|  |
| --- |
| Documento de Diseño Arquitectónico |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del Proyecto:** | DenunciaSeguro | **Fecha de Preparación:** | 04 de septiembre del 2023 |
| **Preparado por:** | Dafna Peña, Jefe de Proyecto / Programador FrontEnd  Fredy Huerta, Arquitecto de Software  Edson Gutierrez, DBA | **Autorizado por:** | Gerencia General |

**Índice**

[**Introducción 2**](#_2ucc8kwlicjk)

[**Descripción del Sistema 2**](#_41iu7z3bb8sf)

[2.1. Objetivos del Sistema 2](#_1c8xuoe7be06)

[2.2. Tecnologías Utilizadas 3](#_b44cifv3gswf)

[**Arquitectura de la Aplicación 3**](#_g9dsx4kcf4mj)

[3.1. Arquitectura de Tres Capas (MVC) 3](#_5edee5x7kqiv)

[3.2. Arquitectura General del Sistema 4](#_qc2e26s8lfi0)

[a) Interfaz de Usuario (UI): 4](#_4ogm3wgox00)

[b) Controlador (Control): 4](#_jpl8fjnaelue)

[c) Modelo (Model): 4](#_i3g3xjwzgzcf)

[d) Base de Datos (DB): 5](#_y95nitctcnht)

[**Medidas de Seguridad en el Software 5**](#_ohurhfhc5tok)

[a) Autenticación 5](#_wg2snd7v1k2w)

[b) Autorización 6](#_v81udbqrsi01)

[c) Protección de Datos 6](#_to7wewy839xd)

[d) Protección contra Amenazas Conocidas 6](#_8zmay1nhru6d)

[e) Gestión de Sesiones: 6](#_dq2ocqbczfkn)

[f) Actualizaciones de Seguridad 6](#_4eb5b2gth675)

[**Diagrama de contexto 7**](#_7nzt44d9f0xa)

[**Conclusiones 7**](#_9d9lrl2rwlan)

[**Recomendaciones 8**](#_2e2g723g6bg7)

|  |
| --- |
| Introducción |

El presente documento de diseño de arquitectura de software proporcionará una visión detallada de cómo se estructurará y funcionará la Plataforma de Denuncias Públicas. A lo largo de este documento, se explorarán aspectos técnicos como la arquitectura de la aplicación, los componentes del sistema, la seguridad, la escalabilidad y otros aspectos clave necesarios para el desarrollo exitoso de este proyecto

|  |
| --- |
| Descripción del Sistema |

“DenunciaSeguro” es una aplicación web que permitirá a los ciudadanos reportar y denunciar una variedad de problemas en sus comunidades de manera eficaz. La plataforma actuará como un canal centralizado para que los ciudadanos informen sobre estos problemas y busca comunicarse en tiempo real con las autoridades locales y nacionales para asegurar una respuesta rápida y eficiente.

La aplicación se basa en un modelo de tres capas (Modelo-Vista-Controlador - MVC), donde el Modelo representa la estructura de datos y la lógica de negocio, la Vista maneja la presentación de la información al usuario y el Controlador gestiona la lógica de negocio y la comunicación entre el Modelo y la Vista.

Una característica clave de la plataforma es la integración con sistemas de información geográfica (GIS) que permitirá la visualización de las denuncias en un mapa interactivo. Esto facilitará a los usuarios y a las autoridades la comprensión de la distribución geográfica de los problemas reportados y la toma de decisiones informadas.

## 2.1. Objetivos del Sistema

* Proporcionar a los ciudadanos una plataforma accesible y fácil de usar para reportar problemas en sus comunidades.
* Utilizar tecnología GIS para visualizar las denuncias en un mapa interactivo, lo que facilitará la comprensión de la ubicación geográfica de los problemas reportados.
* Garantizar la seguridad de los datos y la privacidad de los usuarios a través de medidas de seguridad sólidas.
* Diseñar la plataforma para ser escalable y capaz de manejar un aumento en la carga de usuarios y denuncias.
* Promover la transparencia al hacer que las denuncias y las respuestas de las autoridades sean visibles para toda la comunidad, construyendo así confianza en las instituciones y en el proceso de resolución de problemas.
* Generar datos sobre los problemas denunciados y su ubicación para que las autoridades puedan tomar decisiones informadas sobre la asignación de recursos y la planificación de mejoras a largo plazo en las comunidades.

## 2.2. Tecnologías Utilizadas

* Lenguaje de Programación: Python
* Framework de Desarrollo: Flask
* Base de Datos: SQL

|  |
| --- |
| Arquitectura de la Aplicación |

## 3.1. Arquitectura de Tres Capas (MVC)

El patrón MVC se utiliza para separar las preocupaciones en una aplicación y organizar el código de manera más clara. Se puede describir de la siguiente manera:

* Modelo: El Modelo representa los datos y la lógica de negocio de la aplicación. Sus responsabilidades incluyen la gestión de la información, el acceso a la base de datos, el procesamiento de datos y la lógica que define cómo se comporta la aplicación. El Modelo no se preocupa por cómo se muestra la información ni por cómo el usuario interactúa con ella; simplemente maneja los datos y las operaciones sobre ellos.
* Vista: La Vista es la capa encargada de la presentación de la información al usuario y de la interacción con él. Representa la interfaz de usuario y muestra los datos del Modelo de manera adecuada.Esta capa no realiza ningún procesamiento de datos ni contiene lógica de negocio; simplemente muestra la información de manera legible y permite al usuario interactuar con ella.
* Controlador: El Controlador actúa como intermediario entre el Modelo y la Vista. Recibe las solicitudes del usuario desde la Vista, procesa esas solicitudes, consulta o actualiza el Modelo según sea necesario y luego decide qué Vista debe mostrarse al usuario en respuesta. La lógica de enrutamiento y control está en esta capa. También, garantiza que las acciones del usuario se traduzcan en operaciones apropiadas en el Modelo y que los resultados se presenten adecuadamente en la Vista.

## 3.2. Arquitectura General del Sistema

### Interfaz de Usuario (UI):

* Componentes UI: Páginas web, formularios de denuncia, paneles de usuario, mapas de calor, sistema de evaluación ciudadana, chatbot de asistencia.
* Funcionalidades UI: Permite a los usuarios presentar denuncias, ver el estado de sus denuncias, interactuar con el chatbot de asistencia, acceder a estadísticas y mapas de calor, evaluar las respuestas de las autoridades.
* Relaciones: La interfaz de usuario interactúa con el Controlador y muestra datos del Modelo.

### Controlador (Control):

* Componentes Control: Manejadores de rutas, controladores de denuncias, controladores de usuarios, chatbot de asistencia.
* Funcionalidades Control: Gestiona las solicitudes del usuario, enruta las solicitudes a las funciones apropiadas, verifica la autenticación y autorización de los usuarios, coordina la interacción entre la interfaz de usuario y el modelo, proporciona respuestas a través del chatbot de asistencia.
* Relaciones: El controlador se comunica con la interfaz de usuario y el modelo para procesar y responder a las solicitudes del usuario.

### Modelo (Model):

* Componentes Modelo: Base de datos, lógica de negocio, clases de denuncias, clases de usuarios, sistema de notificaciones.
* Funcionalidades Modelo: Almacena y gestiona datos relacionados con las denuncias y los usuarios, maneja la autenticación y autorización de usuarios, implementa la lógica de negocio (validación de denuncias, notificaciones a las autoridades), gestiona alertas automáticas, almacena y recupera estadísticas.
* Relaciones: El Modelo es consultado y actualizado por el Controlador y proporciona datos a la Interfaz de Usuario.

### Base de Datos (DB):

* Componentes DB: Tablas de base de datos (tablas de denuncias, tablas de usuarios, tablas de estadísticas).
* Funcionalidades DB: Almacena y recupera datos de manera eficiente y segura, almacena estadísticas y registros de actividad, y almacena la información de las denuncias y los usuarios.
* Relaciones: La Base de Datos es utilizada por el Modelo para almacenar y recuperar datos y registros de actividad.

Esta arquitectura permite que los ciudadanos presenten denuncias de manera efectiva, obtengan información relevante sobre el estado de sus denuncias y colaboren con las autoridades. Además, proporciona herramientas de monitoreo, evaluación y mejora continua para los administradores de la plataforma. La seguridad y la privacidad son consideraciones críticas en todas las capas para proteger la información y la identidad de los denunciantes.

|  |
| --- |
| Medidas de Seguridad en el Software |

La seguridad es una consideración crítica en cualquier plataforma de denuncias ciudadanas, ya que se trata de proteger la información sensible de los denunciantes y garantizar la integridad de las denuncias. A continuación, describo las medidas de seguridad que se implementarán en el software:

## Autenticación

* Autenticación de Usuarios: Se implementará un sólido sistema de autenticación para garantizar que solo usuarios autorizados tengan acceso a la plataforma.
* Autenticación de Autoridades: Las autoridades que responden a las denuncias también deberán autenticarse en la plataforma para acceder a la información y las funciones correspondientes.

## Autorización

* Control de Acceso Basado en Roles: Se establecerán roles y permisos para los usuarios y las autoridades. Los usuarios solo tendrán acceso a sus propias denuncias y datos personales, mientras que las autoridades tendrán acceso solo a las denuncias asignadas a ellos.
* Políticas de Autorización: Se establecerán políticas de autorización sólidas para determinar quién puede realizar qué acciones en la plataforma. Esto se aplicará a funciones como la modificación de denuncias, la eliminación de datos o el acceso a estadísticas.

## Protección de Datos

* Encriptación de Datos: Se aplicará encriptación para proteger los datos.

## Protección contra Amenazas Conocidas

* Validación de Datos: Se realizará una validación adecuada de todos los datos ingresados por los usuarios para prevenir ataques de inyección y otros ataques basados en datos maliciosos.
* Monitoreo de Actividad: Se realizará un seguimiento de las actividades en la plataforma para detectar actividades sospechosas o no autorizadas.

## Gestión de Sesiones:

* Cierre de Sesión Seguro: Los usuarios podrán cerrar sesión de manera segura para asegurarse de que no queden sesiones abiertas en dispositivos compartidos.

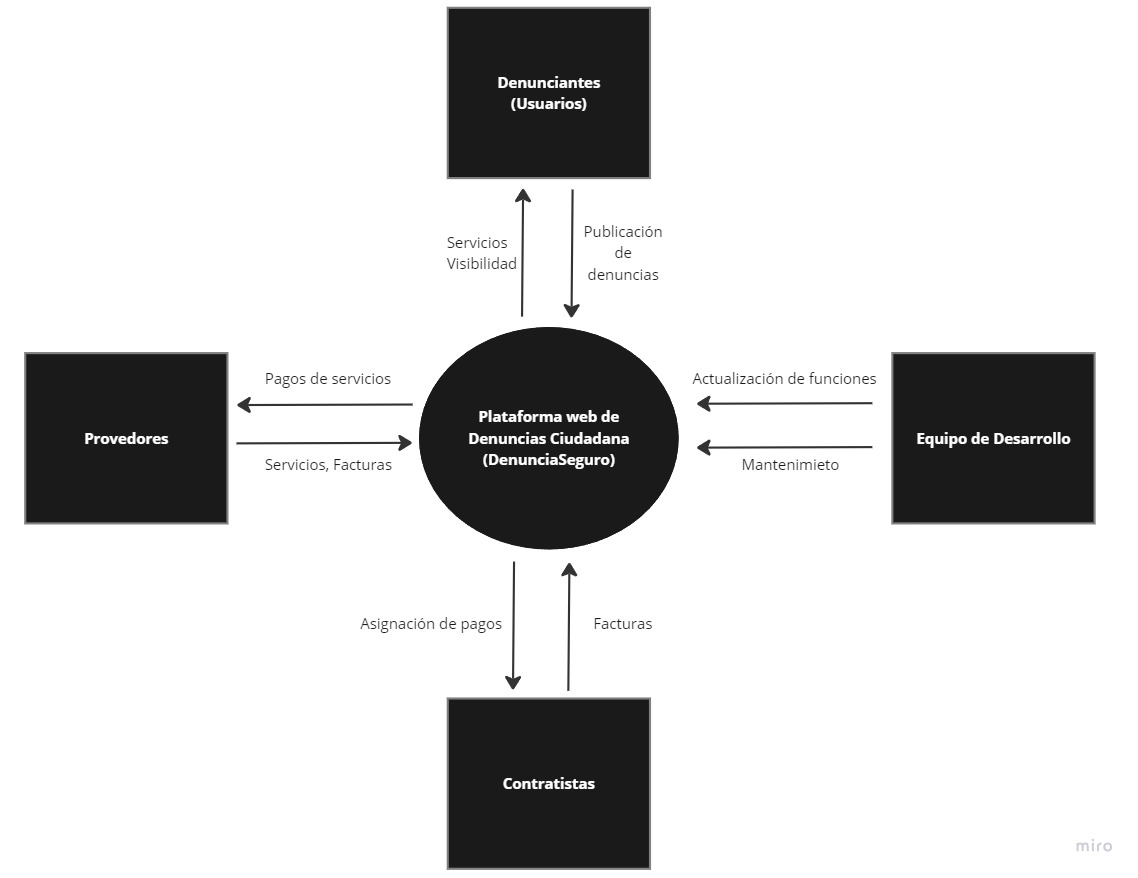
## Actualizaciones de Seguridad

* Mantenimiento Regular: Se realizarán actualizaciones regulares del software para corregir vulnerabilidades conocidas y aplicar parches de seguridad.
* Gestión de Vulnerabilidades: Se llevará a cabo una gestión proactiva de vulnerabilidades para identificar y abordar posibles amenazas.

La seguridad debe ser una preocupación continua en el ciclo de vida del software de denuncias ciudadanas. Se deben realizar pruebas de seguridad regulares y auditorías para garantizar que las medidas de seguridad sean efectivas y estén actualizadas en respuesta a las amenazas cambiantes.

|  |
| --- |
| Diagrama de contexto |

El diagrama de contexto es una herramienta gráfica que se utiliza en el análisis de sistemas y la ingeniería de software para representar de una manera simplificada la interacción entre un sistema o proceso en estudio y su entorno circundante. El objetivo principal del presente diagrama de contexto es mostrar cómo nuestro sistema se comunica con elementos externos como los usuarios, proveedores, equipo de desarrollo y contratistas.



|  |
| --- |
| Conclusiones |

1. Estructura y Objetivos Claros: El diseño de la arquitectura de software está bien definido y tiene objetivos claros, lo que es fundamental para el desarrollo exitoso del proyecto.
2. Uso de Tecnologías Apropiadas: La elección de Python y Flask como tecnologías principales es acertada, ya que ofrecen facilidad durante el desarrollo y una sólida base para el desarrollo web y son ampliamente utilizadas en la industria.
3. Arquitectura MVC: La adopción del patrón Modelo-Vista-Controlador (MVC) para la arquitectura de la aplicación es una buena práctica que separa las preocupaciones y facilita la organización del código.
4. Integración GIS: La integración de sistemas de información geográfica (GIS) es una característica valiosa que mejorará la comprensión de la ubicación geográfica de las denuncias, lo que facilitará la toma de decisiones informadas.
5. Seguridad: La atención a la seguridad se refleja en la implementación de medidas sólidas de autenticación, autorización y protección de datos, lo que es esencial para proteger la privacidad de los usuarios y la integridad de las denuncias.
6. Gestión de Sesiones y Actualizaciones: La gestión de sesiones seguras y las actualizaciones regulares de seguridad son prácticas importantes para mantener la plataforma segura a lo largo del tiempo.

|  |
| --- |
| Recomendaciones |

1. Pruebas de Seguridad: Se deben llevar a cabo pruebas de seguridad periódicas para identificar y mitigar posibles vulnerabilidades. Esto debería ser parte integral del proceso de desarrollo y mantenimiento.
2. Auditorías de Seguridad: Realizar auditorías de seguridad independientes para garantizar la robustez de las medidas implementadas y cumplir con las mejores prácticas de seguridad.
3. Documentación: Asegurarse de que se mantenga una documentación actualizada que describa las políticas de seguridad, roles y permisos, y los procedimientos para la gestión de incidentes de seguridad.
4. Formación en Seguridad: Proporcionar formación en seguridad cibernética a todo el personal involucrado en el desarrollo y mantenimiento de la plataforma para crear conciencia sobre las amenazas y buenas prácticas de seguridad.
5. Monitoreo Continuo: Implementar un sistema de monitoreo continuo para detectar y responder a actividades inusuales o posibles ataques en tiempo real.
6. Plan de Respuesta a Incidentes: Desarrollar y mantener un plan de respuesta a incidentes para abordar de manera efectiva cualquier violación de seguridad que pueda ocurrir en el futuro.
7. Usabilidad: Asegurarse de que la plataforma sea fácil de usar para los ciudadanos, lo que puede requerir pruebas de usabilidad y ajustes en la interfaz de usuario según los comentarios de los usuarios.

En resumen, la Plataforma de Denuncias Públicas tiene un diseño sólido y orientado a la seguridad. Sin embargo, mantener un enfoque continuo en la seguridad, la usabilidad y la actualización es esencial para garantizar su éxito a largo plazo y mantener la confianza de los usuarios.