REPORTA PE – RED SOCIAL DE DENUNCIAS

DOCUMENTO

**DISEÑO ARQUITECTONICO**

**DATOS DEL DOCUMENTO:**

|  |  |
| --- | --- |
| **Versión** | 1.1 |
| **Fecha de Elaboración** | 05 de mayo de 2024 |
| **Fecha de Actualización** | 09 de mayo de 2024 |
| **Fecha de Autorización** | **-** |

**ELABORACIÓN:**

|  |  |  |
| --- | --- | --- |
| **ÁREA** | **DES** | Administrador de base de datos |
| **RESPONSABLES** | Bryan Anthony, Velarde  Lipa Palacios, Renzo  Cueva Mantura, Alvaro Bertilo  Castillo Carranza, Jose Richard | |

**HISTORIAL DE REVISIONES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ÍTEM** | **FECHA** | **VERSIÓN** | **AUTOR(ES)** | **RAZÓN DEL CAMBIO** | **RESPONSABLE DE APROBACIÓN** |
| **01** | 05/05/23 | 1.0 | -Bryan Anthony Velarde  -Lipa Palacion Renzo  -Cueva Mantura Alvaro Bertilo  - Castillo Carranza, Jose Richard | Elaboración de plantilla y estructura del documento |  |

**Tabla de Contenidos**

[1. Introducción 5](#_Toc167623825)

[2. Objetivos 6](#_Toc167623826)

[3. Alcance 7](#_Toc167623827)

[4. Arquitectura 9](#_Toc167623828)

[4.1. Diagrama de contexto 9](#_Toc167623829)

[4.2. Alto nivel 9](#_Toc167623830)

[4.3. MVC 11](#_Toc167623831)

# Introducción

El presente documento de diseño arquitectónico tiene como objetivo principal proporcionar una visión detallada y estructurada de la arquitectura propuesta para el sistema [nombre del sistema]. Este sistema tiene como finalidad implementar una red social para facilitar, sistematizar y visibilizar las denuncias ciudadanas, con el fin de incrementar la transparencia, la rendición de cuentas y la participación ciudadana.

El diseño arquitectónico es un componente crucial en el proceso de desarrollo de software, ya que define la estructura y organización del sistema, así como las interacciones entre sus componentes. Este documento servirá como guía para los desarrolladores, arquitectos de software y otras partes interesadas involucradas en el proyecto, proporcionando una comprensión clara de cómo se construirá y funcionará el sistema.

A lo largo de este documento, se abordarán varios aspectos importantes del diseño arquitectónico, incluyendo la arquitectura de alto nivel, los patrones de diseño utilizados, las tecnologías y herramientas seleccionadas, consideraciones de rendimiento y seguridad, y estrategias de mantenimiento y evolución. Además, se presentarán las decisiones clave tomadas durante el proceso de diseño y se justificarán en función de los requisitos y objetivos del sistema.

Es importante destacar que este documento está sujeto a modificaciones y actualizaciones a medida que avanza el proceso de desarrollo y se obtienen más detalles sobre el sistema.

# Objetivos

La aplicación de Red Social de Reportes de Delitos, desarrollada utilizando tecnologías como HTML, CSS, Python y JavaScript, tiene como objetivo principal proporcionar a los usuarios una plataforma segura y eficiente para informar sobre incidentes delictivos, compartir información relevante sobre la seguridad en sus comunidades y colaborar en la prevención de crímenes.

* **Definir una arquitectura robusta y escalable**: El diseño arquitectónico se centrará en establecer una estructura sólida y adaptable que pueda manejar el crecimiento de usuarios y la carga de trabajo sin comprometer el rendimiento del sistema.
* **Garantizar la seguridad de los datos y la privacidad de los usuarios**: Se implementarán medidas de seguridad sólidas para proteger la información sensible de los usuarios, como los reportes de delitos y los datos de contacto, utilizando prácticas de cifrado, autenticación segura y control de acceso.
* **Fomentar la modularidad y la reutilización de componentes:** Se diseñará la aplicación con un enfoque modular, dividiendo la funcionalidad en componentes independientes y reutilizables que faciliten el desarrollo, la prueba y el mantenimiento del sistema a largo plazo.
* **Optimizar el rendimiento y la experiencia del usuario**: Se prestará especial atención a la optimización del rendimiento del sistema, asegurando tiempos de carga rápidos, una interfaz de usuario receptiva y una experiencia fluida para los usuarios, incluso en condiciones de alta demanda.
* **Facilitar la evolución y la incorporación de nuevas características**: La arquitectura permitirá una fácil adaptación a los cambios en los requisitos del sistema y la introducción de nuevas características y funcionalidades, facilitando así su mantenimiento y evolución continua.

# Alcance

Este diseño arquitectónico abarca los siguientes aspectos:

* **Diseño de la arquitectura del sistema**: Se definirá una arquitectura robusta y escalable que permita la modularidad y la flexibilidad del sistema. Se seleccionarán patrones arquitectónicos adecuados, como MVC (Modelo-Vista-Controlador) o arquitectura de microservicios, para garantizar la adaptabilidad del sistema a medida que crece.
* **Estructura de la base de datos**: Se diseñará una estructura de base de datos eficiente para almacenar la información relevante, como reportes de delitos, datos de usuarios y contactos. Se utilizarán tecnologías de bases de datos relacionales o no relacionales según las necesidades y los requisitos de la aplicación.
* **Medidas de seguridad**: Se implementarán medidas de seguridad sólidas para proteger la información sensible de los usuarios y garantizar la integridad y confidencialidad de los datos. Esto incluirá prácticas de cifrado, autenticación segura, control de acceso y gestión de sesiones.
* **Optimización del rendimiento**: Se realizarán acciones para optimizar el rendimiento del sistema, como minimizar los tiempos de carga de la aplicación, reducir el uso de recursos y mejorar la capacidad de respuesta de la interfaz de usuario.
* **Integración de servicios externos**: Se integrarán servicios externos, como APIs de mapas para mostrar ubicaciones de incidentes, servicios de autenticación para la gestión de usuarios y servicios de almacenamiento en la nube para guardar archivos adjuntos a los reportes.
* **Documentación y seguimiento**: Se proporcionará documentación detallada del diseño arquitectónico, que incluirá diagramas, descripciones de componentes, requisitos de hardware y software, así como instrucciones para el despliegue y mantenimiento del sistema. Se establecerán métricas de rendimiento y se implementará un sistema de seguimiento para monitorear la salud y el funcionamiento del sistema en producción.

# Arquitectura

## Diagrama de contexto

Imagen que contiene Gráfico en cascada

Descripción generada automáticamente

* **Usuarios**: Representan a los usuarios de la aplicación que reportan sucesos, gestionan reportes, visualizan el mapa de seguridad, guardan reportes, buscan reportes y contactos, e inician denuncias.
* **Autoridades gubernamentales**: Representan a las entidades gubernamentales o instituciones a las que se envían las denuncias formales para su seguimiento y procesamiento.

## Alto nivel

**Capa de presentación (Frontend):**

Esta capa es responsable de la interfaz de usuario de la aplicación, que incluye la página web y la aplicación móvil.

Se utilizarán tecnologías web estándar como HTML, CSS y JavaScript para desarrollar una interfaz de usuario atractiva e interactiva.

Frameworks de desarrollo frontend como React.js o Angular.js pueden ser utilizados para facilitar la construcción y gestión de la interfaz de usuario.

**Capa de lógica de aplicación (Backend):**

En esta capa reside la lógica de negocio de la aplicación, encargada de procesar solicitudes de los usuarios, gestionar la autenticación, y manejar la lógica de los reportes de delitos.

Se empleará un lenguaje de programación como Python para implementar la lógica del backend, aprovechando frameworks web como Django o Flask para acelerar el desarrollo y mantener un código organizado y eficiente.

**Base de datos:**

La base de datos almacenará todos los datos necesarios para el funcionamiento de la aplicación, como los reportes de delitos, información de los usuarios y contactos.

Se considerará el uso de una base de datos relacional como PostgreSQL o MySQL, o una base de datos NoSQL como MongoDB, dependiendo de los requisitos específicos de la aplicación y las preferencias del equipo de desarrollo.

**Servicios externos:**

La aplicación puede integrarse con servicios externos para ampliar su funcionalidad, como servicios de mapas para visualizar la ubicación de los incidentes, servicios de autenticación para la gestión de usuarios, y servicios de almacenamiento en la nube para guardar archivos adjuntos a los reportes.

Se evaluarán cuidadosamente las opciones de integración para garantizar la compatibilidad y seguridad de los servicios externos seleccionados.

## MVC

**Modelo (Model)**

El Modelo representa los datos y la lógica de negocio de la aplicación. Incluye las clases y métodos que manejan la interacción con la base de datos, así como las reglas de negocio.

* **Reporte delictivo**: Clase que representa un reporte delictivo. Contiene atributos como **id**, **descripcion**, **ubicacion**, **fecha**, **tipoDelito**, **evidencias**, y métodos para crear, leer, actualizar y eliminar reportes.
* **Usuario**: Clase que representa a un usuario de la aplicación. Incluye atributos como **id**, **nombre**, **email**, **contraseña**, **reportesGuardados**, y métodos para autenticación y gestión de usuarios.
* **Denuncia formal**: Clase que maneja el proceso de enviar una denuncia a las autoridades. Incluye atributos como **id**, **reporte**, **estado**, **fechaEnvio**, y métodos para iniciar, actualizar y consultar el estado de una denuncia.

**Vista (View)**

La Vista es responsable de la presentación de los datos. Incluye las interfaces de usuario (UI) y la lógica de presentación que se muestra al usuario. Las Vistas reciben datos del Controlador y los renderizan al usuario.

* **Formulario reportar suceso**: Vista que muestra el formulario para reportar un suceso delictivo.
* **Lista de reportes**: Vista que presenta la lista de reportes realizados por el usuario.
* **Mapa de seguridad**: Vista que muestra un mapa interactivo con los reportes delictivos geolocalizados.
* **Búsqueda de reportes**: Vista que permite a los usuarios buscar reportes usando diferentes criterios.
* **Detalle de denuncia**: Vista que muestra los detalles de una denuncia formal.

**Controlador (Controller)**

El Controlador actúa como intermediario entre el Modelo y la Vista. Recibe las entradas del usuario a través de la Vista, procesa la lógica de negocio correspondiente en el Modelo y actualiza la Vista con los resultados.

* **Controlador reportar suceso**: Maneja las solicitudes del usuario para reportar un suceso, valida los datos, crea un nuevo reporte en el Modelo y actualiza la Vista.
* **Controlador gestionar reportes**: Controla la visualización, edición y eliminación de los reportes realizados por el usuario.
* **Controlador mapa de seguridad**: Gestiona la interacción con el mapa, recuperando datos de reportes del Modelo y proporcionando los datos a la Vista.
* **Controlador búsqueda de reportes**: Procesa las búsquedas de reportes, solicita los resultados del Modelo y actualiza la Vista con los resultados.
* **Controlador iniciar denuncia**: Maneja el proceso de inicio de denuncia, validando la información y coordinando con el Modelo para enviar la denuncia a las autoridades.