

# Plantilla para documento de especificación de requisitos de software

Nombre del proyecto:	Plataforma de Reporte de Problemas Urbanos
Fecha:	Quito, 14 de abril del 2025
Versión:	1.0
Creado por:	Alex Mendoza Morante, David Tello, Carlos Plua

Historial de revisiones			
Versión	Autor	Descripción de la versión	Fecha de realización
1.0	David Tello	Creacion del documento	14-04-2025

Historial de revisiones			
Revisor	Versión revisada	Firma	Fecha
Alex Mendoza	1.0		17-04-2025

Historial de aprobaciones			
Aprobador	Versión revisada	Firma	Fecha
Carlos Plua	1.0		17-04-2025

# Índice

1

## Introducción

- 1.1 Alcance del producto
- 1.2 Valor del producto
- 1.3 Público objetivo
- 1.4 Uso previsto
- 1.5 Descripción general

2

## Requisitos funcionales

3

## Requisitos de la interfaz externa

- 3.1 Requisitos de la interfaz de usuarios
- 3.2 Requisitos de la interfaz de hardware
- 3.3 Requisitos de la interfaz de software
- 3.4 Requisitos de la interfaz de comunicación

4

## Requisitos no funcionales

- 4.1 Seguridad
- 4.2 Capacidad
- 4.3 Compatibilidad
- 4.4 Confiabilidad
- 4.5 Escalabilidad
- 4.6 Mantenibilidad
- 4.7 Facilidad de uso
- 4.8 Otros requisitos no funcionales

5

## Definiciones y acrónimos

# 1 Introducción

Describe el propósito del documento.

La calidad de vida en las ciudades depende en gran medida del estado de la infraestructura urbana y del funcionamiento adecuado de los servicios públicos. Calles en mal estado, luminarias apagadas, acumulación de residuos y problemas similares afectan no solo la estética del entorno, sino también la seguridad y la salud de las personas que habitan estos espacios. A pesar de que muchas municipalidades cuentan con áreas dedicadas al mantenimiento urbano, los canales disponibles para reportar estos problemas suelen ser limitados, burocráticos o poco accesibles para el ciudadano común.

## 1.1 Alcance del producto

Enumera los beneficios, los objetivos y las metas del producto.

VozUrbana tiene como alcance principal el desarrollo de una aplicación web que permita a los ciudadanos reportar problemáticas urbanas de forma georreferenciada y con un sistema de seguimiento de estado. El sistema incluirá funcionalidades para el registro de usuarios, la creación de reportes por categoría, la carga de imágenes ilustrativas y la visualización de dichos reportes en un mapa interactivo.

## 1.2 Valor del producto

Describe qué valor encontrará el público en tu producto.

En 'VozUrbana', los usuarios encontrarán una plataforma web intuitiva y de código abierto diseñada para transformar su participación en el mejoramiento de sus ciudades. Con esta herramienta, podrán reportar fácilmente problemas urbanos como calles dañadas, mala iluminación o acumulación de residuos— mediante un sistema accesible desde cualquier dispositivo con internet. La plataforma no solo agiliza el proceso de denuncia, sino que también georreferencia y organiza los reportes en un mapa interactivo, permitiendo visualizar problemas recurrentes y priorizar soluciones. Además, fomenta la transparencia al hacer visible el seguimiento de cada caso, empoderando a los ciudadanos para colaborar activamente con las autoridades en la construcción de entornos más seguros, saludables y sostenibles.

## 1.3 Público objetivo

Describe a quién está destinado el producto.

VozUrbana está destinada a ciudadanos comprometidos con su comunidad, vecinos organizados, líderes barriales y cualquier persona que desee reportar problemas urbanos de manera eficiente y transparente. También está dirigida a municipalidades, gobiernos locales y organizaciones civiles que buscan mejorar la gestión urbana mediante la participación ciudadana y datos georreferenciados. La plataforma es ideal para quienes valoran la corresponsabilidad en el espacio público, desde jóvenes con acceso a tecnología hasta asociaciones de vecinos que requieren herramientas sencillas para incidir en soluciones concretas. Además, al ser de código abierto, resulta atractiva para desarrolladores e instituciones que quieran adaptarla a necesidades específicas de sus ciudades.

## 1.4 Uso previsto

Describe de qué manera tu público objetivo usará el producto.

VozUrbana es una plataforma web que permite a ciudadanos reportar problemas urbanos como baches, basura o alumbrado público de forma rápida, con fotos y geolocalización. Los vecinos pueden seguir el estado de sus reportes en tiempo real, mientras las municipalidades usan los datos para priorizar reparaciones con transparencia. Organizaciones y líderes barriales generan informes para impulsar cambios, y los medios verifican soluciones. Al ser de código abierto, desarrolladores pueden adaptarla a nuevas ciudades. Conectando a ciudadanos, gobiernos y sociedad civil, convierte la denuncia pasiva en acción colectiva organizada.

## 1.5 Descripción general

Redacta un resumen de las funciones y características del software.

Enumera los requisitos de diseño, los requisitos gráficos, los requisitos del sistema operativo y las limitaciones del producto.

### Requisitos de diseño:

- Interfaz de usuario intuitiva y fácil de navegar.
- Diseño adaptable a diferentes tamaños de pantalla de dispositivos móviles.
- Uso de imágenes de alta calidad para las recetas.
- Posibilidad de personalizar la apariencia de la aplicación (modo claro/oscuro, tamaño de fuente, etc.).

### Requisitos gráficos:

- Iconografía clara y representativa de las funcionalidades de la aplicación.
- Uso de colores atractivos y que fomenten la experiencia culinaria.
- Animaciones suaves y fluidas para mejorar la interacción del usuario.

- **Requisitos del sistema operativo:**

- Compatibilidad con las últimas versiones de Android e iOS.
- Optimización para el uso eficiente de la batería y los recursos del dispositivo.
- Notificaciones push para alertar sobre nuevas recetas o actualizaciones de la aplicación.

### 3.1 Requisitos de la interfaz de usuario

Describe la lógica detrás de las interacciones entre los usuarios y el software (diseños de pantalla, guías de estilo, etc.).

#### Mapa interactivo:

- Mostrar la ubicación del usuario actual utilizando GPS o selección manual.
- Mostrar los problemas reportados cercanos con marcadores visuales (iconos) que indiquen el tipo de problema.
- Permitir la búsqueda de problemas por ubicación, tipo o estado.

#### Formulario de reporte:

- Campos para la descripción del problema, ubicación (automática o manual), tipo de problema (lista desplegable), fotos/videos (opcional).
- Opción para reportar de forma anónima o con información de contacto.

#### Lista de problemas:

- Mostrar los problemas reportados en formato de lista, con información básica (tipo, ubicación, fecha, estado).
- Filtros para buscar por tipo, estado, ubicación o fecha.
- Opción para ordenar los problemas por relevancia, fecha o cercanía.

### 3.2 Requisitos de la interfaz de hardware

Especifica los dispositivos compatibles con el software, los requisitos de red y los protocolos de comunicación que se utilizarán.

#### Dispositivos compatibles:

- Smartphone y tablets con sistemas operativos Android e iOS (últimas versiones y versiones anteriores compatibles).
- Navegadores web (Chrome, Safari, Firefox, Edge) en computadoras de escritorio y portátiles.

#### Requisitos de red:

- Conexión a internet (Wi-Fi o datos móviles) para acceder al mapa, enviar reportes y recibir actualizaciones.
- Funcionalidad offline limitada (por ejemplo, guardar reportes para enviar más tarde).

### 3.3 Requisitos de la interfaz de software

Incluye las conexiones entre tu producto y otros componentes de software, como el marco de front-end y back-end, las bibliotecas, etc.

#### **Conexiones con otros componentes:**

- Integración con el sistema de gestión de incidencias del gobierno local (API).
- Integración con servicios de mapas (Google Maps, OpenStreetMap).
- Integración con servicios de geolocalización (GPS, servicios de ubicación de red).

### 3.4 Requisitos de la interfaz de comunicación

Enumera los requerimientos de los programas de comunicación que usará tu producto, como los emails o los formularios integrados.

#### **Notificaciones:**

- Notificaciones push para informar sobre el estado de los reportes.
- Notificaciones por correo electrónico para confirmar el envío de reportes y actualizaciones importantes.

#### **Formularios integrados:**

- Formularios de contacto para consultas y soporte técnico.
- Formularios de retroalimentación para mejorar la plataforma.

### 4.1 Seguridad

Incluye las normas de privacidad y protección de datos que deben cumplirse.

#### Privacidad de datos:

- Cumplir con la Ley de Protección de Datos Personales (si aplica) y otras normativas relevantes.
- Anonimizar los datos personales de los usuarios (si se permite el reporte anónimo).
- Implementar medidas de seguridad para proteger los datos de los usuarios contra accesos no autorizados.

### 4.2 Capacidad

Describe la demanda de almacenamiento actual y futura de tu software.

#### Almacenamiento de datos:

- Estimar la cantidad de datos que se generarán (reportes, fotos, videos) y planificar el almacenamiento necesario.
- Utilizar una base de datos escalable para manejar el crecimiento de los datos.

### 4.3 Compatibilidad

Enumera los requisitos mínimos de hardware para el software.

#### Dispositivos móviles:

- Asegurar la compatibilidad con las últimas versiones de Android e iOS.
- Optimizar la aplicación para diferentes tamaños de pantalla y densidades de píxeles.

#### Navegadores web:

- Asegurar la compatibilidad con los navegadores web más utilizados (Chrome, Safari, Firefox, Edge).

### 4.4 Confiabilidad

Calcula cuál es el tiempo de fallo crítico esperado de tu producto con relación a un uso normal.

#### Tiempo de actividad:

- Definir un objetivo de tiempo de actividad (por ejemplo, 99.9%).
- Implementar medidas para minimizar el tiempo de inactividad (redundancia de servidores, monitoreo constante).

## Requisitos no funcionales

### 4.5 Escalabilidad

Calcula la cantidad máxima de trabajo con la que tu software puede funcionar según lo esperado.

Diseñar la plataforma para que pueda manejar un aumento en el número de usuarios y reportes agregando más servidores.

### 4.6 Mantenibilidad

Describe cómo se debe usar la integración continua para implementar funciones o reparar errores rápidamente.

- Diseñar la plataforma con una arquitectura modular para facilitar el mantenimiento y la actualización.
- Documentar el código y la arquitectura de la plataforma para facilitar la comprensión y el mantenimiento.
- Implementar pruebas automatizadas para garantizar la calidad del software y facilitar la detección de errores.

### 4.7 Facilidad de uso

Describe qué tan fácil resultará el uso del software a los usuarios finales.

- Diseñar una interfaz de usuario clara y fácil de usar.
- Proporcionar ayuda y documentación en línea.

### 4.8 Otro

Enumera cualquier otro requisito no funcional.

Optimizar la aplicación para minimizar el consumo de batería en dispositivos móviles.



## Definiciones y acrónimos

<i><b>Acrónimo</b></i>	<b>Significado</b>
<i><b>API</b></i>	Application Programming Interface (Interfaz de Programación de Aplicaciones)
<i><b>CSRF</b></i>	Cross-Site Request Forgery (Falsificación de Petición en Sitios Cruzados)
<i><b>GPS</b></i>	Global Positioning System (Sistema de Posicionamiento Global)
<i><b>HTML</b></i>	HyperText Markup Language (Lenguaje de Marcado de Hipertexto)
<i><b>CSS</b></i>	Cascading Style Sheets (Hojas de Estilo en Cascada)
<i><b>JS</b></i>	JavaScript (Lenguaje de Programación para Interfaz Web)
<i><b>SMTP</b></i>	Simple Mail Transfer Protocol (Protocolo Simple de Transferencia de Correo)
<i><b>SMS</b></i>	Short Message Service (Servicio de Mensajes Cortos)
<i><b>SPA</b></i>	Single Page Application (Aplicación de Página Única)
<i><b>XSS</b></i>	Cross-Site Scripting (Inyección de Scripts en Sitios Cruzados)
<i><b>REST</b></i>	Representational State Transfer (Transferencia de Estado Representacional)
<i><b>MB</b></i>	Megabyte (Unidad de almacenamiento)
<i><b>GB</b></i>	Gigabyte (Unidad de almacenamiento mayor a MB)
<i><b>RAM</b></i>	Random Access Memory (Memoria de Acceso Aleatorio)
<i><b>Git</b></i>	Sistema de control de versiones distribuido
<i><b>Frontend</b></i>	Parte visual/interactiva de una aplicación que ve el usuario final
<i><b>Backend</b></i>	Parte lógica/servidor de una aplicación que gestiona la lógica y datos
<i><b>Browser</b></i>	Navegador Web (software que permite acceder a páginas web)