



2025

Diseño e implementación de una aplicación móvil como herramienta tecnológica para la prevención de la violencia en Totonicapán, Guatemala.

MANUAL TECNICO

Sistema y aplicación “SOS sin miedo”

QUETZALTENANGO

V.0.1

Para más información, por favor contactase

Danny Francisco Sapón Tax

(502) 5574 3737

dsapont@miumg.edu.gt

Contenido

Acerca del proyecto.....	5
1. Introducción.....	6
2. Dependencias:.....	6
Dependencias para la aplicación móvil.....	6
Para administrador Web	8
3. Arquitectura general del sistema	11
Componentes principales	11
4. Instalación del sistema web	13
a. Clonar el repositorio	13
b. Configurar entorno	13
c. Instalar dependencias.....	14
d. Migrar base de datos.....	14
e. Generar clave de aplicación.....	14
f. Configurar acceso público	14
5. Estructura de carpetas (Laravel API).....	14
6. Base de datos (MySQL)	15
7. APIs y endpoints principales	15

8.	Integraciones externas	16
9.	Aplicación móvil (Flutter)	17
10.	Seguridad y control de acceso	17
11.	Logs y depuración	18
12.	Despliegue	18
13.	Respaldo y mantenimiento	19
14.	Escalabilidad y futuras mejoras	19
15.	Diccionario de Datos	19
16.	Créditos técnicos	25
17.	Diagramas UML	26
	Diagrama de Clases	26
	Diagramas de Comportamiento	27
	Diagrama de secuencias	27
	Diagrama de Actividades	28
	Diagrama de Estado	29
	Diagrama de Componentes	30
	Diagrama de Despliegue	31

Acerca del proyecto

Aplicación móvil desarrollada en *Flutter* con el lenguaje *Dart*, que permite reportar hechos de violencia para personas de todas las edades. Además, cuenta con una plataforma web desarrollada en *Laravel* con *PHP*, que facilita la gestión y derivación de los casos reportados desde la aplicación móvil.

1. Introducción

En el presente documento se describe los aspectos internos, arquitectónicos y funcionales del sistema SOS Sin Miedo, que combina una plataforma web institucional y una aplicación móvil ciudadana, conectadas mediante una API REST desarrollada en Laravel.

Objetivo técnico:

Proporcionar a personal técnico y desarrolladores la información necesaria para:

- Instalar, configurar, mantener y escalar el sistema.
- Comprender el flujo completo backend y frontend.
- Integrar nuevos módulos o mejorar los ya existentes.

2. Dependencias:

Dependencias para la aplicación móvil

Paquete	Versión	Descripción técnica	Uso dentro del proyecto
flutter	SDK	Framework principal para construir la aplicación móvil multiplataforma.	Base de toda la app. Proporciona los widgets y librerías nativas para Android e iOS.
http	^1.2.0	Cliente HTTP estándar de Dart para realizar peticiones REST.	Se usa para comunicación inicial o llamadas básicas a la API Laravel.
dio	^5.7.0	Cliente HTTP avanzado con interceptores, manejo de errores y	Principal herramienta para conectar con la API Laravel (envío de denuncias,

Paquete	Versión	Descripción técnica	Uso dentro del proyecto
		cabeceras personalizadas.	consultas y autenticación).
geolocator	^14.0.2	Permite obtener la localización del dispositivo con alta precisión.	Captura coordenadas GPS (latitud y longitud) al realizar una denuncia.
url_launcher	^6.3.2	Abre enlaces desde la app (teléfonos, correos, direcciones web).	Abre números de emergencia o recursos web desde la sección de “Ayuda”.
firebase_core	^4.2.0	Librería base para inicializar todos los servicios de Firebase.	Integración obligatoria para login con número telefónico.
firebase_auth	^6.1.1	Implementa la autenticación Firebase vía SMS.	Permite a los usuarios autenticarse mediante token SMS para las denuncias autenticadas.
provider	^6.0.5	Paquete de manejo de estado basado en	Gestiona el estado global de controladores (AuthController,

Paquete	Versión	Descripción técnica	Uso dentro del proyecto
		el patrón Provider.	ReportController, etc.).
flutter_secure_storage	^9.0.0	Permite guardar datos sensibles cifrados en el dispositivo.	Guarda de forma segura el token del usuario (Bearer Token Laravel Sanctum).
flutter_launcher_icons	^0.14.4	Personaliza el ícono y splash de la app.	Genera los íconos oficiales de la aplicación (SOS Sin Miedo).
cupertino_icons	^1.0.8	Paquete de íconos estilo iOS (complementario al Material).	Mejora visual y compatibilidad entre plataformas (Android/iOS).

Para administrador Web

Dependencias principales (Laravel Core)

Estas vienen instaladas por defecto al crear un proyecto Laravel:

Paquete	Descripción
"laravel/framework"	Núcleo del framework Laravel. Incluye Eloquent, Routing, Queues, Validation, etc.

Paquete	Descripción
"laravel/sanctum"	Manejo de autenticación por Token (usado en la app móvil para proteger tus endpoints).
"laravel/tinker"	Consola interactiva para desarrollo y depuración (PHP REPL).
"fruitcake/laravel-cors"	Habilita CORS para permitir peticiones desde la app Flutter o el frontend.
"guzzlehttp/guzzle"	Cliente HTTP usado para enviar peticiones externas (lo utilizas en Twilio).

Dependencias adicionales instaladas por configuración del proyecto

Estas no son propias de Laravel, pero las agregamos tú y yo para funcionalidades específicas del sistema:

Paquete	Uso dentro del proyecto
"twilio/sdk"	Comunicación con la API de Twilio para enviar SMS de alerta.
"kreat/laravel-firebase"	Validación de tokens de Firebase (para login desde la app móvil).
"spatie/laravel-permission"	Asignación de roles y permisos (usuarios de la plataforma web).
"barryvdh/laravel-debugbar" (<i>en desarrollo</i>)	Depuración visual de consultas, rutas y performance en entorno local.

Paquete	Uso dentro del proyecto
"laravel/ui" o "laravel/breeze" (dependiendo del setup web)	Generadores de scaffolding para login/register/reset con autenticación por correo.
"intervention/image" (<i>opcional</i>)	Procesamiento de imágenes (si planeas permitir subir evidencias o fotos).
"barryvdh/laravel-dompdf" (<i>opcional</i>)	Generación de reportes PDF o descargas institucionales.

Dependencias de desarrollo

Estas ayudan durante el desarrollo, pero no son necesarias en producción:

Paquete	Uso
"fakerphp/faker"	Generación de datos falsos para pruebas (usuarios, denuncias, etc.).
"phpunit/phpunit"	Framework de pruebas para Laravel.
"mockery/mockery"	Mocking para test unitarios.
"nunomaduro/collision"	Mejora el manejo visual de errores en la terminal.
"spatie/laravel-ignition"	Mejora la detección y visualización de errores durante desarrollo.

Dependencias NPM (para el panel web con Tailwind CSS)

En el frontend de Laravel (panel de administración o portal web), usas las siguientes dependencias Node:

Paquete	Descripción
"tailwindcss"	Framework CSS utilizado en todas las vistas del panel.
"autoprefixer"	Añade compatibilidad automática para navegadores.
"postcss"	Procesador de CSS para Tailwind.
"vite"	Compilador moderno (Laravel Mix ya fue reemplazado).
"alpinejs"	Librería ligera para interactividad en el portal web.

Se instala con:

-
-
- npm install
- npm run build

3. Arquitectura general del sistema

Componentes principales

Componente	Descripción
Frontend Web (Laravel Blade / Tailwind)	Panel para operadores e instituciones. Permite login con correo, validación, gestión de denuncias y usuarios.
Backend (Laravel API + Sanctum)	API REST que gestiona autenticaciones, registros de denuncias, alertas SMS y consultas.
Aplicación móvil (Flutter)	App Android que permite al ciudadano enviar denuncias (autenticadas o anónimas).
Autenticación Firebase	Uso de Firebase Auth para validación vía número telefónico en la app.

SMS Alerts (Twilio)	Servicio externo que envía alertas por SMS a unidades de atención.
Base de datos MySQL	Persistencia de entidades principales: denuncias, usuarios, tipos de violencia, unidades, etc.

Patrón de arquitectura:

Modelo Cliente - Servidor con API Intermedia

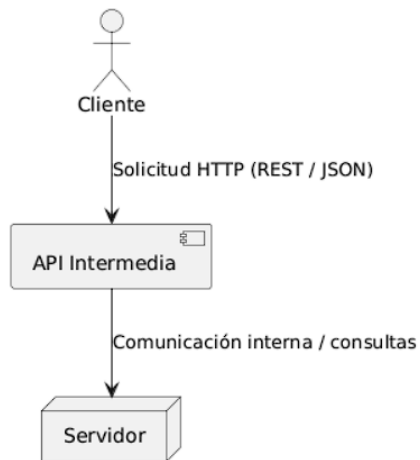
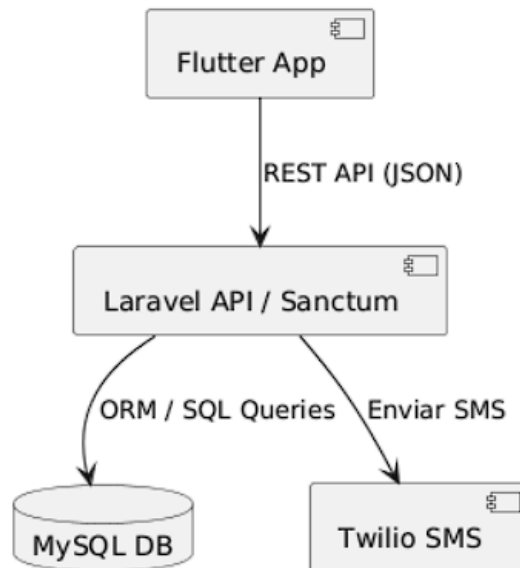


Diagrama general (resumen):

Arquitectura General - Flutter + Laravel + MySQL + Twilio



Requisitos técnicos

Servidor Backend (Laravel)

- PHP 8.2 o superior
- Composer 2
- MySQL 8.0 o MariaDB
- Node.js (para compilar Tailwind en el panel web) en este caso se compila en local todo y se sube ya compilado dado que se usa un servidor compartido, al usar un servidor dedicado, se rewliza la compilación en el servidor
- Extensiones requeridas: pdo, json, mbstring, openssl

Configuración recomendada

Entorno	Valor
Servidor web	Nginx
Sistema operativo	Windows Server
Hosting	VPSm o server Compartido con mínimo 2 GB RAM
Certificado SSL	Obligatorio (para API segura en móviles)

4. Instalación del sistema web

a. Clonar el repositorio

```
git clone https://github.com/DannyS12/sos_sinmiedo.git
cd sos_sinmiedo
```

b. Configurar entorno

Editar .env con las credenciales necesarias:

```
APP_NAME="SOS Sin Miedo"
APP_ENV=production
APP_URL=https://sossinmiedo.midominio.com
```

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=sos_db
DB_USERNAME=root
DB_PASSWORD=tu_password
```

```
TWILIO_ACCOUNT_SID=xxxxxxxxxxxxx
TWILIO_AUTH_TOKEN=xxxxxxxxxxxxx
TWILIO_PHONE_NUMBER=+502XXXXXXX
```

```
SANCTUM_STATEFUL_DOMAINS=sistemas502.com
```

c. Instalar dependencias

```
composer install
npm install && npm run build
```

d. Migrar base de datos

```
php artisan migrate --seed
```

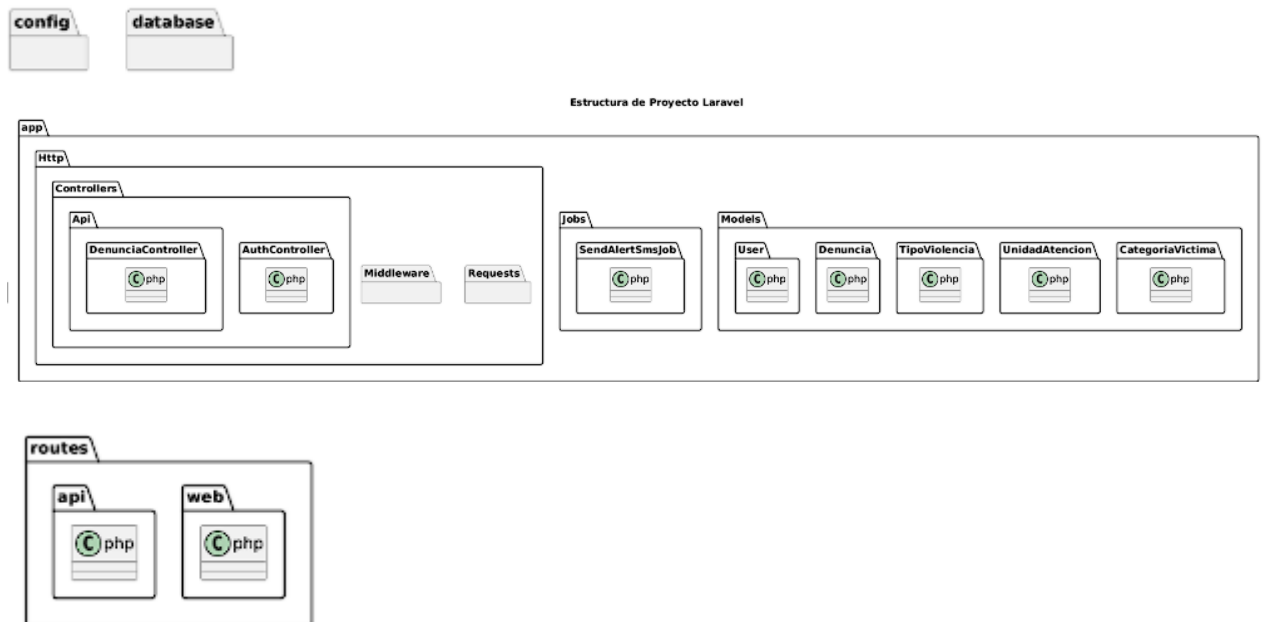
e. Generar clave de aplicación

```
php artisan key:generate
```

f. Configurar acceso público

Asegúrate de que el servidor apunte a la carpeta /public.

5. Estructura de carpetas (Laravel API)



6. Base de datos (MySQL)

Tablas principales

Tabla	Descripción
users	Usuarios de la plataforma (web y app autenticada).
roles	Define permisos y niveles de acceso.
denuncias	Registros de las denuncias realizadas.
tipos_de_violencia	Catálogo de tipos de violencia.
unidades_de_atencion	Instituciones o unidades receptoras de casos.
categorias_victimas	Agrupaciones por rango etario o condición.
ubicaciones	Coordenadas geográficas de las denuncias.

Relaciones principales:

User (1) --- (N) Denuncia
Denuncia (N) --- (1) TipoViolencia
Denuncia (N) --- (1) UnidadAtencion
Denuncia (N) --- (1) CategoriaVictima
Denuncia (1) --- (1) Ubicacion

7. APIs y endpoints principales

Método	Ruta	Descripción	Auth
POST	/api/denuncias/anonima	Registrar denuncia anónima	No

POST	/api/denuncias	Registrar denuncia autenticada	Sí
GET	/api/denuncias/mis-denuncias	Listar denuncias del usuario autenticado	Sí
GET	/api/tipos-violencia	Catálogo de tipos de violencia	No
GET	/api/unidades	Listado de unidades de atención	Sí
POST	/api/login	Login web (correo/contraseña)	No
POST	/api/register	Registro web + validación de correo	No
POST	/api/password/reset	Solicitud para restaurar contraseña	No

Las peticiones autenticadas se manejan con Laravel Sanctum (Bearer Token).

8. Integraciones externas

Firebase Authentication (Flutter App)

- Firebase se usa solo para el flujo de autenticación vía número telefónico.
- Una vez validado el token Firebase en el backend, Laravel genera un token Sanctum.

Twilio

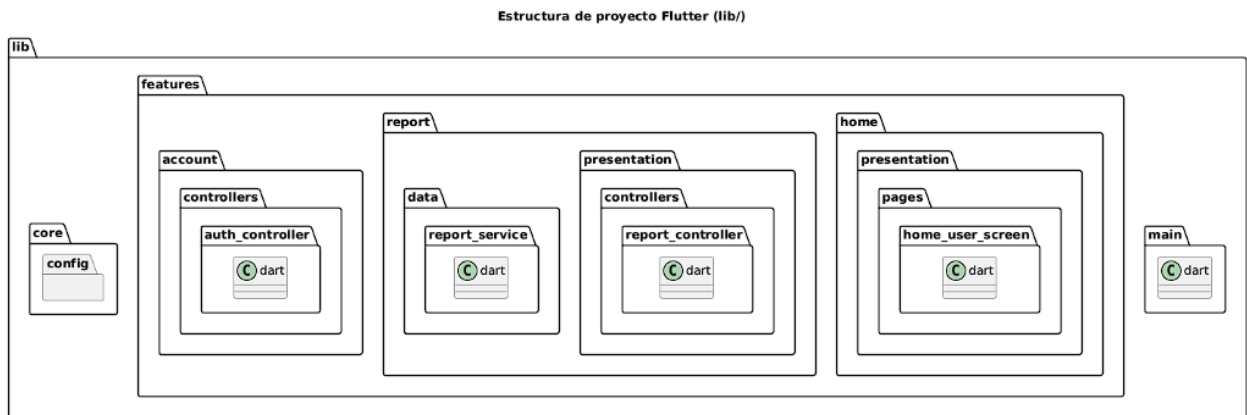
- Se usa para enviar SMS automáticos con el módulo SendAlertSmsJob.
- El envío se dispara tras el guardado exitoso de una denuncia.

- Los errores (por número inválido, etc.) se registran en storage/logs/laravel.log.

9. Aplicación móvil (Flutter)

- Clona desde la misma dirección que la versión web
https://github.com/DannyS12/sos_sinmiedo.git

Estructura general



Funcionalidades principales

- Inicio de sesión (Firebase Phone Auth)
- Denuncia anónima o autenticada
- Envío a endpoints Laravel vía Dio
- Consulta de denuncias personales (/mis-denuncias)
- Ver detalle en modal con ubicación y estado

10. Seguridad y control de acceso

- **Laravel Sanctum** maneja tokens personales para app y web.
- **Roles y permisos** asignados a nivel de usuario en roles y role_user.
- Todas las rutas críticas (/denuncias, /mis-denuncias) están protegidas por middleware auth:sanctum.
- **Validación del formato Twilio (E.164)** evita fallos de SMS.
- Se recomienda activar HTTPS obligatorio para API y app.

11. Logs y depuración

Ubicación	Descripción
storage/logs/laravel.log	Errores de Laravel o Twilio
flutter_logs/ (opcional)	Registro de errores en la app
php artisan tinker	Pruebas de endpoints
php artisan queue:work	Procesamiento de SMS en background (en caso de cola real)

12. Despliegue

Backend Laravel

- Subir código a servidor (Plesk / VPS / Hostinger / etc.)
- Configurar dominio y SSL
- Asegurar que DocumentRoot → /public
- Ejecutar:

```
composer install --no-dev
php artisan migrate --force
php artisan config:cache
php artisan route:cache
```

Aplicación móvil

1. Configurar archivo lib/core/config/app_config.dart con URL base (ej: <https://sossinmiedo.midominio.com/public/api>).
2. Compilar versión release:

```
flutter build apk --release
```

3. Firma y distribución (Play Store / instalación directa).

13. Respaldo y mantenimiento

Proceso	Frecuencia	Herramienta
Backup de base de datos	Diario	mysqldump / cron
Logs del servidor	Semanal	logrotate o limpieza manual
Actualización de dependencias	Mensual	composer update
Certificados SSL	Anual	certbot o panel hosting

14. Escalabilidad y futuras mejoras

- Implementación de **notificaciones push** (Firebase Messaging).
- Integración de **mapas interactivos** para ver ubicación de denuncias.
- Panel analítico para estadísticas de casos.
- Modo offline para denuncias sin conexión.
- Migración de colas SMS Twilio a **Redis Queue Worker**.

15. Diccionario de Datos

Términos generales del sistema

Término	Definición	Contexto
Sistema	Conjunto de componentes (web, móvil, backend, BD) que interactúan para cumplir los objetivos.	Se refiere a la solución completa: portal web + app móvil + API.
Aplicación Móvil (App)	Software diseñado para dispositivos móviles (smartphones, tablets).	La app Flutter que usan los ciudadanos para denunciar.

Término	Definición	Contexto
Portal Web	Interfaz de usuario accesible vía navegador web.	La plataforma Laravel para operadores y administradores.
Backend	Parte del sistema que procesa la lógica de negocio, gestiona la BD y la API.	El servidor Laravel que recibe denuncias, envía SMS y gestiona datos.
Frontend	Interfaz de usuario con la que interactúa el usuario final.	La app Flutter y las vistas Blade/Tailwind del portal web.
API (Application Programming Interface)	Conjunto de reglas que permite a diferentes programas comunicarse entre sí.	La API REST de Laravel que conecta la app móvil con el backend.
Base de Datos (BD)	Colección organizada de información estructurada.	MySQL, donde se almacenan denuncias, usuarios, tipos de violencia, etc.
Middleware	Software que actúa como puente entre sistemas operativos o aplicaciones.	En Laravel, filtra peticiones (ej. auth:sanctum para proteger rutas).
Endpoint	URL específica de una API a la que se envía una petición.	Ej. /api/denuncias/anonima o /api/denuncias/mis-denuncias.
Token	Cadena de caracteres que representa una credencial de autenticación.	Usado por Firebase (SMS) y Laravel Sanctum (Bearer Token).

Término	Definición	Contexto
Geolocalización (GPS)	Determinación de la posición geográfica de un objeto o persona.	Coordenadas (latitud/longitud) adjuntas a cada denuncia.
Logs	Registros de eventos, errores o actividades del sistema.	Archivos donde se guardan errores de Laravel, Twilio, etc. (storage/logs/laravel.log).

Términos de usuario y autenticación

Término	Definición	Contexto
Usuario Ciudadano	Persona que utiliza la aplicación móvil para realizar denuncias.	Accede por número telefónico o de forma anónima.
Usuario Institucional	Personal autorizado que gestiona denuncias desde el portal web.	Accede por correo electrónico y tiene un rol asignado.
Autenticación	Proceso de verificar la identidad de un usuario.	Login con Firebase (móvil) o correo/contraseña (web).
Autorización	Proceso de determinar qué acciones puede realizar un usuario autenticado.	Definido por los roles y permisos en el portal web.
Perfil de Usuario	Conjunto de atributos y permisos asociados a una cuenta de usuario.	Determina las funcionalidades accesibles en el portal web (ej. Administrador, Operador).

Término	Definición	Contexto
Denunciante	Persona que realiza una denuncia.	Puede ser anónimo o un usuario autenticado.
Token de Verificación (SMS)	Código numérico enviado por SMS para confirmar la identidad.	Usado por Firebase para el login móvil.
Token de Acceso (Bearer Token)	Credencial generada por el backend para acceder a recursos protegidos.	Generado por Laravel Sanctum, usado por la app móvil.
Cerrar Sesión (Logout)	Finalizar la sesión activa de un usuario.	Disponible en la app móvil y el portal web.

Términos de denuncias y gestión de casos

Término	Definición	Contexto
Denuncia	Reporte formal de un incidente o acto de violencia.	El registro principal del sistema, creado desde la app o el portal.
Denuncia Anónima	Reporte sin identificación del denunciante.	Permite a los usuarios enviar información sin revelar su identidad.
Denuncia Autenticada	Reporte vinculado a la cuenta de un usuario registrado.	Permite al usuario hacer seguimiento de su caso.

Término	Definición	Contexto
Número de Caso	Identificador único asignado a cada denuncia.	Facilita el seguimiento y la referencia de un reporte.
Tipo de Violencia	Clasificación del incidente reportado (ej. física, psicológica).	Campo obligatorio en el formulario de denuncia.
Estado de la Denuncia	Fase actual en la que se encuentra el proceso de una denuncia.	Ej. Pendiente, En Proceso, Cerrado.
Unidad de Atención	Institución o departamento responsable de gestionar un tipo de denuncia.	Asignada automáticamente según el tipo de violencia y/o ubicación.
Categoría de Víctima	Clasificación de la persona afectada (ej. menor, adolescente, adulto).	Ayuda a las unidades a priorizar y especializar la atención.
Evidencia	Material de apoyo (fotos, videos) adjunto a una denuncia.	Opcional en el formulario de denuncia.
Alerta SMS	Mensaje de texto automático enviado a contactos de emergencia.	Disparado por Twilio al registrar una denuncia.
Mapa Geográfico	Herramienta visual integrada en el sistema que representa espacialmente las denuncias mediante marcadores en un mapa digital, permitiendo la localización y	Utilizado cuando se necesita visualizar los casos en el mapa

Término	Definición	Contexto
	análisis geoespacial de los incidentes reportados.	
Ruta de Denuncia	Flujo o pasos que sigue una denuncia desde su creación hasta su resolución.	Información disponible en la sección de ayuda.

Términos de tecnologías e integraciones

Término	Definición	Contexto
Flutter	Framework de Google para desarrollar apps móviles multiplataforma.	Usado para construir la aplicación móvil de SOS Sin Miedo.
Laravel	Framework PHP para el desarrollo de aplicaciones web y APIs.	Usado para el backend y el portal web.
Firebase	Plataforma de Google para desarrollo de apps (autenticación, bases de datos, etc.).	Usado específicamente para la autenticación por número telefónico (Firebase Auth).
Twilio	Plataforma de comunicación en la nube para envío de SMS, llamadas, etc.	Usado para enviar las alertas SMS a las unidades de atención.
MySQL	Sistema de gestión de bases de datos relacionales.	Base de datos principal del sistema.

Término	Definición	Contexto
Sanctum	Paquete de autenticación de Laravel para APIs y SPAs.	Genera y valida los tokens de acceso para la app móvil.
Dio	Cliente HTTP para Dart/Flutter con características avanzadas.	Usado en la app Flutter para todas las peticiones a la API Laravel.
Provider	Patrón de gestión de estado para Flutter.	Organiza y comparte la lógica de negocio (controladores) en la app.
Tailwind CSS	Framework CSS utilitario para diseño rápido de interfaces.	Usado en el frontend del portal web.

Glosario de acrónimos

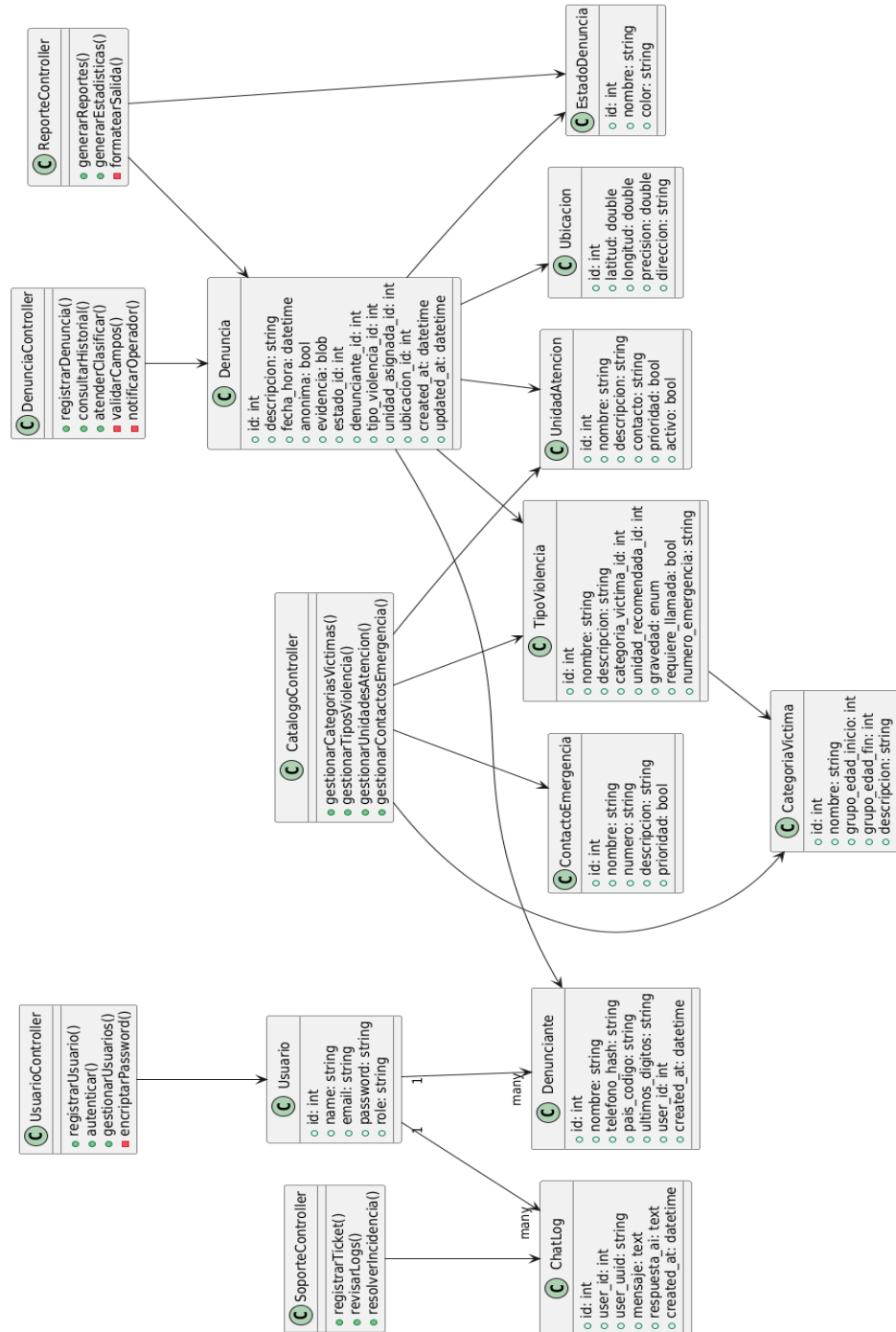
Acrónimo	Significado
API	Application Programming Interface
BD	Base de Datos
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
SMS	Short Message Service
URL	Uniform Resource Locator
UX	User Experience
UI	User Interface

16. Créditos técnicos

Área	Responsable
Arquitectura e implementación Laravel	Danny Francisco Sapón Tax
Aplicación móvil Flutter	Danny Francisco Sapón Tax
Integraciones Firebase y Twilio	Danny Francisco Sapón Tax
Diseño de base de datos y API REST	Danny Francisco Sapón Tax
Infraestructura y despliegue	Danny Francisco Sapón Tax

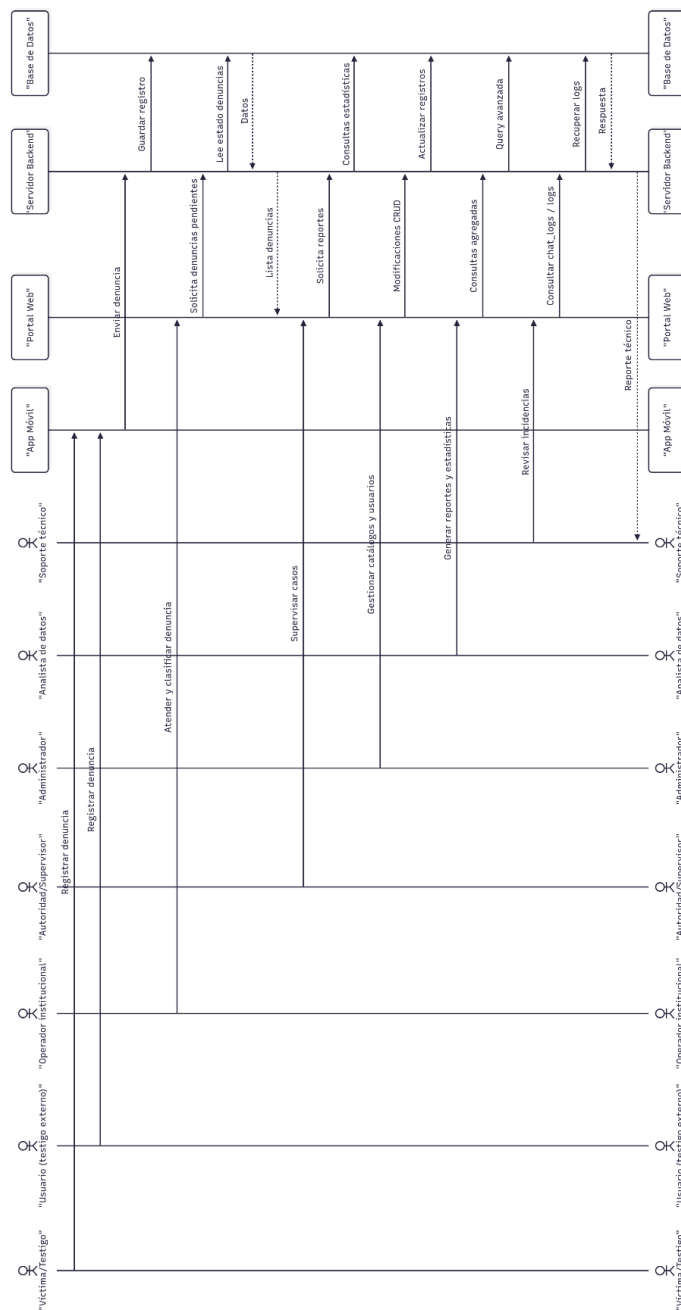
17. Diagramas UML

Diagrama de Clases



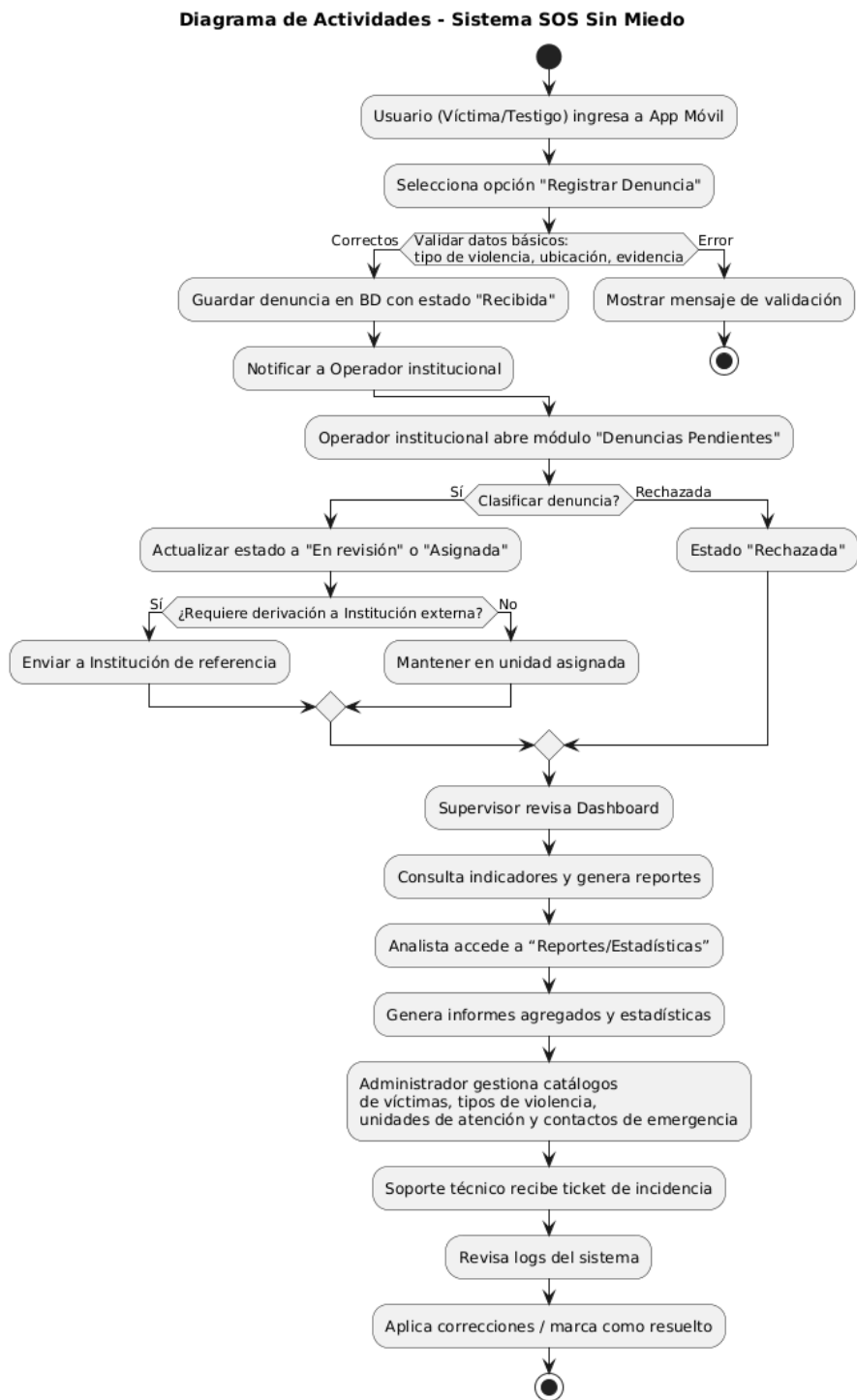
Nota. Imagen de autoría propia para el sistema

Diagrama de secuencias



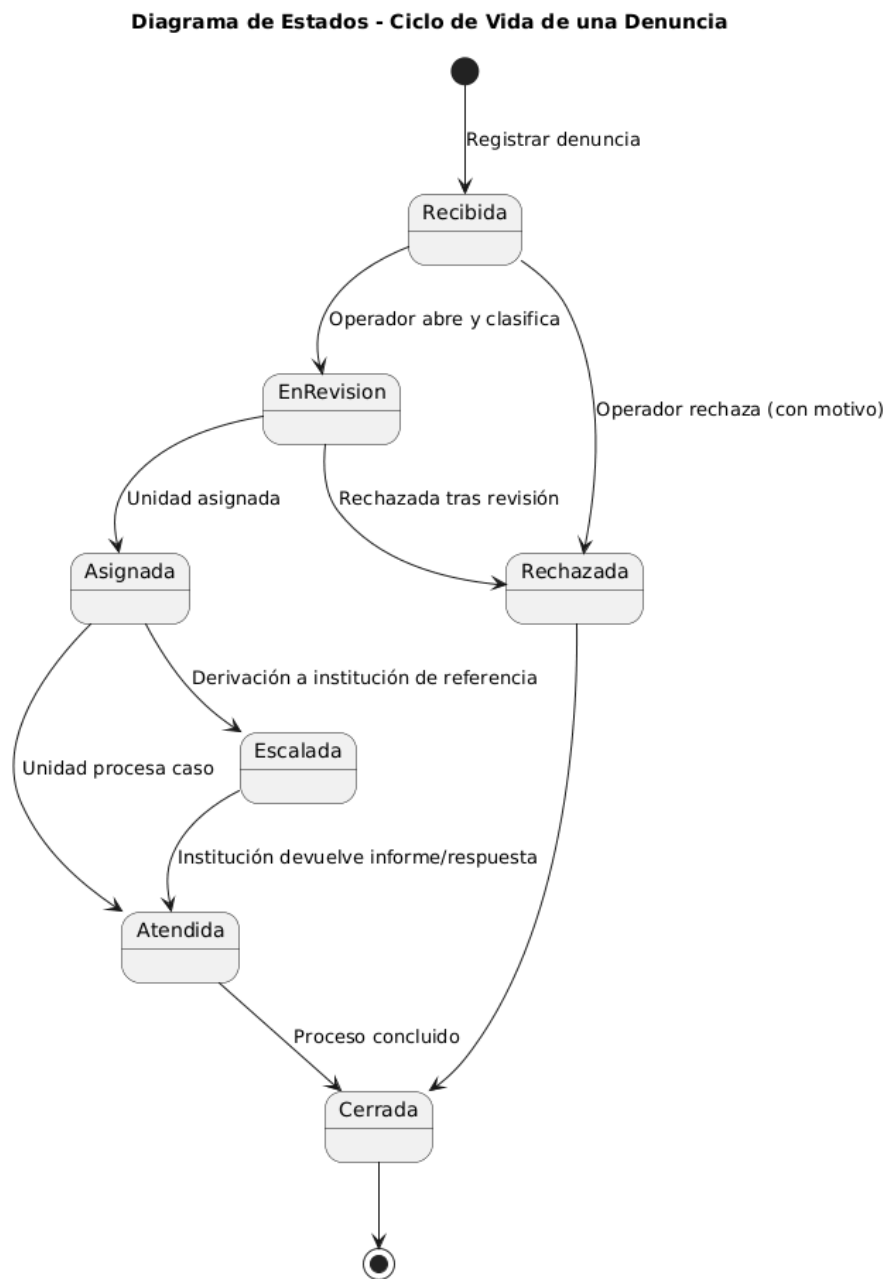
27

Diagrama de Actividades



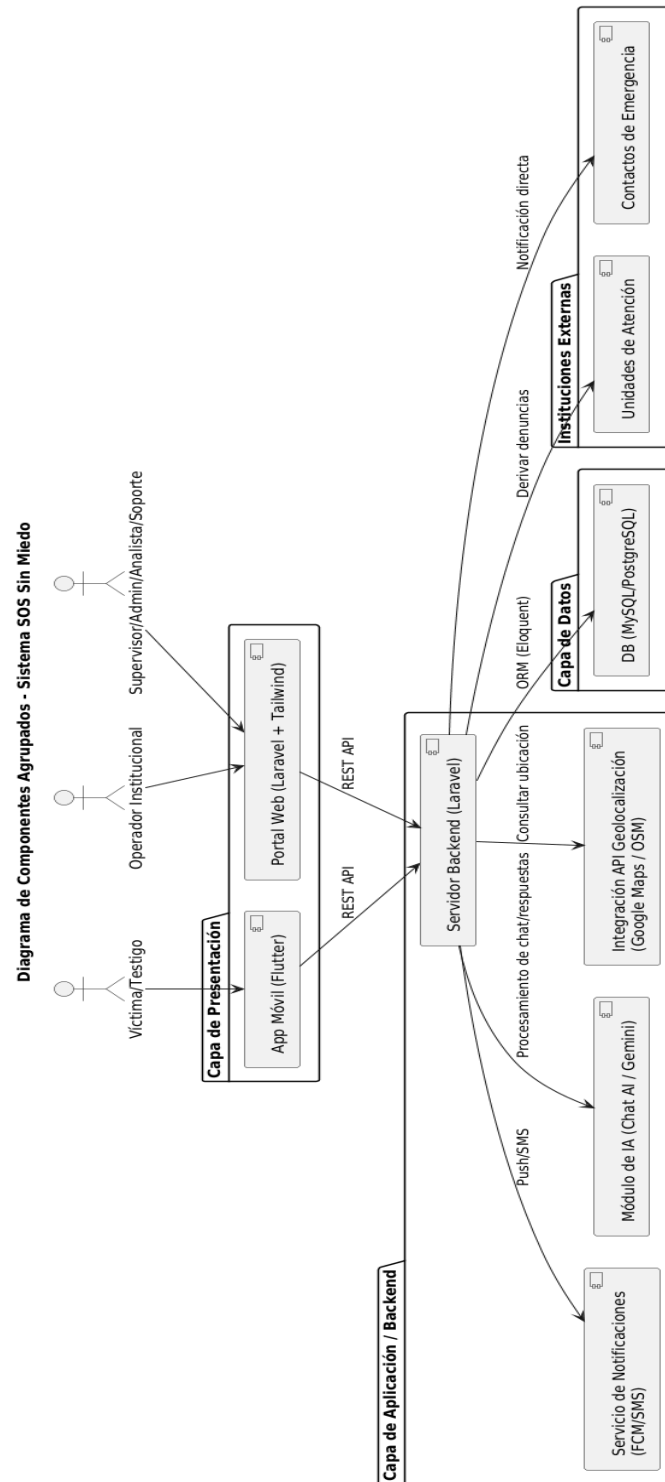
Nota. Imagen de autoría propia para el sistema

Diagrama de Estado



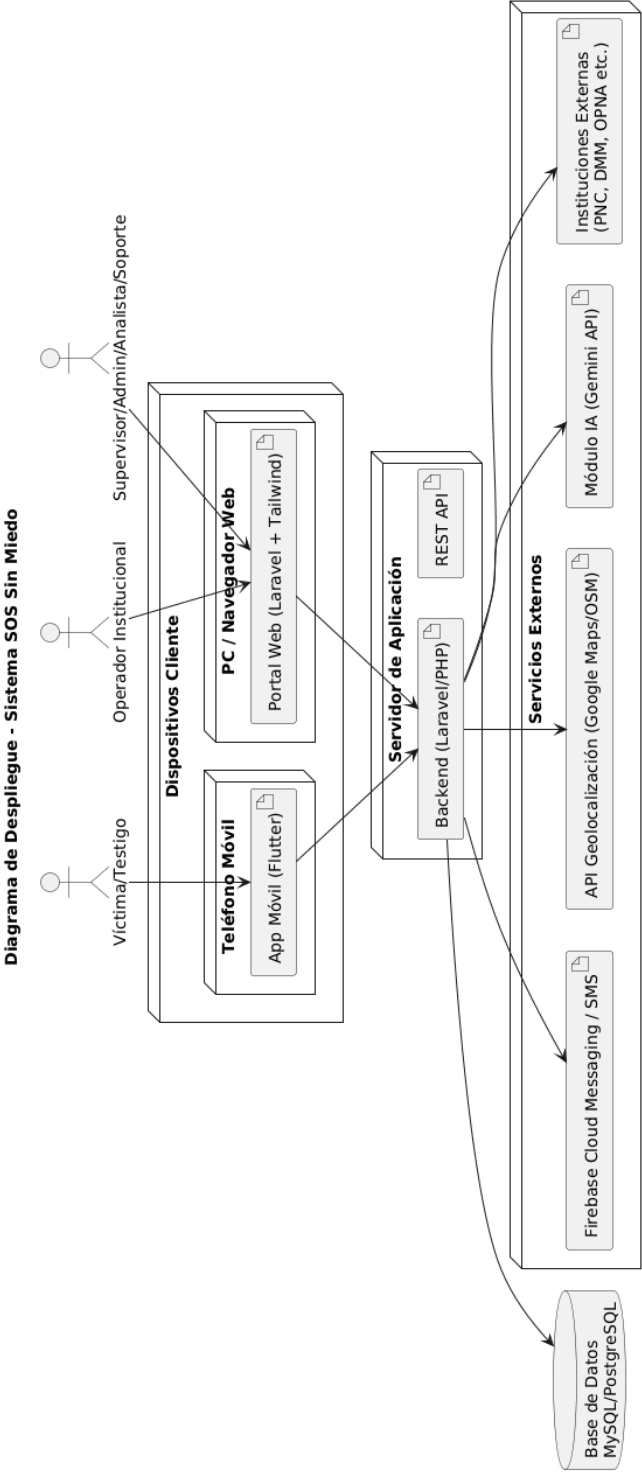
Nota. Imagen de autoría propia para el sistema

Diagrama de Componentes



Nota. Imagen de autoría propia para el sistema

Diagrama de Despliegue



Nota. Imagen de autoría propia para el sistema

