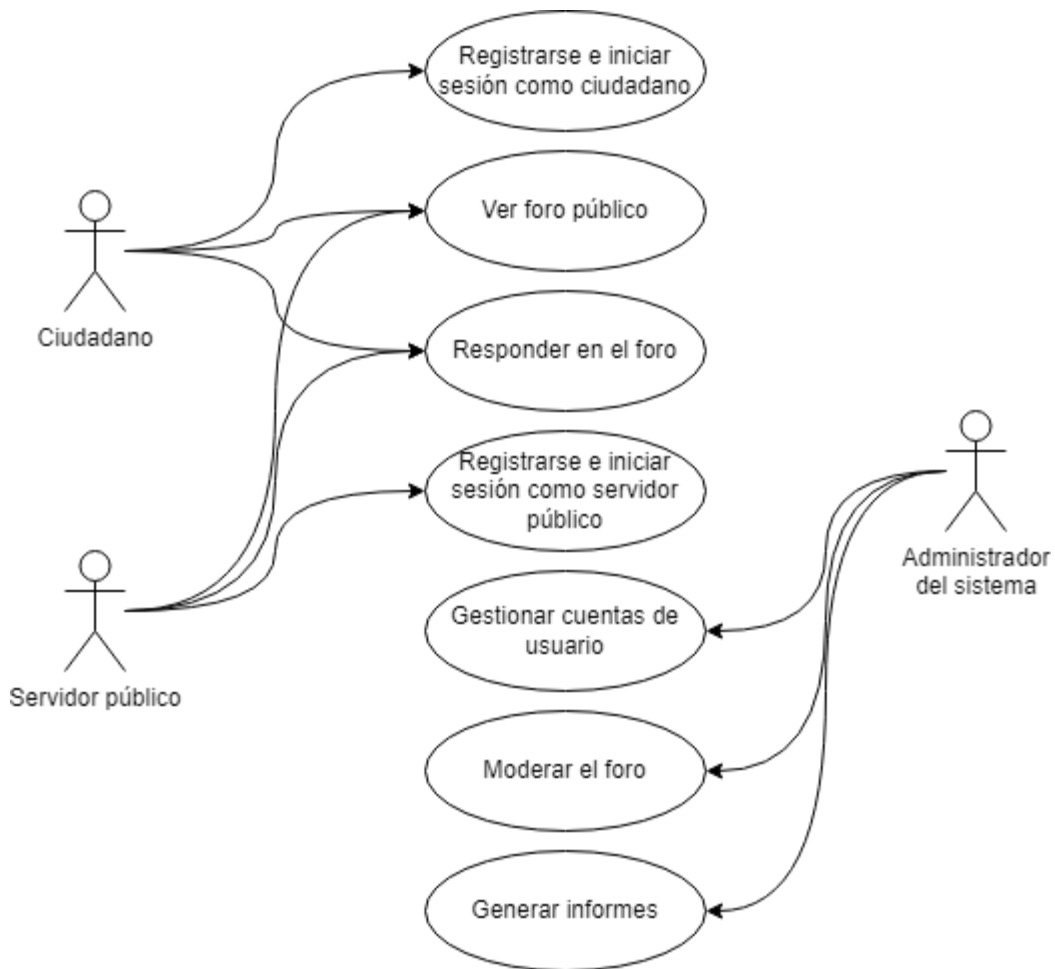
La arquitectura del software seguirá un enfoque Modelo-Vista-Controlador (MVC), que separa claramente la lógica de negocio, la presentación y el control de las interacciones.

- Tecnologías y herramientas a utilizar.

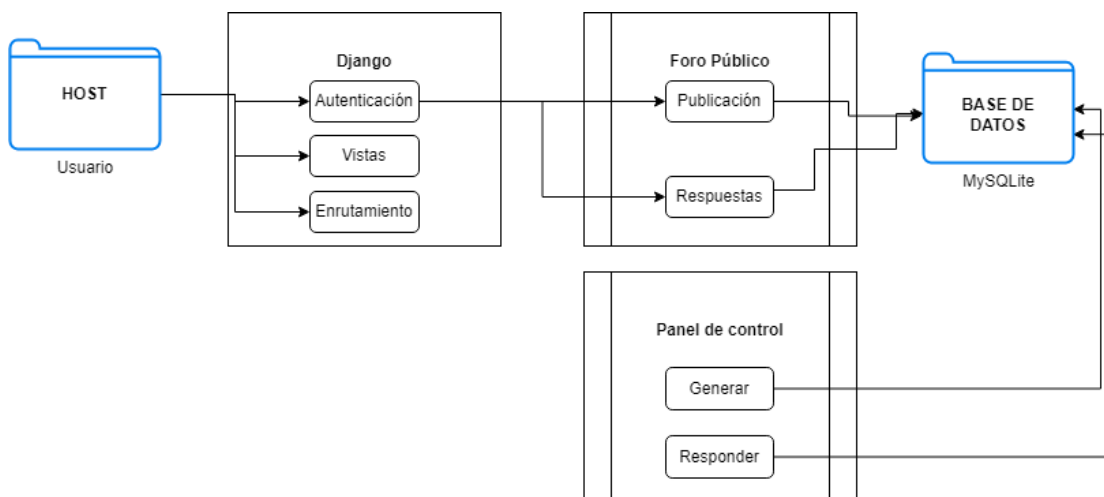
1. Lenguaje de programación: Python
2. Framework: Django
3. Vistas y diseño: HTML, CSS y Bootstrap
4. Base de datos: SQL

- Diagramas de flujo, diagramas de clases u otros artefactos de diseño.

### Diagrama de casos de uso



### Diagrama de componentes



## 6. **\*\*Desarrollo\*\***

- Metodología de desarrollo a utilizar (ágil, cascada, etc.).

Dado que el proyecto fue realizado por una sola persona y se utilizó un enfoque de desarrollo individual, no se siguió una metodología formal como las metodologías ágiles o en cascada.

- Cronograma de desarrollo con hitos clave.

### *Fase de Preparación:*

- Definición de requisitos y alcance.
- Creación de los diagramas de diseño.
- Creación de plan maestro.

### *Fase de Desarrollo:*

- Creación de la estructura de la base de datos.
- Implementación de la autenticación de usuarios.
- Desarrollo de la creación y visualización de solicitudes.
- Integración de Bootstrap para el diseño de la interfaz.

### *Fase de Pruebas:*

- Realización de pruebas unitarias y de integración.
- Identificación y corrección de errores.
- Pruebas de usabilidad.

### *Fase de Finalización:*

- Creación de informes y generación de historiales de solicitudes.
- Ajustes finales de diseño y presentación.
- Evaluación general y revisión de la funcionalidad.



- Asignación de tareas y responsabilidades.

- Diseño de la Interfaz de Usuario.
- Desarrollo del Sistema de Registro y Autenticación.
- Implementación de la Creación y Visualización de Solicitudes.
- Integración de Bootstrap.
- Creación de Informes y Generación de Historiales.

## **7. \*\*Pruebas y Validación\*\***

- Plan de pruebas detallado.

### **1. Pruebas Unitarias:**

- Evaluar individualmente cada componente del sistema para asegurar que funcionan correctamente.

### **2. Pruebas de Integración:**

- Verificar que los componentes se integran adecuadamente y que la comunicación entre ellos es efectiva.

### **3. Pruebas de Funcionalidad:**

- Realizar pruebas de cada funcionalidad, asegurando que cumpla con los requisitos.

### **4. Pruebas de Interfaz de Usuario:**

- Evaluar la usabilidad y el diseño de la interfaz de usuario, incluida la respuesta a diferentes tamaños de pantalla.

### **5. Pruebas de Comunicación Bidireccional:**

- Verificar que la comunicación entre los ciudadanos y los servidores públicos funciona correctamente.

## 6. Pruebas de Rendimiento:

- Evaluar cómo responde la plataforma ante cargas de trabajo variables y asegurarse de que se mantenga la eficiencia.

## 7. Pruebas de Seguridad:

- Identificar y mitigar posibles vulnerabilidades.

### - Escenarios de prueba y casos de prueba.

#### 1. Registro y Autenticación:

- Escenario: Un usuario se registra en la plataforma.
- Caso de Prueba: Verificar que el usuario puede completar el proceso de registro y acceder a su cuenta.

#### 2. Creación de Solicitudes:

- Escenario: Un ciudadano crea una nueva solicitud.
- Caso de Prueba: Verificar que la solicitud se almacena en la base de datos y se muestra en la interfaz de usuario.

#### 3. Seguimiento de Solicitudes:

- Escenario: Un usuario verifica el estado de una solicitud.
- Caso de Prueba: Verificar que el estado de la solicitud se muestra correctamente en función de su estado actual.

#### 4. Comunicación Bidireccional:

- Escenario: Un ciudadano recibe una respuesta de un servidor público.
- Caso de Prueba: Verificar que la respuesta se muestra de manera adecuada y se puede visualizar.

- Criterios de aceptación.

- La plataforma debe permitir el registro exitoso de usuarios.
- Las solicitudes creadas por los usuarios deben almacenarse en la base de datos.
- Los ciudadanos deben poder rastrear el estado de sus solicitudes con precisión.

## **8. \*\*Implementación y Despliegue\*\***

- Estrategia de implementación.

La estrategia de implementación se centra en la transición del desarrollo a la etapa de prueba y producción. Dado que este proyecto se ha desarrollado individualmente, la estrategia puede seguir un enfoque simple:

1. Pruebas en Entorno de Desarrollo: Realizar pruebas exhaustivas en el entorno de desarrollo para garantizar que todas las funcionalidades y características se comporten según lo esperado.
2. Pruebas en Entorno de Prueba: Implementar el software en un entorno de prueba simulado que refleje el entorno de producción. Realizar pruebas adicionales para verificar la compatibilidad y la estabilidad.
3. Despliegue en Entorno de Producción: Lanzar el software en el entorno de producción real para que los ciudadanos y los servidores públicos puedan comenzar a usarlo.

- Plan de despliegue en entornos de prueba y producción.

El plan de despliegue incluye configurar entornos de prueba y producción, copiar bases de datos y realizar pruebas exhaustivas antes del lanzamiento.

## 9. **\*\*Mantenimiento y Soporte\*\***

- Plan para el mantenimiento futuro del software.

1. Monitorización Continua: Establecer un sistema de monitorización para rastrear el rendimiento y la disponibilidad del software en tiempo real. Esto ayudará a identificar problemas antes de que afecten a los usuarios.
2. Recolección de Comentarios: Fomentar la retroalimentación de los usuarios para identificar áreas de mejora y nuevas funcionalidades que puedan ser incorporadas en futuras actualizaciones.
3. Actualizaciones Planificadas: Programar actualizaciones regulares para mantener el software al día y mejorar su rendimiento. Estas actualizaciones pueden incluir mejoras en la seguridad, correcciones de errores y nuevas características.
4. Soporte al Usuario: Proporcionar canales de soporte para que los usuarios puedan informar sobre problemas y recibir ayuda en caso de dificultades.

- Proceso de corrección de errores y actualizaciones.

El proceso de corrección de errores y actualizaciones implica identificar y analizar problemas a través de informes de usuarios y pruebas internas, priorizarlos según su gravedad, desarrollar soluciones y nuevas características, realizar pruebas exhaustivas para garantizar la estabilidad, implementar las actualizaciones en entornos de prueba y producción, comunicar estos cambios a los usuarios y recopilar su retroalimentación para una mejora continua del software en términos de funcionalidad, seguridad y satisfacción del usuario.

## 10. **\*\*Gestión de Riesgos\*\***

- Identificación de riesgos potenciales.

1. Retraso en el Desarrollo: Posibles retrasos en el proceso de desarrollo debido a la complejidad de las funcionalidades y la curva de aprendizaje de las tecnologías.

2. Problemas de Compatibilidad: Incompatibilidad con diferentes navegadores, dispositivos y sistemas operativos que podrían afectar la experiencia del usuario.
3. Fallas de Seguridad: Vulnerabilidades y posibles ataques cibernéticos que podrían comprometer la seguridad de los datos de los usuarios.
4. Baja Adopción: La plataforma podría no ser adoptada por los ciudadanos o servidores públicos, afectando su utilidad y éxito general.

- Planes de mitigación y contingencia para los riesgos.

1. Retraso en el Desarrollo:

- Mitigación: Diseñar un cronograma realista y asignar suficiente tiempo para cada fase del desarrollo.
- Contingencia: Establecer un margen de tiempo adicional en el cronograma para hacer frente a posibles retrasos.

2. Problemas de Compatibilidad:

- Mitigación: Realizar pruebas exhaustivas en diferentes navegadores y dispositivos durante el desarrollo.
- Contingencia: Tener un plan para solucionar problemas de compatibilidad y mejorar la experiencia del usuario según los comentarios.

3. Fallas de Seguridad:

- Mitigación: Implementar prácticas sólidas de seguridad desde el inicio del desarrollo, como encriptación y autenticación segura.
- Contingencia: Tener un plan de respuesta a incidentes y actualizaciones de seguridad programadas para abordar cualquier vulnerabilidad identificada.

4. Baja Adopción:

- Mitigación: Involucrar a los usuarios clave desde etapas tempranas para comprender sus necesidades y obtener retroalimentación.

- Contingencia: Tener un plan de marketing y divulgación para promover la plataforma entre los ciudadanos y los servidores públicos.

## 11. **\*\*Presupuesto y Recursos\*\***

### - Estimación de costos y presupuesto.

- Costo de desarrollo del software y diseño de la interfaz.
- Costo de herramientas de desarrollo, licencias y software adicional.
- Costo de pruebas y herramientas de prueba automatizada.
- Posibles gastos para pruebas de seguridad y rendimiento.
- Costo de configuración y mantenimiento de entornos de prueba y producción.
- Posibles gastos relacionados con servicios de alojamiento.
- Costo de mantenimiento continuo y posibles actualizaciones.

### - Recursos necesarios: personal, herramientas, equipos, etc.

- Desarrollador de software para la implementación del proyecto.
- Diseñador para la creación de la interfaz de usuario.
- Personal de pruebas y control de calidad para las etapas de validación.
- Computadoras y equipos de desarrollo para el equipo.
- Herramientas de desarrollo, como el entorno de programación y las herramientas de diseño.
- Licencias para las tecnologías utilizadas, como Python, Django, Bootstrap, etc.
- Servidores y servicios de alojamiento para implementación y prueba