# Proyecto aplicativo móvil VerCity 1.0

Diagrama

El contenido generado por IA puede ser incorrecto.

**Proyecto**: Desarrollo de aplicación móvil y panel administrativo para reportes ciudadanos

**Versión:** 1.0

**Fecha:** 2025

**Equipo:** Scrum Team (PO, Scrum Master/Documentador Ágil Senior, Devs, QA, UX)

**Objetivo**

Facilitar la comunicación entre ciudadanos y administración para la gestión eficiente de fallas en servicios locales mediante una app móvil y panel administrativo con trazabilidad, métricas y canales de seguimiento.

**Alcance**

Módulos: App móvil (iOS/Android), Panel web admin, API, Base de datos, Integración de notificaciones, Dashboard.

Limitaciones: Integración con sistemas legacy municipal bajo alcance adicional.

**Requerimientos funcionales**

|  |  |  |
| --- | --- | --- |
| **Id. RF-01** | **Nombre.** Registro e inicio. | **Diagrama.** |
| **Resumen.** Registro e inicio de sesión de usuarios (correo, teléfono, Oauth opcional) | |
| Entrada. Datos del usuario (correo, teléfono o autenticación OAuth) | Criterios de aceptación.  - El usuario puede registrarse con correo o teléfono.- La autenticación OAuth funciona con Google/Facebook.- El token JWT expira correctamente.- Error adecuado si credenciales son incorrectas. |
| Proceso.  Validación de datos, creación de cuenta, generación de token JWT, manejo de sesión | Prioridad.  **Alta** |
| Salida.  Usuario registrado y autenticado correctamente | Alcance.  Módulo de autenticación y control de acceso |
| **Id. RF-02** | **Nombre.** Reportar incidencia. | **Diagrama.** |
| **Resumen.** Reportar incidencia con foto(s), descripción, categoría y ubicación (GPS o seleccionar en mapa) | |
| Entrada. Información del ciudadano (, nombre ciudadano, texto, foto, categoría, prioridad, ubicación GPS o seleccionada en mapa “o dirección”, descripción) | Criterios de aceptación.  - Se permite mínimo una foto y máximo tres.- Ubicación se guarda con coordenadas válidas.- Categoría y descripción obligatorias.- El sistema confirma el envío con notificación. |
| Proceso.  Validación y almacenamiento en base de datos; creación de registro “reporte”; enlace con usuario | Prioridad.  **Alta** |
| Salida.  Reporte almacenado con ID único y estado inicial “En revisión” | Alcance.  Módulo ciudadano (frontend app + backend de reportes) |
| **Id. RF-03** | **Nombre.** Clasificación automática | **Diagrama.** |
| **Resumen.** Clasificación automática preliminar por categoría (sugerencia). | |
| Entrada.  Texto del reporte y foto | Criterios de aceptación.  - Sugerencia visible al usuario antes de enviar.- Precisión mínima del 70% (en etapa piloto).- Puede modificarse manualmente por el usuario. |
| Proceso.  Algoritmo de categorización (IA o reglas) sugiere categoría probable | Prioridad.  **Media** |
| Salida.  Reporte preclasificado o con sugerencia de categoría | Alcance.  Servicio backend con modelo ML / motor de reglas |
| **Id. RF-04** | **Nombre.** Visualización | **Diagrama.** |
| **Resumen.** Visualización de estado del reporte (En revisión, En proceso, Resuelto, Rechazado). | |
| Entrada.  Comentario escrito o respuesta del operador | Criterios de aceptación.  - Cada comentario guarda autor, fecha, texto y notifica al receptor, foro  - Los comentarios se listan cronológicamente  - No se permiten mensajes vacíos ni ofensivos. |
| Proceso.  Registro de comentarios por reporte y notificación al otro actor | Prioridad.  **Alta** |
| Salida.  Conversación asociada al ticket | Alcance.  Módulo de mensajería interna (app + panel) |

|  |  |  |
| --- | --- | --- |
| **Id. RF-05** | **Nombre.** Comentarios y comunicación. | **Diagrama.** |
| **Resumen.** Comentarios y comunicación entre ciudadano y equipo gestor. | |
| Entrada.  Reporte marcado como resuelto por operador | Criterios de aceptación.  - Usuario recibe notificación y puede validar o rechazar cierre.  - Si confirma, estado final = RESUELTO.  - Si rechaza, estado vuelve a “En revisión”. |
| Proceso.  Envío de notificación al ciudadano para confirmar solución | Prioridad.  **Alta** |
| Salida.  Reporte pasa a estado “Resuelto” o “Reabierto” | Alcance.  Flujo de cierre y seguimiento de tickets |
| **Id. RF-06** | **Nombre.** Verificación / validación de cierre | **Diagrama.** |
| **Resumen.** Mecanismo de verificación / validación de cierre (usuario confirma solución). | |
| Entrada.  Credenciales de acceso y rol asignado | Criterios de aceptación.  - Ciudadano: puede crear y ver sus reportes.  - Operador: asignar, actualizar estado, comentar.  - Administrador: configurar, supervisar y exportar datos.  - Roles gestionables desde el panel admin. |
| Proceso.  Control de permisos y vistas según rol | Prioridad.  **Alta** |
| Salida.  Interfaces y permisos distintos por tipo de usuario | Alcance.  Sistema de control de roles y permisos |

**Requisitos no funcionales (RNF)**

|  |  |
| --- | --- |
| **Id. RNF-01** | **Nombre.** Escalabilidad |
| **Resumen.** Backend soporta al menos X usuarios concurrentes (definir en alcance). | |
| Entrada.  Carga simultánea de solicitudes | Criterios de aceptación.  - Soporta ≥ 5.000 usuarios simultáneos sin caídas.- Monitoreo de CPU/RAM ≤ 75% bajo carga normal.- Logs sin errores críticos durante prueba de estrés. |
| Proceso.  Escalado de backend (balanceo, microservicios) | Prioridad.  **Alta** |
| Salida.  Sistema mantiene rendimiento estable | Alcance.  Infraestructura backend (API + DB + balanceo) |
| **Id. RNF-02** | **Nombre.** Disponibilidad |
| **Resumen,** 99% SLA para API en horario operativo; tolerancia a fallos degradados. | |
| Entrada.  Métricas de disponibilidad y uptime | Criterios de aceptación.  - Uptime mensual ≥ 99%.- Recuperación automática ante fallo (failover).- Alertas configuradas en caso de caída. |
| Proceso.  Implementación de monitoreo y redundancia | Prioridad.  **Alta** |
| Salida.  API disponible ≥ 99% durante horas hábiles | Alcance.  Infraestructura cloud / hosting de API |
| **Id. RNF-03** | **Nombre.** Escalabilidad |
| **Resumen.** Backend soporta al menos X usuarios concurrentes (definir en alcance). | |
| Entrada.  Carga irregular o fallo parcial de servicios | Criterios de aceptación.  - API responde con modos degradados sin pérdida de datos.- Logs de fallos recuperables registrados.- Notificación automática al administrador. |
| Proceso.  Uso de colas, reintentos y mensajes asíncronos | Prioridad.  **Media** |
| Salida.  Sistema sigue funcionando aunque un servicio falle parcialmente | Alcance.  Arquitectura de resiliencia y recuperación |
| **Id. RNF-04** | **Nombre.** Rendimiento |
| **Resumen.** Tiempo de respuesta API < 300 ms para endpoints críticos bajo carga normal. | |
| Entrada.  Peticiones de usuarios | Criterios de aceptación.  - 95% de peticiones responden < 300 ms.- Logs de performance muestran cumplimiento continuo.- Test de carga semanal aprobado. |
| Proceso.  Optimización de consultas y caché | Prioridad.  **Alta** |
| Salida.  Respuesta ágil en la app móvil | Alcance.  Backend de API / endpoints críticos |
| **Id. RNF-05** | **Nombre.** Usabilidad |
| **Resumen.** Accesible WCAG AA mínimo para interfaces públicas. | |
| Entrada.  Elementos UI/UX | Criterios de aceptación.  - Contraste colores ≥ 4.5:1.- Textos alternativos para imágenes.- Navegación por teclado.- Cumple criterios WCAG AA. |
| Proceso.  Diseño y desarrollo accesible | Prioridad.  **Media** |
| Salida.  Interfaz usable para personas con discapacidad | Alcance.  Diseño de interfaz móvil y panel web |
| **Id. RNF-06** | **Nombre.** Portabilidad |
| **Resumen.** App compatible con Android 10+ y iOS 13+ (o especificar). | |
| Entrada.  Dispositivo móvil | Criterios de aceptación.  - Pruebas exitosas en Android 10–14 y iOS 13–17.- Sin errores críticos en instalación o arranque.- Ajuste de UI responsiva por tamaño de pantalla. |
| Proceso.  Compilación multiplataforma (Flutter/React Native) | Prioridad.  **Alta** |
| Salida.  Aplicación ejecutable y estable en OS soportados | Alcance.  Aplicación móvil multiplataforma (entorno cliente) |