PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE HÁBITAT, INFRAESTRUCTURA Y CREATIVIDAD

CARRERA DE: SISTEMAS DE INFORMACIÓN



TRABAJO DE INTEGRACIÓN CURRICULAR

Diseño y desarrollo de una plataforma geo colaborativa PWA para el reporte y visualización de incidencias urbanas y eventos comunitarios en el Distrito Metropolitano de Quito

AUTOR:

André Sebastián Arcentales Arciniega

TUTOR:

Juan Francisco Chafla Altamirano

QUITO DM, JUNIO DE 2026

Dedicatoria

Dedico este trabajo a mi familia, quienes han sido mi soporte en todo el proceso universitario. Gracias por su incondicional apoyo moral y económico, y por el gran esfuerzo que hicieron para que yo estudiara esta carrera. Agradezco su guía y educación durante todo este recorrido, lleno de enseñanzas y recuerdos que hicieron posible llegar hasta aquí.

Agradecimientos

Agradezco profundamente a todas las personas que me han apoyado en esta etapa. A mi familia, porque, como mencioné anteriormente, sin su respaldo no estaría aquí. A mis compañeros, por su ayuda invaluable para comprender conceptos difíciles y sacar adelante los proyectos; su compañía hizo que este camino fuera mucho más cálido y divertido. A mis profesores, por transmitirme los conocimientos técnicos necesarios para mi vida profesional y por los grandes consejos humanos que me han brindado a lo largo de estos años. Finalmente, a todos mis amigos y a las personas que han formado parte de mi vida durante este tiempo, por su compañía y comprensión constante.

Índice

Tabla de contenido

1 CAPÍTULO 1: Introducción 1

1.1 Tema 1

1.2 Justificación 1

1.3 Planteamiento del Problema 2

1.4 Objetivos 2

1.4.1 General 2

1.4.2 Específicos 3

1.5 Alcance 3

2 CAPÍTULO 2: Fundamentación teórica 5

2.1 Introducción 5

2.2 Información Geográfica Voluntaria (VGI) 6

2.2.1 Tipos de VGI 6

2.3 Crowdsourcing 7

2.4 Calidad 8

2.4.1 Medidas de Calidad 9

2.4.2 Indicadores de Calidad 12

2.5 Sesgos 16

2.5.1 Sesgo de Participación 16

2.5.2 Sesgo Temático y de Propiedad 17
2.5.3 Sesgo Arquitectónico y Barreras de UI/UX 17
2.5.4 Sesgo de Retención y el Ciclo de Retroalimentación 18
2.5.5 Sesgo de Conformidad Social y Contaminación de Datos 19
2.5.6 Sesgo por Ataques y Desinformación 19
2.5.7 Sesgo del Conocimiento Local 20
2.6 Motivación 20
2.6.1 Motivaciones Extrínsecas 22
2.6.2 Cómo Atraer y Retener Usuarios 23
2.7 Privacidad y Riesgos 24
2.7.1 Vigilancia Lateral 25
2.7.2 Vigilancia Participativa 25
2.7.3 Vigilantismo como Arma 26
2.7.4 El Dilema entre Privacidad y Utilidad 26
2.7.5 Estigmatización Social y el Mapa como Productor de Miedo 26
3 CAPÍTULO 3: Análisis y diseño del sistema 30
3.1 Metodología de Desarrollo 30
3.1.1 Justificación de la metodología seleccionada 30
3.1.2 ¿Qué es AUP? 31
3.1.3 Aplicación de AUP en el proyecto 32

3.2 Análisis de Requerimientos 33

3.2.1 Requisitos Funcionales 33

3.2.2 Requerimientos No Funcionales 36

3.3 Actores del Sistema 36

3.4 Historias de Usuario 38

3.5 Casos de Uso 41

3.6 Arquitectura del Sistema 44

3.6.1 Aplicaciones Web Progresivas (PWA): justificación de la elección tecnológica 47

3.6.2 Arquitectura offline-first 49

3.7 Diseño de la Base de Datos 51

3.8 Diagramas de Secuencia 53

3.8.1 Crear un reporte (con validación de ubicación) 54

3.8.2 Guardar sin internet y enviar al reconectar 55

3.8.3 Confirmar un reporte (validación ciega) 55

3.9 Diseño de la Interfaz de Usuario (UI/UX) 56

3.9.1 Usabilidad: medidas de Nielsen 58

3.9.2 Heurísticas para mapas web interactivos 60

3.9.3 Proceso de exploración y selección del diseño 64

3.9.4 Principios y lineamientos de interfaz 67

3.10 Decisiones de Diseño Relevantes 74

4 CAPÍTULO 4: Desarrollo e implementación 75

4.1 Entorno y herramientas de trabajo (Disciplina de Entorno) 75

4.1.1 Herramientas principales 75

4.2 Buenas prácticas y organización del código *(Entorno + Gestión de Configuración)* 76

4.2.1 Principios de ingeniería 76

4.2.2 Estándares de codificación 77

4.2.3 Gestión de Configuración (control de versiones) 77

4.3 Iteraciones de Construcción 77

4.3.1 4.3.1. Iteración 1 Fundación 78

4.3.2 4.3.2. Iteración 2 Identidad y Acceso 79

4.3.3 Iteración 3 Core Geoespacial (Mapa) 81

4.3.4 Iteración 4 Reportes Ciudadanos 83

4.3.5 Iteración 5 Visualización y Exploración 84

4.3.6 4.3.6. Iteración 6 PWA y Experiencia Offline 86

4.3.7 Iteración 7 Geocolaboración y Calidad del Dato 88

4.3.8 Iteración 8 Gamificación y Confianza Ciudadana 90

5 CAPÍTULO 5: Análisis de Pruebas 92

5.1 Estrategia de Pruebas 92

5.2 Pruebas Durante el Desarrollo (por iteración) 93
5.2.1 Resumen de lo verificado en las iteraciones principales 93
5.2.2 Resumen de pruebas (todas las iteraciones): 98
5.3 Pruebas Integrales Finales 99
5.4 Resultados, Problemas Encontrados y Correcciones 102
5.4.1 Resultados 102
5.4.2 Problemas encontrados y correcciones 103
5.5 Verificación de Calidad (Requisitos No Funcionales) 104
5.6 Pruebas de usabilidad 106
5.6.1 Introducción y objetivos 106
5.6.2 Metodología 106
5.6.3 Procedimiento 108
5.6.4 5.6.4 Resultados 109
5.6.5 Correcciones aplicadas 112
5.6.6 Conclusiones de las pruebas de usabilidad 113
6 CAPÍTULO 6: Conclusiones y recomendaciones 116
6.1 Conclusiones 116
6.1.1 Verificación de objetivos 116
6.1.2 6.1.2 Conclusiones generales 117
6.1.3 Limitaciones 118

6.2 Recomendaciones 118

6.2.1 Trabajo futuro técnico 118

6.2.2 Trabajo futuro de adopción 119

7 Referencias 120

Índice de Tablas

Tabla 1 Resumen de Iteraciones 35

Tabla 2 *Resumen RNF* 36

Tabla 3 *Permisos de Usuarios* 37

Tabla 4 Resumen RF 38

Tabla 5 *Tecnologias Usadas* 50

Tabla 6 *Paleta de Colores* 73

Tabla 7 *Resumen de Pruebas por Iteración* 98

Tabla 8 *Resumen de Pruebas Finales* 100

Tabla 9 *Resumen Resultados Pruebas* 102

Tabla 10 *Participantes Pruebas Usabilidad* 107

Tabla 11 *Tareas Pruebas de Usabilidad* 107

Tabla 12 *Tabla de Tiempos Pruebas de Usabilidad* 110

Tabla 13 *Errores por Tarea Pruebas de Usabilidad* 110

Tabla 14 *Puntaje SUS Pruebas de Usabilidad* 111

Tabla 15 Problemas Encontrados Pruebas de Usabilidad 111

Índice de Figuras

Figura 1 *Diagrama de Casos de Uso General* 42

Figura 2 *Diagrama de Arquitectura* 46

Figura 3 *Modelo de Base de Datos* 52

Figura 4 *Diagrama de Secuencia (Crear Reporte)* 54

Figura 5 *Diagrama de Secuencia (Offline)* 55

Figura 6 *Diagrama de Secuencia (Confirmar Reporte)* 56

Figura 7 *Dirección A UI* 66

Figura 8 *Dirección B UI* 66

Figura 9 *Dirección C UI* 67

Figura 10 *Pantalla Principa Modo Oscuro Modo Clarol* 69

Figura 11 *Creacion de Reporte UI* 70

Figura 12 *Creación de Reporte 2 UI* 71

Figura 13 *Detalle de Reporte UI* 72

1 CAPÍTULO 1: Introducción

1.1 Tema

Diseño y desarrollo de una plataforma geo colaborativa PWA para el reporte y visualización de incidencias urbanas y eventos comunitarios en el Distrito Metropolitano de Quito

1.2 Justificación

Este proyecto es importante porque busca agrupar la información geográfica crítica, actualmente dispersa en grupos de *Whatsapp*, publicaciones de Facebook o diferentes plataformas, en una única fuente de verdad que sea verificable y centralizada, creando así, una ciudad informada y comunicada con su entorno.

Para lograr el objetivo, el proyecto plantea el uso de tecnologías modernas y adaptadas para nuestro contexto. Se optó por la tecnología de PWA (Aplicación Web Progresiva), para facilitar su uso tanto en móviles como en computadoras, además de una arquitectura *offline-first*, especialmente en contextos donde la conectividad es intermitente o costosa (McIntosh et al., 2016; Biørn-Hansen et al., 2018) y un *stack* tecnológico moderno y estable (React, MapLibre, PostGIS). Además, también se dio énfasis a la motivación y calidad de los datos generados con mecanismos como reputación, gamificación, votación ciega, anti duplicado, entre otras.

Por último, el impacto del proyecto es significativo, ya que bajo el principio de los ciudadanos como sensores de su propio entorno (Goodchild, 2007), al informar a los ciudadanos sobre su entorno, pueden tomar mejores decisiones, tienen una participación activa con su vecindario, existe una mejor cohesión entre vecinos, crea una red de inteligencia colectiva que mejora la calidad de los datos, agrupa reportes de eventos positivos y negativos dejando de lado

estigmatizaciones, a diferencia de plataformas centradas únicamente en el crimen, que tienden a elevar la percepción de peligro y estigmatizar sectores (Riddell, 2023), y centralizando toda la información en una misma plataforma, y también da una vía para la visualización de los mayores problemas que se tienen en los diferentes sectores de Quito para una posible intervención futura.

1.3 Planteamiento del Problema

La información crucial para la vida urbana en Quito se encuentra actualmente fragmentada en diversas plataformas como redes sociales y grupos de mensajería, sin un contexto geográfico preciso. Esta dispersión impide que los ciudadanos tengan una conciencia situacional completa de su entorno, lo que afecta negativamente su capacidad para tomar precauciones de seguridad, planificar sus desplazamientos de manera eficiente y participar en la vida comunitaria. El problema central radica en la inexistencia de una herramienta que centralice el reporte de incidencias críticas, tales como luminarias dañadas, aceras en mal estado, o alertas por robos y zonas de peligro, y que a su vez permita la visibilización de eventos positivos, incluyendo reuniones sociales y eventos comunitarios, que son esenciales para la participación ciudadana.

1.4 Objetivos

1.4.1 General

Desarrollar una Aplicación Web Progresiva (PWA) accesible desde dispositivos móviles que brinde una experiencia similar a una aplicación nativa, pero con las ventajas de una página web, que permita a los ciudadanos del Distrito Metropolitano de Quito reportar y visualizar de forma geolocalizada tanto incidencias negativas como eventos positivos, con

el fin de habilitar las condiciones para fomentar una comunidad más informada y participativa.

1.4.2 Específicos

1. Fundamentar teóricamente el uso de plataformas geo colaborativas y tecnologías PWA para la gestión de incidencias urbanas
2. Analizar los requerimientos funcionales y no funcionales del sistema para definir las características esenciales de la plataforma.
3. Diseñar la arquitectura de la información y las interfaces de usuario (UI/UX) para la aplicación responsiva, priorizando la usabilidad para el registro y la visualización de reportes en un mapa interactivo
4. Implementar el sistema *back-end* para la gestión de datos geolocalizados y desarrollar la aplicación cliente que permitan a los usuarios crear, visualizar y filtrar reportes ciudadanos.
5. Validar el sistema mediante pruebas técnicas para probar el cumplimiento de los requisitos definidos, y pruebas con usuarios para validar la usabilidad de la aplicación.

1.5 Alcance

El alcance del proyecto SÍ INCLUYE:

- Una única aplicación web responsiva: Se desarrollará una plataforma web que funcione de manera óptima tanto en navegadores de escritorio como en dispositivos móviles.
- Funcionalidades del MVP (Producto Mínimo Viable):
    - Registro y autenticación de usuarios.
    - Creación de reportes geolocalizados, permitiendo al usuario seleccionar una categoría, añadir una breve descripción y adjuntar una fotografía.

- Visualización de todos los reportes públicos en un mapa interactivo con filtros básicos por categoría
- Un *back-end* simplificado: Se implementará la lógica de servidor necesaria para gestionar usuarios y almacenar los datos de los reportes de forma segura.
- Pruebas de usabilidad: Se realizará una validación de usabilidad y concepto con un grupo reducido y específico de usuarios
- Funcionamiento offline y sincronización automática
- Geo colaboración (votación ciega, anti-duplicados, comentarios)
- Gamificación (insignias, perfil de reputación)

Para ser viable, el alcance NO INCLUYE:

- Una aplicación móvil nativa (para iOS o Android): La aplicación web será la única plataforma de acceso.
- Funcionalidades avanzadas: Se excluyen notificaciones en tiempo real, validación de reportes por IA
- Módulo de administración con interfaz gráfica: No existirá una interfaz para la moderación, todo usarán reglas automáticas
- Envío real de correos electrónicos: En este MVP, solamente se simularán
- Integración con sistemas externos: No se contempla la conexión con APIs de entidades gubernamentales, servicios de emergencia o plataformas de terceros.

2 CAPÍTULO 2: Fundamentación teórica

2.1 Introducción

Históricamente, todos los datos hechos de manera geográfica han sido creados por expertos, por geógrafos. Esto ha hecho que la información geográfica solo la puedan crear unos pocos, con información limitada, y con posibles sesgos en la información.

Pero todo esto cambió con todas las facilidades que nos brinda la tecnología, dando paso a la *neogeografía*. Este concepto habla sobre cómo los inexpertos, los usuarios, pueden generar su propia información; dicho por Sui (2008, como se citó en Flanagin & Metzger, 2008, p. 138), una *"geografía sin geógrafos"*, dejando de lado a los expertos. Con estas nuevas herramientas se mejora enormemente la calidad y cantidad de información geográfica.

Con el proyecto de titulación propuesto, se plantea justamente esto: hacer que los ciudadanos puedan producir su propia información geográfica, ayudándose entre sí y creando una red de información en la que puedan confiar y de la cual puedan formar parte. De esta manera, se transforma la forma en la que se recopilan los datos; ya no se depende únicamente de sensores tradicionales como GPS o cámaras, sino que se incorpora un sensor mucho más valioso, numeroso e inteligente: los propios seres humanos.

Esta diferencia en quién genera los datos no solo es una cuestión tecnológica, sino que también tiene su enfoque político. Ya no se obtiene información de medios tradicionales, centralizados y con gran poder (*top-down*); ahora la información va a fluir desde los mismos ciudadanos (*bottom-up*). Este enfoque cambia la forma en que los ciudadanos ven la información, percibiendo como más confiable, ya que ciertos datos solo pueden ser conocidos y comunicados por personas locales, quienes viven y entienden el contexto real de su propio entorno.

2.2 Información Geográfica Voluntaria (VGI)

Este tipo de información fue tan importante que tuvo su propio nombre: la VGI (Información Geográfica Voluntaria). Según Goodchild (2007), se refiere a la creación de información geográfica por parte de ciudadanos, a menudo sin formación específica e independientemente de su experiencia o conocimientos.

En una tipología de VGI, los trabajos de Antoniou et al. (2010) y Craglia et al. (2012) (como se citó en Senaratne et al., 2017) clasificaron la VGI según el tipo de geografía explícita/implícita. En la VGI explícita, los colaboradores se centran en actividades de mapeo. Los datos que se asocia implícitamente con una ubicación geográfica pueden ser cualquier tipo de medio: texto, imagen o vídeo que se refiera o esté asociado con una ubicación geográfica específica. Por ejemplo, *microblogs* geolocalizados (como los tuits), imágenes geolocalizadas de *Flickr* o artículos de *Wikipedia* que hacen referencia a ubicaciones geográficas.

Harvey (2013, como se citó en See et al., 2016) también hace una diferencia entre la VGI y la CGI (*Contributed Geographic Information*): mientras que la VGI se obtiene por una participación voluntaria y una decisión explícita del usuario, la CGI se obtiene por datos pasivos recopilados sin una decisión explícita (como la recopilación de la ubicación en los teléfonos celulares). En el contexto del presente proyecto de titulación, se utiliza VGI en sentido estricto (no CGI), donde los ciudadanos toman la decisión explícita de compartir datos geográficos, respetando en todo momento la privacidad de los usuarios.

2.2.1 Tipos de VGI

Dentro de la VGI, Senaratne et al. (2017) clasifican la información según su tipo de dato. A continuación, se detallan los tres tipos que utilizará el proyecto de titulación:

- **VGI basado en mapas:** Abarca fuentes que generan geometrías formales como puntos, líneas y polígonos.
- **VGI basado en imágenes:** Consiste en la captura de fotografías de algún objeto geográfico a las que se les añade una referencia geoespacial.
- **VGI basado en texto:** Corresponde a la información que se puede referenciar en el propio texto o que se asocia a la geolocalización del lugar donde se creó el mismo.

El proyecto de titulación utilizará estos tres tipos de VGI: se usará el mapa para marcar la ubicación del reporte a generar (VGI basado en mapas); se emplea el VGI basado en imágenes, de forma opcional, para agregar una fotografía al reporte; y se utilizará el VGI basado en texto para brindar una descripción y contexto detallados de la situación reportada. En síntesis, el proyecto de titulación se enmarca en la VGI explícita, a diferencia de la CGI, donde los ciudadanos deciden voluntariamente compartir datos geográficos en estos tres formatos complementarios.

2.3 Crowdsourcing

El *crowdsourcing* se describe como la externalización de una tarea tradicional mediante una convocatoria abierta a un grupo numeroso pero indefinido de personas (Howe, 2006, como se citó en Pak et al., 2017). Pak et al. (2017) añaden que este concepto ha evolucionado a lo largo del tiempo, pero todas las definiciones tienen en común que representan acciones participativas organizadas para un beneficio mutuo. De este modo, quienes coordinan o participan en el *crowdsourcing* se benefician de lo producido colectivamente y utilizan el resultado para resolver problemas comunes (Brabham, 2008; Estellés-Arolas & González-Ladrón-de-Guevara, 2012, como se citó en Pak et al., 2017).

El *crowdsourcing* aprovecha a las personas interesadas en generar los datos, cosa que de otra forma hubiera sido muy difícil. Esto es especialmente importante en el contexto de este proyecto de titulación, ya que como menciona Erickson (2010, como se citó en Pak et al., 2017), los mayores beneficiados de esto son los propios ciudadanos, y las mejores personas para generar los datos, debido al entendimiento de su entorno, también son los mismos ciudadanos, por lo que se crea un círculo de ayuda que poco a poco se va retroalimentando y mejorando la vida de los propios ciudadanos.

En el presente proyecto de titulación se utiliza exactamente este concepto. Esta es la metodología por la cual los usuarios van a tener interés en generar los datos: un grupo de personas (los usuarios) genera reportes para un beneficio mutuo, usando esos mismos reportes para estar informados de su entorno. Asimismo, paralelamente, puede generar un beneficio para comunicar y puntualizar ciertos problemas que se tienen en las comunidades, para que los ciudadanos o las mismas autoridades puedan evaluar los mayores problemas existentes en cada localidad.

2.4 Calidad

El hecho de que los datos sean generados por ciudadanos sin formación específica plantea una pregunta importante: ¿qué tan confiables son estos datos? Flanagin y Metzger (2008) ofrecen un enfoque diferente que sirve de introducción para los distintos puntos de vista de calidad que sostienen diversos autores. Estos señalan que, en este tipo de sistemas, los conceptos de calidad difieren significativamente de los tradicionales, ya que utilizan la *inteligencia colectiva* a su favor. Mencionan que nociones como calidad o precisión no se pueden evaluar de manera tradicional al no existir una fuente de verdad única y absoluta, por lo que la credibilidad ya no reside en autoridades tradicionales, sino que se construye mediante evaluaciones entre

pares. Y aunque no todos los autores son tan drásticos y siguen teniendo en cuenta ciertos criterios tradicionales, la inteligencia colectiva siempre se mantiene como un pilar fundamental.

En concordancia con esta visión, See et al. (2016) sostienen que varios estudios comparativos demuestran que la información geográfica generada mediante *crowdsourcing* puede ser tan precisa y útil, o incluso mejor, que la información producida por autoridades oficiales. No obstante, estos autores también destacan la importancia de mantener un equilibrio entre la calidad de los datos y el interés de los usuarios: a mayor número de controles de calidad implementados, mayor es la fricción que experimenta el usuario. Por lo tanto, en el diseño del sistema se debe hallar un punto óptimo entre una experiencia de usuario fluida y los controles de calidad necesarios.

Una forma de ver la calidad nos la da Senaratne et al. (2017). Primero explica que la calidad en VGI es muy variable, ya que los datos son producidos por un sinfín de personas diferentes, usando medios diferentes y con objetivos diferentes, por lo que cada persona genera datos con diferente nivel de detalle, dirección, precisión, propósitos y filtros. Esto hace muy difícil medir la calidad de los datos que se generan. Para ello, Senaratne separa en dos grandes grupos de indicadores para medir la calidad en VGI: las **medidas de calidad** y los **indicadores de calidad**.

2.4.1 Medidas de Calidad

Estas medidas son objetivas y cuantificables si se dispone de datos para comparar con una "verdad de terreno" u organismos oficiales. Estos indicadores se alinean con los estándares de la ISO (Organización Internacional de Normalización, por sus siglas en inglés: International Organization for Standardization) y se fundamentan en el análisis de los errores numéricos y geométricos existentes en los datos.

2.4.1.1 Completitud

La completitud describe la relación entre los objetos representados del mundo real y sus conceptualizaciones en la base de datos, midiéndose a través de la ausencia de información (errores de omisión) o la presencia de datos en exceso (errores de comisión). Cualquiera de estos dos escenarios es propenso a ocurrir en el proyecto de titulación. Por un lado, los errores de omisión pueden presentarse debido a una baja participación o poca generación de datos por parte de los usuarios. Por otro lado, los errores de comisión se evidenciarían si se crean múltiples reportes para un mismo incidente en una misma ubicación. Para mitigar esto, el sistema propuesto debe sugerir al usuario confirmar un reporte preexistente en lugar de duplicarlo, permitiendo consolidar la información en un único registro verificado.

2.4.1.2 Consistencia

La consistencia se refiere a la coherencia lógica en las estructuras de los datos espaciales digitalizados. Con el fin de garantizarla, se requiere realizar validaciones automáticas antes de almacenar los registros en la base de datos, implementando controles tanto en el *frontend* como en el *backend* que ofrezcan comentarios (*feedback*) claros al usuario para corregir posibles errores de ingreso. Más específicamente, Senaratne et al. (2017) distinguen cuatro categorías de consistencia, de las cuales dos resultan críticas para este sistema:

- **Consistencia conceptual:** Asegura que los datos sigan estrictamente las reglas lógicas y relaciones de diseño del sistema.
- **Consistencia de dominio:** Garantiza que los valores de los datos ingresados se encuentren dentro de los rangos lógicos preestablecidos (tales como formatos de fechas, coordenadas geográficas de ubicaciones, etc.).

2.4.1.3 Precisión

La precisión se refiere al grado de proximidad entre una medición de una cantidad y el valor verdadero aceptado de esa cantidad, y se presenta en forma de precisión posicional, precisión temporal y precisión temática:

- **Exactitud posicional:** Qué tan cercana es la coordenada medida en la plataforma y la posición real del evento.
- **Exactitud temática:** Corrección en la clasificación del evento o incidencia (ej. clasificar correctamente un bache frente a un daño de acera).
- **Exactitud temporal:** Validez y orden secuencial correcto de los eventos reportados.

La precisión es un aspecto clave en el proyecto de titulación. En cuanto a la precisión posicional, se contemplan varias alternativas, siendo la primera y principal el uso del GPS del dispositivo del usuario, una herramienta altamente precisa cuando el reporte se genera en el mismo lugar de los hechos. En caso contrario, se ofrecen opciones para que el usuario marque la ubicación exacta mediante un mapa interactivo en 3D y confirmación textual del nombre del lugar. En cuanto a la exactitud temática, se clasifican los reportes en categorías visualmente claras con colores y emojis distintivos, facilitando al usuario su identificación inmediata y abarcando tanto incidentes (baches, robos) como eventos comunitarios (ferias barriales, reuniones). Por último, respecto a la exactitud temporal, la cual resulta fundamental para registrar el momento de un incidente o planificar un evento programado, el sistema registra automáticamente la marca de tiempo de cada reporte y la muestra de manera amigable.

2.4.2 Indicadores de Calidad

Al igual que los anteriores, estos siguen el mismo marco ISO pero se los separa por su naturaleza cualitativa:

- **Propósito:** El uso previsto para el cual se crearon los datos. Se debe asegurar que el proyecto de titulación solamente se use para el propósito que tiene y no se use para otros fines; para ello se implementan las categorías que guían al usuario en cómo se puede usar, aunque tomando en cuenta que los usuarios tienen total libertad y puede darse el caso de que se agregue algún reporte que no esté alineado con el propósito de informar a la ciudadanía.
- **Uso:** Los contextos o sistemas en los cuales los datos ya han sido utilizados. En el caso de la presente plataforma, los datos serán consumidos únicamente por ciudadanos.
- **Linaje:** Describe la historia de los datos (desde su recolección y adquisición hasta su estado actual). En el proyecto de titulación, esto se refiere a la auditoría que se puede hacer de los reportes, desde la recolección (la generación del reporte) y modificación (la edición de algún reporte). Según el autor y la ISO, se deben documentar estas fases en el sistema, mostrando a los usuarios la fecha de creación, fecha de modificación, y cuántos usuarios han confirmado, para tener una trazabilidad del dato y mayor confianza en el reporte.

Senaratne et al. (2017) también ofrece indicadores de calidad más abstractos, los cuales son más subjetivos y surgieron de la necesidad de medir la calidad cuando los métodos ISO no son aplicables o no son suficientes:

- **Vaguedad:** La vaguedad se refiere a la falta de precisión semántica con la que se capturan los datos. Es algo muy común en el VGI y aplicable al sistema. En todos los tipos de datos pueden surgir cierta ambigüedad: en las imágenes puede existir baja resolución o mal enfoque, en el texto puede existir poca descripción del reporte, y en la ubicación puede marcarse con poca precisión. Para corregir esto, se puede aplicar un mínimo de caracteres para el reporte del evento y una confirmación total de todos los datos a registrar, para que el usuario, antes de generar el reporte, pueda verificar si los datos subidos son correctos.
- **Conocimiento local:** Se refiere al conocimiento que tienen los contribuyentes (los usuarios) sobre el entorno que están reportando. El proyecto de titulación está enfocado en generar reportes en tiempo real, con foto y ubicación actual, por lo que la mayoría de los reportes generados serán por personas que tienen un gran conocimiento local, ya que viven o pasan frecuentemente por esos lugares.
- **Experiencia:** Se refiere a la participación de un contribuyente (usuario) con la plataforma. Senaratne et al. (2017) señala que esto puede medirse tanto por el tiempo que el usuario está en la plataforma como por la cantidad de registros que ha realizado. En el perfil se podría mostrar una sección de la experiencia del usuario, donde se pueda ver de un vistazo el tiempo que lleva registrado en el sistema, el número de registros que ha hecho y el número de confirmaciones que ha realizado. Con ello se asegura mostrar la experiencia del usuario y resaltar a los usuarios con más experiencia.
- **Reconocimiento:** Es dar cierta acreditación a un contribuidor basado en todos los datos generados, tanto de reportes propios como de revisiones de pares. En el proyecto de titulación se podrían generar ciertos logros o insignias según la generación de contenido

que realice el usuario; por ejemplo, se podría agregar cierto reconocimiento a los usuarios que hayan realizado "50 reportes" o "10 confirmaciones", o por la reputación que tienen.

- **Reputación:** Según Maué (2007, como se citó en Senaratne et al., 2017), la reputación se evalúa por un historial de interacciones pasadas o por las calificaciones dadas por otros usuarios. Esta idea de reputación encaja perfectamente con el punto anterior; se pueden generar ciertos reconocimientos en base a la reputación del usuario, generando varios tipos de reconocimientos (insignias/logros) visibles para los demás usuarios, para reconocer fácilmente a personas cuyos reportes han sido ampliamente validados.
- **Confianza:** Se basa en características subjetivas como la fiabilidad percibida del contribuyente, evaluada a partir de buenas calificaciones y una alta frecuencia de uso. En el proyecto de titulación, el perfil del usuario mostraría la frecuencia de uso, el número de reportes y las calificaciones positivas acumuladas, de modo que otros ciudadanos puedan evaluar la fiabilidad de quien generó cada reporte.
- **Calidad de contenido textual:** Para medir la calidad que tiene un texto en el contexto del VGI, se puede basar en las características principales que mencionan Senaratne et al. (2017): la longitud del texto, la estructura, el estilo, la legibilidad, el historial de revisiones, la similitud temática, el uso de terminología técnica, etc. En el proyecto de titulación se aplicará un control básico de longitud mínima en la descripción del reporte, dejando como trabajo futuro la implementación de análisis más sofisticados de legibilidad o detección de spam textual.
- **Credibilidad:** Muchas comunidades científicas consideran que la credibilidad es una propiedad objetiva de la calidad, sobre todo en áreas como la ciencia de la información. Pero en otras áreas como la comunicación y la psicología, tratan a la credibilidad como

una variable subjetiva que se basa en la percepción de cada individuo. Flanagin y Metzger (2008) dividen la credibilidad en dos dimensiones: por un lado, una dimensión subjetiva, la confiabilidad; y por otro, una dimensión objetiva, la experiencia. Además, añaden que para poder medir la experiencia se pueden evaluar variables como precisión, autoridad, competencia o credenciales de la fuente. La credibilidad funciona como un concepto integrador: la confiabilidad se construye a través de los mecanismos de confianza y reputación, mientras que la dimensión objetiva se acerca a los indicadores de experiencia y linaje. El proyecto de titulación no válida autoridad ni credenciales formales de los usuarios, por lo que la credibilidad se construye a partir de la combinación de estos indicadores y no mediante un mecanismo adicional.

Más allá del marco conceptual de Senaratne et al. (2017), See et al. (2016) confirman que una gran cantidad de los conceptos mencionados ya han sido aplicados a plataformas funcionales, con herramientas concretas para validar estos indicadores: verificaciones automatizadas, revisión y ranking entre pares, el uso de observaciones múltiples sobre un mismo sitio como mecanismo de verificación cruzada, y en casos más especializados, la revisión por expertos. Todo esto confirma que los mecanismos propuestos para la plataforma tienen respaldo en sistemas ya existentes, lo que fortalece su viabilidad de implementación.

Por último, estos autores recalcan la importancia de ciertos temas abordados, señalando que la comunidad científica tiene como prioridad el desarrollo de metodologías de credibilidad, reputación y confianza aplicadas a este tipo de información.

2.5 Sesgos

Pak et al. (2017) señalan que los mapas colaborativos como el de la plataforma propuesta nunca serán neutrales; siempre habrá cierto sesgo en la información según quién utiliza el proyecto de titulación, quién lo conozca y tenga acceso a la tecnología. Según un estudio realizado por los mismos autores, llegaron a la conclusión de que existía marginación para ciudadanos de ciertas etnias y personas con menores ingresos.

Todos estos son sesgos que se deben tomar en cuenta a la hora de desarrollar el proyecto de titulación, siendo conscientes de que no va a mapear todos los incidentes o eventos posibles y que ciertos datos generados estarán sesgados por cuestiones que están fuera del control del sistema.

2.5.1 Sesgo de Participación

Fritz et al. (2017) advierten sobre una regla que afecta directa y universalmente a todas las aplicaciones cuyo contenido es generado por el usuario. Esta regla, denominada la regla del 1% (Nielsen, 2006, como se citó en Fritz et al., 2017), establece que el 90% del contenido es generado por el 1% de los usuarios, mientras que el 9% restante solo lo genera a veces, y el 90% de usuarios nunca genera ningún contenido.

Estas cifras pueden cambiar de aplicación en aplicación, pero esta regla sirve para tener en cuenta que solo el 1% de los usuarios generará la mayoría del contenido, mientras que la gran mayoría no generará nunca nada. La mayoría de los usuarios que tendrá el sistema solo va a usar el proyecto de titulación para visualizar incidentes, y esto no es un problema ni un fallo del proyecto de titulación.

Los estudios hechos tanto en *Wikipedia* como en *OpenStreetMap* (referente mundial en mapas) dieron resultados similares, donde el 98% del contenido era generado solo por el 3.5% de los usuarios (Neis et al., 2011, como se citó en Fritz et al., 2017).

Siguiendo la misma línea, estos datos advierten que habrá cierto sesgo en los datos, ya que la generación de reportes del sistema va a estar centrada en solo un pequeño porcentaje de usuarios, lo que resultará en reportes con un gran sesgo hacia estas pocas personas generadoras.

Los sesgos que se describen a continuación aplican, por tanto, principalmente a este grupo reducido de usuarios generadores.

2.5.2 Sesgo Temático y de Propiedad

Un hallazgo importante y lógico es que los datos estarán generados solamente por las personas que tienen interés en ello. Pak et al. (2017) ponen el ejemplo de un bache: este será principalmente reportado por las personas que tienen un automóvil, y será mayormente ignorado por los demás.

Por lo tanto, es importante saber el segmento que va a usar el proyecto de titulación, ya que cada segmento tendrá sus propias prioridades y sesgar la información a sus intereses. Con categorías amplias se podría incluir a la gran mayoría de segmentos, aunque con el uso y estadísticas del sistema se podría ver claramente cómo ciertas categorías se van llenando de reportes mientras otras no se usan tanto, algo que no se puede evitar ni predecir en el tiempo y contexto actuales.

2.5.3 Sesgo Arquitectónico y Barreras de UI/UX

Este es un sesgo que sí se puede controlar y mejorar. Por un lado, Pak et al. (2017) señalan que el costo de un teléfono celular y los planes de datos pueden llegar a ser costosos para un segmento de la población y, por lo tanto, excluirlo.

Para ello se aplican diferentes estrategias. Primero, el principal sesgo es el *hardware*: no todos cuentan con un PC o un celular de última generación, lo que podría excluir a un gran segmento de la población. Para ello, el sistema debe ser eficiente y poder usarse en celulares sin consumir muchos recursos. Por esto se optó por una PWA (cuyas ventajas se explicarán a profundidad más adelante), lo cual permite usar el sistema sin descargar nada y poder usarlo desde un celular móvil. Además, se debe asegurar que este proyecto de titulación sea eficiente y pueda rendir en todos los tipos de celulares.

Por último, se deben tomar en cuenta a las personas que no tienen planes de datos o que tienen una conexión limitada. Para no excluir a este segmento, se aplicará una arquitectura *offline-first*, la cual permitirá usar el proyecto de titulación sin internet, y cuando se tenga señal, los datos se suben y descargan automáticamente.

2.5.4 Sesgo de Retención y el Ciclo de Retroalimentación

También existe un sesgo inevitable en el proyecto de titulación. Un estudio de Sjoberg et al. (2015, como se citó en Pak et al., 2017) revela que el primer problema resuelto en la plataforma eleva un 54% la probabilidad de participar nuevamente; como por ejemplo reportar un bache y luego tenerlo arreglado.

Esto está fuera del alcance del sistema, ya que no se conecta con ningún organismo ni con el gobierno para poder corregir ciertos problemas. Lo único que hace el proyecto de titulación es poder informar a los ciudadanos y dar a visualizar ciertos problemas que, con la visibilización necesaria, se lleguen a resolver. El proyecto de titulación podría convertirse en un medio de denuncia de ciertos problemas en la comunidad para que las autoridades los resuelvan, pero la implementación y viabilidad de esto queda por fuera del alcance, y se debe entender

como una limitación que posiblemente afecta negativamente a la motivación y cantidad de usuarios contribuyentes.

2.5.5 Sesgo de Conformidad Social y Contaminación de Datos

Flanagin y Metzger (2008) vienen a refutar una idea que se tenía sobre la inteligencia colectiva y la sabiduría de masas. Estos autores explican que estos conceptos son correctos cuando los datos generados individualmente son precisos. Pero si los primeros reportes son imprecisos, esto genera una ola que poco a poco arrastra el error, ya que las opiniones generadas por los usuarios dependen fuertemente de ese primer reporte.

Este sesgo es un problema que se va a perpetuar con las elecciones hechas sobre la reputación, validación, reconocimiento, etc. Se están dando más vías a los usuarios para generar este efecto de arrastre, y se debe tomar en cuenta a la hora de analizar los datos generados por la plataforma. Como mitigación parcial, se contempla que las confirmaciones entre pares sean independientes y no muestran al usuario las calificaciones previas antes de emitir la suya, reduciendo el efecto de arrastre en la validación inicial de cada reporte.

2.5.6 Sesgo por Ataques y Desinformación

El impacto de este sesgo puede llegar a ser crítico en el sistema, un riesgo que Zhou et al. (2021) advierten. Como ejemplo, los autores informan sobre un ataque realizado en la plataforma de *Waze* y diferentes formas de manipulación y desinformación, cómo simular atascos e incidentes para disminuir la eficacia de la plataforma. Este es un problema real que en otras plataformas similares tiene un impacto enorme, tanto en el sistema como en la vida de los usuarios.

Contrarrestar estos ataques es una tarea difícil, y Zhou et al. (2021) menciona la dificultad que esto significa para todas las plataformas similares. La implementación de

mecanismos de detección de comportamiento automatizado y antispam queda propuesta como trabajo futuro dentro del sistema.

2.5.7 Sesgo del Conocimiento Local

La motivación es un factor clave en los sistemas con contenido generado por el usuario, y esta motivación también puede crear sesgos. Coleman et al. (2009, como se citó en Fritz et al., 2017) mencionan que las personas son motivadas a reportar mayormente en sus áreas locales, todo esto impulsado por un "orgullo del lugar" (*pride of place*). También comentan que para que los usuarios generen datos fuera de su entorno, se deben generar dinámicas comunitarias o causas humanitarias.

Esto generaría el sesgo de que sectores con poco sentido de pertenencia o cohesión social tuvieran menos participación, y la única forma de contrarrestar este sesgo sería por medio de actividades comunitarias, algo que está fuera del alcance del proyecto de titulación.

La información generada por la plataforma nunca será un reflejo completo ni neutral de la realidad. Algunos sesgos, tales como el arquitectónico o el de conformidad social, pueden mitigarse desde el diseño del sistema, mientras que otros, como el de participación, el temático o el de retención, son limitaciones estructurales de cualquier plataforma de contenido generado por el usuario.

2.6 Motivación

¿Por qué los ciudadanos estarían dispuestos a gastar mucho tiempo en crear contenido para sitios VGI? Esta fue la pregunta que planteó Goodchild (2007), y cuya respuesta es clave para la viabilidad de cualquier plataforma de este tipo. Según Goodchild (2007), una de las motivaciones principales es poder ver un mosaico creciendo a partir de los datos que se van generando. Ver en el proyecto de titulación cómo el mapa se va llenando de reportes es una

motivación que hace sentir al usuario parte de algo más grande y que está contribuyendo con la sociedad.

Budhathoki y Haythornthwaite (2012, como se citó en Fritz et al., 2017) clasificaron las motivaciones en dos grandes categorías: la motivación intrínseca, que se da cuando la motivación nace del propio usuario; y la extrínseca, que nace del exterior, como incentivos financieros o funciones de reputación y reconocimiento. Entre los factores intrínsecos identificados por Fritz et al. (2017) que resultan más relevantes para el proyecto de titulación propuesto, se destacan los siguientes:

- ***Unique ethos*** **(ethos único):** Es la idea de que los datos que se generan sean de bien público y de conocimiento general, no contenido cerrado.
- **Altruismo:** Colaborar por el hecho de ser parte de una causa social, ayudar a los demás. En el caso del proyecto de titulación, sería ayudar a los demás ciudadanos a estar más informados de su entorno y hacerlos más participativos y parte de la comunidad.
- **Instrumentalidad:** Proveer datos que de otra manera serían erróneos o nulos. Los usuarios estarían motivados a generar reportes que de otra forma no existirían, o a mejorar la información dada por reportes de otros usuarios. Esto se vincula directamente con la autoexpresión y el orgullo local ya comentados.
- **Autoeficacia:** Sentir que se está siendo eficaz al generar información. La plataforma puede dar estadísticas a los usuarios para motivarlos, indicándose cuántas personas han visto su reporte.
- **Aprendizaje:** Uno de los aspectos principales de la plataforma: ganar nuevos conocimientos. La plataforma se enfoca en generar conocimiento para los ciudadanos.
- **Enriquecimiento personal:** La satisfacción al contribuir.

- **Libertad de expresión:** Sentir la libertad de escoger qué información y de qué manera generarla. Por ello, el tener categorías amplias motiva a los usuarios al darles libertad.

2.6.1 Motivaciones Extrínsecas

En cuanto a los factores extrínsecos, Fritz et al. (2017) identifican los siguientes como los más aplicables al contexto de la plataforma:

- **Fortalecimiento de relaciones sociales:** Uno de los puntos fuertes del proyecto; se refiere a fortalecer relaciones sociales. Esto se puede lograr mediante eventos comunitarios que se sabe existirán en la plataforma. Así, las comunidades y los vecinos se unirán más y tendrán una mejor cercanía.
- **Comunidad:** Sentirse parte de algo más grande. Ver cómo los reportes ayudan y colaboran a llenar de información a toda la localidad, ayudando a los demás ciudadanos a estar comunicados.
- **Reputación:** Reconocimiento del sistema o por individuos en la comunidad. Esto, como se mencionó anteriormente, puede llegar a ser beneficioso tanto para crear credibilidad como para generar motivación en los usuarios.
- **Reciprocidad:** La idea de que si uno participa, los demás participarán. Esto puede generar una ola, ya que más personas estarán interesadas en el proyecto de titulación si este tiene la información necesaria.
- **Confianza en el sistema:** Si el sistema es seguro, los usuarios contribuirán. Para ello se debe generar confianza tanto en los datos como en el sistema, usando las estrategias vistas en la sección de calidad y aplicando una buena UI/UX que se profundizará más adelante.

2.6.2 Cómo Atraer y Retener Usuarios

Para atraer a nuevos usuarios, Fritz et al. (2017) señalan que se deben abordar dos preguntas fundamentales: cómo se va a reclutar y cómo se va a motivar y retener. En cuanto al reclutamiento, Fritz et al. (2017) advierten que lo más importante es no hacer una convocatoria a un público general, sino escoger bien un *target* o comunidades ya existentes y enviar mensajes sobre los beneficios que tienen al participar. Además, también resaltan otras alternativas más eficaces. La más importante es hacer campañas en medios de comunicación como las redes sociales, lo cual resulta muy efectivo para generar un impacto inicial muy grande, pero advierten que debe combinarse con otras técnicas para retener, ya que su impacto decrece muy rápidamente. Otra alternativa son los eventos de mapeo comunitario. Aunque esto excede el alcance del proyecto de titulación, es importante conocerlo por si en un futuro se llegara a implementar, y saber las vías disponibles.

En cuanto a la retención, Fritz et al. (2017) proponen tres estrategias principales:

Reducción de la fricción y usabilidad. Fritz et al. (2017) indican que la principal característica que debe tener el proyecto de titulación es que sea simple, lo más simple y fácil de usar. Se debe asegurar que el sistema sea fácil de usar y que no genere fricciones, lo cual se profundizará más adelante.

Gamificación. Otro aspecto importante, según Fritz et al. (2017), es la gamificación. Se trata de convertir la recolección de datos (algo que puede resultar tedioso) en un proceso donde los usuarios tengan a la mano mecánicas de juego, como medallas, tablas de clasificaciones, entre otras. Esto funciona principalmente en audiencias jóvenes y es muy aplicable al sistema, ya que puede combinarse con las ideas abordadas sobre reputación y reconocimiento. Esto motivará

a los usuarios a participar, quienes dejarán de ver esto como una tarea tediosa y lo verán como un logro o competencia que hará su contribución mucho más entretenida.

Visualización del impacto. Los usuarios se sienten mucho más motivados cuando ven que sus contribuciones aportan en algo. Como se comentó en el sesgo de retención, una gran parte de los usuarios se ven motivados a seguir generando su contenido si ven que el reporte que generaron se resolvió. Fritz et al. (2017) también comentan el caso de *OpenStreetMap*, donde los colaboradores pueden ver rápidamente sus aportes en el mapa, dando una retroalimentación visual muy importante. Budhathoki y Haythornthwaite (2012, como se citó en Fritz et al., 2017) resaltan que las iniciativas de VGI deben incluir buenas representaciones visuales.

En el proyecto de titulación, se podrá tener retroalimentación del impacto tanto en el mapa como en el impacto en otras personas. Cuando un usuario genere un reporte, automáticamente lo podrá ver reflejado en el mapa y disponible para todo el mundo. Además, será capaz de ver cuántas visualizaciones y cuántas confirmaciones ha tenido su reporte. La motivación en sistemas VGI depende, por tanto, de la combinación de incentivos intrínsecos (tales como el altruismo, el aprendizaje y la autoeficacia) con mecanismos extrínsecos (como la reputación, la comunidad y la gamificación), complementados por estrategias de reclutamiento, reducción de fricción y visualización del impacto.

2.7 Privacidad y Riesgos

Cuanta más información comparten los ciudadanos, más útil resulta el sistema, pero también mayores son los riesgos para su privacidad. Este dilema ético y técnico requiere un análisis profundo de las diversas dinámicas de vigilancia que surgen en plataformas colaborativas, las cuales abarcan desde la supervisión mutua entre ciudadanos hasta los riesgos colectivos de estigmatización territorial.

2.7.1 Vigilancia Lateral

La vigilancia lateral se refiere a individuos que monitorean la conducta de otros individuos (Andrejevic, 2005, como se citó en Trottier, 2017). Es una vigilancia entre pares. Trottier (2017) comenta que esta vigilancia nace del miedo y desconfianza hacia otros individuos: se busca ver lo que hacen los demás, detrás de las máscaras sociales, para protegerse de sus actos.

Asimismo, Trottier (2017) sostiene que esta vigilancia no solo se produce por temor, sino también por el cuidado mutuo de los individuos, quienes monitorean su entorno con el fin de prevenir riesgos. En consecuencia, las personas se han vuelto más precavidas al compartir datos en internet y sumamente meticulosas a la hora de exponer información. En el contexto del proyecto de titulación propuesto, esto puede exponer a los ciudadanos a cierta vigilancia lateral al registrar datos geolocalizados como la ubicación. No obstante, el sistema también canaliza de forma constructiva este tipo de vigilancia: se informa a los ciudadanos sobre incidentes en su entorno para su protección, promoviendo una red comunitaria donde los propios vecinos contribuyen activamente a la seguridad colectiva.

2.7.2 Vigilancia Participativa

Según Trottier (2017), la vigilancia participativa se refiere a un proceso en el cual los usuarios comparten información sobre sí mismos a cambio de recibir un beneficio o servicio. Esta dinámica constituye el canal principal por el cual las personas se sienten motivadas a aportar sus datos, y a su vez, sustenta la existencia de la vigilancia lateral. Todo este entramado de vigilancia no representa un fenómeno aislado; por el contrario, la autocrítica y la observación de los demás están profundamente ligadas a la cultura interactiva de los medios digitales contemporáneos.

2.7.3 Vigilantismo como Arma

Trottier (2017) también advierte que el vigilantismo digital se utiliza con frecuencia como un arma de control social, donde los ciudadanos exponen información confidencial de otros individuos a modo de protesta, sanción moral o venganza, afectando gravemente su reputación. En este contexto, el autor resalta que las plataformas que facilitan la coordinación o la extracción libre de datos geográficos y personales comparten una responsabilidad indirecta en estas prácticas de vigilantismo, al proveer las herramientas que posibilitan dicha exposición.

2.7.4 El Dilema entre Privacidad y Utilidad

Dicho todo esto, se puede entender la gravedad de exponer datos sensibles como la ubicación y todos los peligros que esto conlleva. Pero a su vez, se puede entender que un proyecto de titulación como el propuesto, sin información confiable y verificable, no tendría el mismo impacto. Yu (2016, como se citó en Dunkel et al., 2020) introduce el dilema fundamental: un exceso de privacidad hace que la información sea inútil. Por ello, para poder generar información creíble y confiable, y a su vez respetar la privacidad que ciertas personas necesitan, la plataforma contempla opciones de privacidad para las personas que quieran mantenerse anónimas al reportar incidentes, mientras que las demás podrán seguir siendo visibles y generando la reputación, reconocimiento y diferentes dinámicas descritas anteriormente, tanto para generar motivación como para dar credibilidad.

2.7.5 Estigmatización Social y el Mapa como Productor de Miedo

Los riesgos anteriores se centran en el individuo, pero la exposición de datos geográficos también tiene consecuencias colectivas: la visualización de la información puede moldear la percepción que los ciudadanos tienen de ciertos lugares. En este sentido, la función de reportar incidentes del proyecto de titulación propuesto puede influir notablemente en cómo se perciben

los diferentes entornos urbanos. Por ejemplo, si una localidad presenta una alta concentración de reportes de delitos, los usuarios de la plataforma tienden a estigmatizar esos sectores; por el contrario, la presencia de reportes sobre eventos comunitarios positivos puede mejorar la percepción de seguridad. Al respecto, Riddell (2023) expone el caso de *Citizen*, una aplicación móvil para el reporte de crímenes que generó un aumento en la percepción de peligro constante entre los vecinos, elevando la paranoia colectiva. Esta estigmatización social no nace únicamente de las plataformas tecnológicas, sino que tiene orígenes socioculturales e históricos que pueden afianzar los sesgos ya mencionados; de este modo, el sistema propuesto podría servir involuntariamente como una vía para perpetuar y visibilizar estas estigmatizaciones preexistentes.

2.7.5.1 Estrategias de Mitigación

Con el fin de combatir estos riesgos de estigmatización y desconfianza, Trottier (2017) propone diversas estrategias de mitigación. Su recomendación primordial es de carácter educativo, señalando que no basta con regular la tecnología disponible si no se educa a los ciudadanos para comprender las consecuencias de estigmatizar o exponer a los demás, promoviendo al mismo tiempo el reconocimiento crítico de los patrones históricos de exclusión. Dado que esta labor educativa excede el alcance del presente proyecto de titulación, el diseño del sistema se centra en implementar aspectos tecnológicos y arquitectónicos controlables, los cuales se describen a continuación:

Sula (2016, como se citó en Mooney et al., 2017) señala que los riesgos se reducen cuando no se recopilan datos personales en primer lugar. Este principio se aplica en la plataforma al no almacenar datos personales de los usuarios además de los necesarios. El único dato

personal importante es la ubicación de los reportes, y por lo tanto es la única fuente de estigmatización presente: la estigmatización por localidad.

En cuanto a soluciones técnicas, Mooney et al. (2017) plantean que se pueden difuminar los datos. En este caso, se referiría a hacer que la ubicación de los reportes no sea totalmente exacta, lo cual iría en contra de todas las ventajas mencionadas del proyecto de titulación y, por lo tanto, no se aplicará.

Otro enfoque lo ofrece Riddell (2023), quien pone como ejemplo de otras plataformas el hecho de dar acceso a las comunidades marginadas o estigmatizadas al poder para corregir la información. Se menciona el ejemplo de una sección de comentarios, algo que se implementará en la plataforma para ayudar a los ciudadanos tanto a complementar información como a corregirla.

Por último, en el caso del proyecto de titulación propuesto, no se caería en la misma categoría de estos ejemplos, ya que el proyecto de titulación no se centra únicamente en incidentes de connotación negativa, sino que también se pueden reportar incidentes positivos o eventos comunitarios, lo cual ayudaría a que el proyecto de titulación no contribuya solamente al miedo y la mala percepción de los lugares, alejando así la estigmatización negativa.

En conclusión, este eje ha analizado el marco conceptual que sustenta la democratización geográfica propuesta por el proyecto de titulación. Se partió desde la *neogeografía* y la transición del experto al ciudadano, definiendo la VGI como el mecanismo mediante el cual los usuarios comparten voluntariamente datos geográficos en tres formatos: mapas, imágenes y texto. El *crowdsourcing* explica la lógica de participación colectiva para un beneficio mutuo, mientras que el análisis de calidad, que comprende desde las medidas ISO hasta los indicadores abstractos de

credibilidad, reputación y confianza, establece cómo evaluar y fortalecer la información generada.

Los sesgos inherentes al sistema, desde la desigualdad de participación hasta los riesgos de desinformación, fueron identificados junto con sus posibles mitigaciones. La motivación, tanto intrínseca como extrínseca, fue sistematizada como el motor que impulsa la contribución ciudadana, complementada con estrategias concretas de atracción y retención.

Finalmente, el análisis de privacidad y estigmatización social reconoce las tensiones éticas del sistema: el dilema entre utilidad y privacidad de los datos, y el riesgo de que la visualización geográfica refuerce percepciones negativas sobre ciertos lugares. Todo ello configura un marco teórico que guía las decisiones de diseño del proyecto de titulación, equilibrando participación, calidad, motivación y responsabilidad social.

3 CAPÍTULO 3: Análisis y diseño del sistema

3.1 Metodología de Desarrollo

3.1.1 Justificación de la metodología seleccionada

El proyecto actual y la forma de trabajo que se requiere realizar tienen ciertas características únicas que pocas metodologías soportan. Este trabajo es individual, por lo que inmediatamente se descarta una gran parte de metodologías comunes. A su vez, se cuenta con unas fechas establecidas, y ciertos objetivos a cumplir. Con un tiempo corto y con el desarrollo de una sola persona, lo mejor es adoptar una metodología ágil, que sea capaz de sobrellevar todos los inconvenientes que puedan surgir en el desarrollo, y a su vez, concentrar el esfuerzo en el desarrollo para cumplir con el tiempo y objetivos establecidos, evitando retrasos y posibles errores que cambien o pongan en riesgo al actual proyecto. Con todo esto, queda un conjunto de metodologías, en su mayoría metodologías ágiles populares, con adaptaciones para el desarrollo de una persona.

En esta situación se pensó en alternativas como una adaptación de XP (Extreme Programming), pero se descartó al sacrificar muchas de las características que se piden en un trabajo de titulación donde es importante justificar y documentar, y sobre todo planificar el proyecto desde un inicio.

Para cumplir con todos los requisitos, se optó por Agile Unified Process (AUP), el cual adopta las técnicas ágiles, pero a su vez requiere cierta formalidad y planificación inicial, la cual se realizó y se presentó en el capítulo de introducción. AUP fue creado teniendo como objetivo principal "el desarrollo de sistemas en tiempo real y aplicaciones basadas en web" (Flora & Chande, 2014), lo que se alinea directamente con la naturaleza del proyecto.

AUP no solo descarta las alternativas: se ajusta al proyecto por las cinco filosofías que promueve (Flora & Chande, 2014). Su simplicidad permite documentar el proceso en pocas páginas sin restar tiempo al MVP propuesto; su agilidad permite solucionar problemas y adaptarse a cambios de requerimientos, muy útil para el tiempo disponible; su enfoque en actividades de alto valor prioriza el desarrollo en lo que sí aporta valor y deja al final lo menos importante, igualmente importante para poder elegir las funcionalidades más importantes del proyecto; y su independencia de herramientas es coherente con el *stack* que se definirá más adelante.

3.1.2 ¿Qué es AUP?

AUP es una metodología de desarrollo de software, creada por Scott Ambler, que surgió a partir de Rational Unified Process (RUP). Esta metodología combina algunos flujos de trabajo de RUP con prácticas ágiles (Stober & Hansmann, 2010). Como lo describen sus propios formuladores, AUP "describe un enfoque simple y fácil de entender para desarrollar software de aplicaciones de negocio usando técnicas y conceptos ágiles, manteniéndose fiel a RUP" (Flora & Chande, 2014).

AUP integra diferentes técnicas ágiles como el Desarrollo Guiado por Pruebas (TDD), el Método Ágil, la gestión ágil del cambio y la refactorización de base de datos (Flora & Chande, 2014). E incorpora el concepto de Agile Model Driven Development (AMDD), que aplica el enfoque iterativo del TDD a los modelos del sistema (Stober & Hansmann, 2010).

AUP simplifica los nueve flujos de trabajo de RUP a **siete disciplinas**: Modelado, Implementación, Prueba, Despliegue, Gestión de la Configuración, Gestión del Proyecto y Entorno (Flora & Chande, 2014). También conserva las **cuatro fases** tradicionales de RUP: Inicio (*Inception*), Elaboración (*Elaboration*), Construcción (*Construction*) y Transición

(*Transition*). Estas se exploran con mayor detalle, junto con su aplicación en el proyecto, en la siguiente sección.

3.1.3 Aplicación de AUP en el proyecto

Fase de Inicio (*Inception*). Esta fase corresponde, en esencia, a lo desarrollado en la introducción del trabajo. Definición del alcance inicial, estimación del tiempo, identificación de riesgos y se termina con el hito LCO, el cual es el acuerdo al que se llega con todo lo definido (en este caso, la aprobación del plan de titulación).

Fase de Elaboración (*Elaboration*). Aquí se debe validar la viabilidad del proyecto mediante un análisis y diseño. Aquí es donde se crean los requerimientos, historias de usuario, casos de uso, arquitectura del sistema, el diseño de la base de datos, entre otros. Estos documentos pueden cambiar con el tiempo, pero ayudan en un principio a tener una vista general del proyecto a desarrollar.

Ciclo 0 — Preparación del entorno. Como paso previo a la Construcción, y antes de escribir código, se establece el *Development Case* (manual de proceso del proyecto) que define las iteraciones, las herramientas de modelado y los estándares de codificación y control de versiones.

Fase de Construcción (*Construction*). Concentra la mayor parte del trabajo. Aquí es donde se desarrolla el sistema, dividido en varias iteraciones. Cada iteración produce un incremento ejecutable. Pero antes de comenzar las iteraciones se deben preparar dos disciplinas, el entorno y la gestión de configuración y estándares. Cada iteración sigue un ciclo de inicio (escoger las historias de usuario a realizar), desarrollo (TDD e integración continua) y cierre (evaluar y retrospectiva), terminando en un producto funcional y probado. Adicionalmente, sobre cada incremento ya construido se aplicaron pruebas posteriores al desarrollo para validar su

correcto funcionamiento; si bien no forman parte estricta del flujo de AUP, sí fueron importantes para probar detalles relevantes.

Fase de Transición (*Transition*). Esta es la última fase donde se cierra el proyecto. Aquí se realizan pruebas de aceptación (UAT), corrección de defectos, documentación final y el lanzamiento a producción, entorno de ejecución local en nuestro caso (Flora & Chande, 2014).

Las secciones siguientes de este capítulo (análisis de requerimientos, actores, historias de usuario, casos de uso, arquitectura, diseño de la base de datos, diagramas de secuencia y diseño de la interfaz) constituyen los artefactos correspondientes a la fase de elaboración de la metodología AUP.

3.2 Análisis de Requerimientos

Se analizan los requerimientos del proyecto, tanto los requisitos funcionales (RF), donde se explica lo que hace la app, como los requisitos no funcionales (RNF), donde se explica qué tan bien lo hace.

En total existen 41 requisitos funcionales y 17 requisitos no funcionales, los cuales se pueden ver en anexos, pero a continuación se mostrará un resumen de todo lo hecho.

3.2.1 Requisitos Funcionales

Se presentan los requisitos divididos en las diferentes iteraciones que luego se explican a detalle, pero esto da una visión de qué es lo que se va a hacer en cada uno con los requisitos funcionales explicados.

Primero se comienza con la iteración de la fundación, aquí es donde se hace el *setup* de todo el proyecto, con el *frontend, backend* y base de datos funcionando en su forma más básica. En esta iteración no existen requerimientos funcionales.

La segunda iteración, identidad y acceso. Aquí existen 8 RFs, y están enfocados en la autenticación que tiene la aplicación, desde el registro de una cuenta, la autenticación federada con Google, diferentes requisitos de seguridad, la gestión de sesión, recuperación de contraseña, entre otros que aseguran que el sistema es seguro, y de fácil acceso por los ciudadanos.

La tercera iteración es Mapa y Ubicación. Aquí están todos los requerimientos que necesita el proyecto al mostrar y utilizar el mapa. Aquí se tienen 7 RFs, entre los cuales está el mapa de teselas vectoriales propuestos, ubicación por GPS, geocodificación inversa, visualización por clústeres, validación espacial, entre otros.

La cuarta iteración es Creación de Reportes. Aquí existen 6 RFs, los cuales nos dan las pautas de lo que necesita el sistema para que la creación de reportes sea rápida, efectiva, y con facilidad de creación para la mayoría de los usuarios. Entre los requisitos está un formulario para escoger la categoría del reporte, el poder subir una foto, un formulario para el título y descripción, y otros requisitos técnicos como el almacenamiento de los reportes.

La quinta iteración, Visualización. Aquí es donde se pule toda la visualización de la información faltante. Aquí se cubren los requisitos de los filtros para visualizar los reportes, compartir los reportes por enlaces, un *dashboard* de mis reportes, con historial y borrado.

La sexta iteración, PWA/Offline:

Aquí se convierte la página normal a una PWA, y a su vez, con la ayuda de esta tecnología se puede hacer página offline-first. Para ello hay requerimientos empezando por la instalabilidad de la aplicación, la aplicación de tecnologías para hacer offline y optimización de recursos y peticiones.

La séptima iteración, Geocolaboración. Aquí se centra en requerimientos hablados en la fundamentación teórica como la validación entre pares, el módulo de comentarios, el linaje del dato y moderación de contenido.

Por último, en la octava iteración, Gamificación y Confianza. Aquí es donde se aplican los requerimientos que ayudan a que los usuarios estén más motivados a usar la aplicación. Entre ellos está un motor de logros e insignias, un catálogo de vitrinas de logros, estadísticas del impacto del reporte y un perfil de confianza sobre métricas de participación y reputación.

Tabla 1
Resumen de Iteraciones

Área	N.º de funciones	Ejemplos
Identidad y acceso	8	Registro, ingreso con Google, roles invitado/ciudadano
Mapa y ubicación	7	Mapa de Quito, GPS en tiempo real, geofencing del DMQ
Creación de reportes	6	Formulario por categoría, foto opcional, ubicación validada
Visualización	5	Carga por zona visible, filtros, "Mis reportes"
Funcionamiento sin internet	6	Instalación PWA, consulta offline, envío al reconectar
Colaboración entre vecinos	5	Confirmar/cuestionar, comentar, aviso de duplicados
Confianza y reconocimientos	4	Perfil de confianza, insignias, visualizaciones
Total	**41**	—

3.2.2 Requerimientos No Funcionales

Por otra parte existen diferentes requerimientos no funcionales, los cuales se aplicaron a lo largo del proyecto y que eran necesarios para que el proyecto funcionara tal y como se espera. En esta tabla se hace una agrupación para mejor entendimiento sobre un resumen de todos los RNFs aplicados.

Tabla 2
Resumen RNF

Categoría	Qué garantiza	Ejemplo
Seguridad y protección de datos	Datos protegidos y cumplimiento legal	Cifrado, contraseñas protegidas, LOPDP
Rapidez	Apertura y respuesta veloces en datos móviles	Carga útil en pocos segundos sobre 4G
Facilidad de uso	Uso simple, accesible y móvil	Funciones principales en el menor número de toques posible, diseño desde 320 px
Sin internet y fiabilidad	Sin caídas al perder la red; datos íntegros	Modo offline ante errores
Mantenibilidad y portabilidad	Código ordenado y compatible	Arquitectura en capas; Chrome y Safari

3.3 Actores del Sistema

Algo importante a saber, es quiénes van a usar el sistema. En este caso, solamente existen dos actores: Invitados y Usuarios.

Los invitados son las personas que no se han registrado, pero quieren ver los reportes existentes. Se prefirió crear este actor para que las personas no tengan ninguna fricción al ver reportes creados por los demás usuarios y así tengan más curiosidad al usar la aplicación y tengan incentivos en crear una cuenta.

Por otro lado, están los usuarios registrados, los cuales tienen acceso a todo. Visualizar los reportes, crear reportes y colaborar con los demás ciudadanos, además de gestionar su perfil.

A continuación, se presenta una tabla de resumen para poder visualizar fácilmente qué accesos tiene cada actor en el sistema

Tabla 3
Permisos de Usuarios

Acción	Invitado	Ciudadano
Ver el mapa de Quito	✅	✅
Filtrar por categoría	✅	✅
Ver el detalle de un reporte	✅	✅
Registrarse / iniciar sesión	✅	—
Crear un reporte	❌	✅
Subir foto	❌	✅
Reportar en modo anónimo	❌	✅
Confirmar o cuestionar	❌	✅
Comentar	❌	✅
Ver perfil e historial	❌	✅
Ganar y ver insignias	❌	✅

Ver visualizaciones de sus reportes	❌	✅

Aclaración:

Aunque el proyecto actual no presenta un administrador, todo el sistema está listo para que este se implemente. Pero se decidió deliberadamente aislarlo del MVP ya que la calidad de los datos se cuida entre propios usuarios y existen filtros de lenguajes y reportes duplicados. Aun así, es importante señalar que puede llegar a ser importante en ciertas situaciones pero queda fuera del alcance actual del proyecto.

3.4 Historias de Usuario

Las historias de usuario traducen los requerimientos a la voz del ciudadano: en lugar de describir lo que el sistema debe hacer en términos técnicos, expresan lo que una persona real necesita lograr y por qué. Sirven de puente entre el análisis y el desarrollo, y permiten verificar cada función desde la perspectiva de quien la usará.

Se redactan con el formato "Como [tipo de usuario], quiero [acción], para [beneficio]" y cada una incorpora criterios de aceptación que definen cuándo se considera terminada. En total se definieron 29 historias distribuidas en las 7 áreas del sistema, enlazadas con sus respectivos requerimientos funcionales (RF).

A continuación, se muestra una tabla resumen de todas las historias de usuario existentes y sus asociaciones con los RF

Tabla 4
Resumen RF

Área	ID	Título	RF
Identidad	HU-01	Registro tradicional	RF-1.1
Identidad	HU-02	Ingreso con Google	RF-1.2

Identidad	H U-03	Navegación como invitado	RF-1.6
Identidad	H U-04	Cierre de sesión	RF-1.4
Identidad	H U-26	Recuperar contraseña	RF-1.8
Identidad	H U-27	Eliminar cuenta	RF-1.5
Mapa	H U-05	Mapa del DMQ	RF-2.1
Mapa	H U-06	GPS en tiempo real	RF-2.2
Mapa	H U-07	Ubicación manual (pin)	RF-2.4, RF-2.5
Reportes	H U-08	Formulario de reporte	RF-3.1, RF-3.2
Reportes	H U-09	Foto del reporte	RF-3.3
Reportes	H U-10	Validación de geofencing	RF-2.3, RF-3.4
Visualización	H U-11	Carga por zona visible	RF-4.1
Visualización	H U-12	Ficha de detalle	RF-4.2
Visualización	H U-13	Filtros	RF-4.3
Visualización	H U-14	Mi historial	RF-4.6
Visualización	H U-28	Compartir reporte	RF-4.5
Visualización	H U-29	Eliminar mis reportes	RF-4.6
Sin internet	H U-15	Instalar la PWA	RF-5.1
Sin internet	H U-16	Guardar reporte offline	RF-5.4
Sin internet	H U-17	Envío automático al reconectar	RF-5.4
Sin internet	H U-25	Reporte anónimo	RF-5.6
Colaboración	H U-18	Aviso de duplicado cercano	RF-6.1
Colaboración	H U-19	Confirmación ciega	RF-6.2

Colaboración	HU-20	Comentarios	RF-6.3, RF-6.5
Colaboración	HU-21	Linaje del dato	RF-6.4
Reconocimientos	HU-22	Perfil de confianza	RF-7.1
Reconocimientos	HU-23	Insignias	RF-7.2, RF-7.3
Reconocimientos	HU-24	Impacto de mis reportes	RF-7.4

3.4.1.1 Historias representativas

Ahora se profundiza sobre las historias de usuario más representativas como ejemplo. Todas las 29 historias de usuario completas estarán en los anexos. En estos ejemplos se utilizó Given-*When*-*Then* (Dado, cuando, entonces), pero en las demás historias de usuario, por rapidez solamente se usan viñetas con los criterios de aceptación

HU-09 — Reportar con foto. "Como ciudadano testigo de un reporte, quiero adjuntar una fotografía, para aportar evidencia visual."

- **Dado** que estoy creando un reporte, **cuando** pulso "Añadir foto", **entonces** se abre la cámara del dispositivo.
- **Dado** que capturé una foto, **cuando** la confirmó, **entonces** se comprime a ≤500 KB antes de enviarse**.**
- **Dado** que no deseo adjuntar foto, **cuando** envío el reporte, **entonces** el sistema lo acepta igualmente, pues la foto es opcional para no frenar un reporte urgente.

HU-11 — Carga por zona visible. "Como usuario con un plan de datos limitado, quiero que solo se descarguen los reportes de la zona que estoy viendo, para reducir mi consumo de datos móviles."

- **Dado** que tengo el mapa abierto, **cuando** desplazo o hago zoom, **entonces** solo se solicitan al servidor los reportes contenidos en el área visible.

HU-16 / HU-17 — Reporte offline. "Como usuario sin conexión, quiero que mi reporte se guarde localmente y se envíe automáticamente al recuperar la señal, para no perder la información reportada en zonas sin cobertura."

- **Dado** que no hay red, **cuando** envío un reporte, **entonces** se almacena en el dispositivo con el aviso "Pendiente de conexión".
- **Dado** que existen reportes pendientes, **cuando** se restablece la conexión, **entonces** se envían automáticamente sin intervención del usuario y se notifica el éxito.

HU-18 — Aviso de duplicado. "Como ciudadano, quiero que el sistema me avise si ya existe un reporte parecido cerca, para confirmarlo en lugar de duplicarlo."

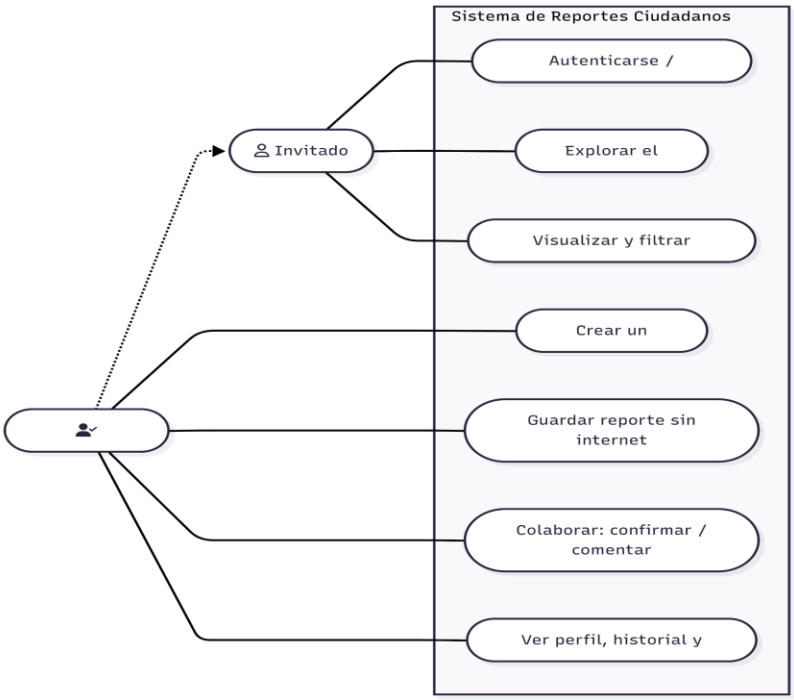
- **Dado** que estoy fijando la ubicación de un nuevo reporte, **cuando** hay reportes similares cercanos, **entonces** se muestra un aviso que no bloquea el flujo y puede descartarse con un solo toque.

3.5 Casos de Uso

En esta sección se van a detallar los casos de uso hechos. Estos casos de uso describen las interacciones entre los actores y el sistema. A diferencia de las anteriores historias de usuario, las cuales describen el qué y por qué, aquí se detalla el cómo se llevan a cabo estas acciones.

A continuación se mostrará el diagrama general, 3 ejemplos de casos detallados y un catálogo de resumen de los 20 casos. Los casos de uso completos estarán en anexos.

Figura 1
Diagrama de Casos de Uso General



CU-07 — Crear un reporte

- **Actor:** Ciudadano.
- **Precondiciones:** sesión iniciada; con conexión.
- **Resultado:** reporte activo y visible en el mapa.

Flujo principal:

1. El ciudadano fija el punto en el mapa.
2. La app valida que la ubicación esté dentro de Quito.
3. El ciudadano elige categoría, título, descripción y, opcionalmente, una foto.
4. El ciudadano envía el reporte.
5. El servidor revalida la ubicación, guarda la foto y el reporte.

6. La app muestra un mensaje de éxito.

Flujos alternativos:

- **Fuera de Quito:** se bloquea el envío.
- **Sin conexión:** el flujo continúa en el CU-08.
- **Usuario invitado:** se le pide iniciar sesión.

RF: RF-3.1 a RF-3.4, RF-2.3 · HU: HU-08 a HU-10.

CU-14 — Confirmar o cuestionar un reporte

- **Actor:** Ciudadano (distinto del creador).
- **Precondiciones:** sesión iniciada; reporte abierto; no haber votado antes.
- **Resultado:** voto registrado; se revelan los conteos; se actualiza la reputación.

Flujo principal:

1. El ciudadano abre la ficha del reporte (con los conteos ocultos).
2. Vota "Confirmar" o "Cuestionar".
3. El servidor valida que no sea el autor ni un voto repetido y lo guarda.
4. La app revela los conteos.

Flujos alternativos:

- **Ya votó o es el autor:** no se permite el voto.
- **Disputa:** no borra el reporte, solo lo marca para revisión.

RF: RF-6.2 · HU: HU-19.

CU-08 — Guardar un reporte sin internet

- **Actor:** Ciudadano.
- **Precondiciones:** sesión iniciada; app instalada; sin conexión.
- **Resultado:** el reporte se sincroniza solo al reconectar.

Flujo principal:

1. El ciudadano crea el reporte.
2. La app detecta que no hay red.
3. Guarda el reporte en el dispositivo con el aviso "Pendiente de conexión".
4. Al volver la señal, el reporte se envía automáticamente y la app avisa del éxito.

RF: RF-5.4 · HU: HU-16, HU-17.

3.6 Arquitectura del Sistema

Ahora se detallará la arquitectura del sistema. Para ello se optó por una arquitectura cliente-servidor de tres capas:

- **Aplicación en el dispositivo (PWA):** Esta es la aplicación que ve y usa el ciudadano, aquí se descargará el mapa de Quito y guarda reportes pendientes. Funciona sin internet.
- **Servidor:** Se encarga de la lógica del sistema, recibe los datos, valida y los procesa. Entre sus funciones están revalidar la ubicación y aplicar las reglas de moderación, insignias y duplicados, entre otras.

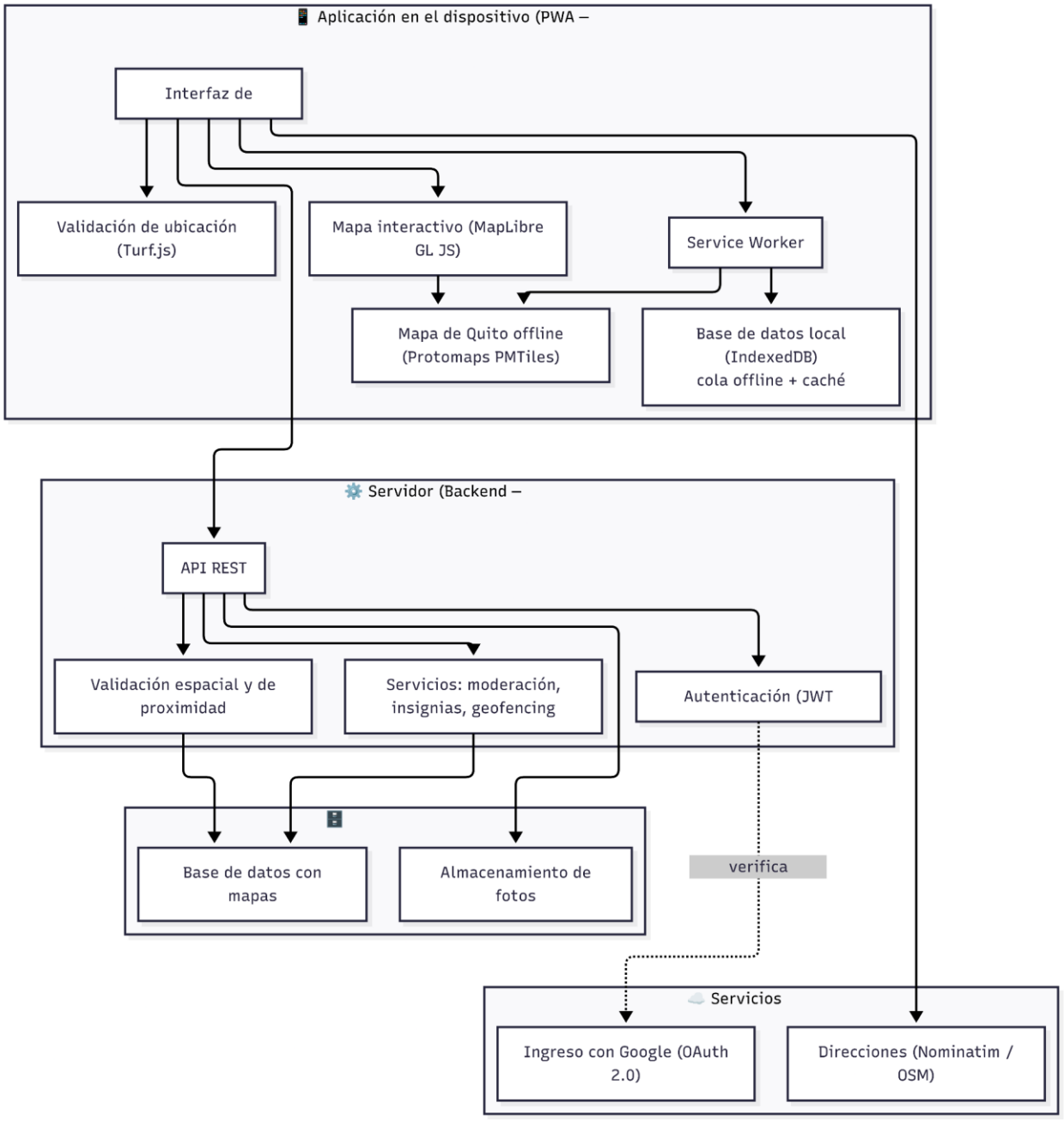
- **Base de datos:** Aquí se almacenan todos los datos. Además, puede hacer consultas rápidas y adaptadas a las necesidades geográficas de la aplicación gracias a PostGIS. Las fotos van a un almacenamiento aparte (Supabase)

  También se cuentan con otros servicios externos:

ingreso con Google, almacenamiento de fotos y obtención de direcciones a partir de coordenadas

Se optó por esta arquitectura porque separa las responsabilidades (presentación, lógica y datos), lo que hace que sea más fácil de mantener y permite que la aplicación siga funcionando sin conexión de forma independiente

Figura 2
Diagrama de Arquitectura



En el diagrama, se muestran las tres capas en cada recuadro, con sus componentes y las flechas sobre cómo fluye la información. Además, también se muestran los servicios externos en otro recuadro ya que no forman parte del sistema pero son importantes en el funcionamiento

A continuación, como ejemplo, se detalla cómo sería el recorrido de un reporte en esta arquitectura

1. Se crea en el celular del ciudadano.
2. La app valida que la ubicación esté dentro de Quito.
3. El reporte viaja al servidor.
4. El servidor revalida la ubicación y guarda la foto y se guardan los datos en la BD.
5. El reporte queda disponible para que otros usuarios lo vean.

La ubicación se valida dos veces (en el celular y en el servidor): la primera avisa al instante, mientras que la segunda garantiza que ningún reporte fuera de Quito entre al sistema por seguridad.

3.6.1 Aplicaciones Web Progresivas (PWA): justificación de la elección tecnológica

La capa cliente descrita en la arquitectura se implementa como una Aplicación Web Progresiva (PWA); a continuación, se fundamenta esta elección tecnológica. En 2015, la diseñadora Frances Berriman y el ingeniero de *Google Chrome* Alex Russell acuñaron el término "aplicaciones web progresivas" para describir las aplicaciones que aprovechan las nuevas funciones compatibles con los navegadores modernos, incluidos los *service workers* y los manifiestos de aplicaciones web, que permiten a los usuarios actualizar las aplicaciones web a aplicaciones web progresivas en su sistema operativo nativo (Tandel & Jamadar, 2018).

Las aplicaciones web progresivas (PWA) permiten desarrollar aplicaciones web con un enfoque de funcionamiento sin conexión. Si bien tradicionalmente las aplicaciones web no funcionaban correctamente sin conexión a Internet, las PWA permiten descargarlas, instalarlas y utilizarlas sin conexión en diversos sistemas, incluidos dispositivos móviles y ordenadores

personales (Biørn-Hansen et al., 2018). En pocas palabras, una PWA es un sitio web creado con tecnologías web que funciona como una aplicación y no requiere instalación como una aplicación nativa (Tandel & Jamadar, 2018).

Las PWA cuentan con varias características útiles para el proyecto de titulación:

Progresiva. Esto garantiza que funcionarán para cualquier usuario sin importar su ubicación geográfica, el navegador o la capacidad de su dispositivo (Tandel & Jamadar, 2018).

Responsiva. Las interfaces se adaptan a cualquier dispositivo con diferentes tamaños de pantalla como laptops, monitores, celulares (Tandel & Jamadar, 2018).

Sensación de aplicación (*app-like*). Aunque no sea una aplicación nativa de celular, sí emula su comportamiento y diseño (Biørn-Hansen et al., 2018).

Instalable. El usuario puede agregar esta aplicación a su pantalla de inicio y tenerla guardada en su celular, sin pasar por tiendas de aplicaciones como *Play Store* o *App Store* (Tandel & Jamadar, 2018).

Actualización en segundo plano. La PWA se actualiza automáticamente, sin que el usuario tenga que hacerlo manualmente, facilitando agregar nuevas versiones o parches (Tandel & Jamadar, 2018).

Notificaciones. Mediante la API Web Push, la PWA puede enviar notificaciones al usuario en su teléfono, igual a lo que hace una aplicación nativa (Biørn-Hansen et al., 2018).

Segura. La PWA, por defecto, usa conexión HTTPS, garantizando la privacidad e integridad de la información transmitida en el sistema (Tandel & Jamadar, 2018).

Compartible (*linkable*). Dado que es una página web en el núcleo, se pueden compartir directamente a través de un enlace URL. Esto facilita que las personas compartan sus reportes simplemente mandando enlaces (Biørn-Hansen et al., 2018).

Independiente de conectividad. Usando la tecnología de *service workers*, el proyecto de titulación puede funcionar en situaciones sin internet o donde la conexión es limitada (Tandel & Jamadar, 2018). Esta característica se profundiza en la siguiente sección.

Como ya se vio, los usuarios pueden ser muy variados, con diferentes habilidades, diferentes capacidades, y cada uno con sus propias limitaciones de *hardware* y conectividad. Para que el proyecto de titulación sea usable y no genere ningún sesgo en sus datos, se debe aplicar una tecnología que incluya a la gran mayoría de usuarios. Con todas las facilidades que brinda PWA, ayuda en varios frentes. Primero ayuda a que las personas puedan ver ciertos reportes sin instalar el proyecto de titulación, facilitando a personas con poco almacenamiento y quitando la fricción que se tiene al instalar una aplicación común. Segundo, esta tecnología mejora a las páginas convencionales, haciéndolas instalables y usables de manera *offline*, ofreciendo funciones de aplicaciones nativas, pero con las características de las páginas web.

3.6.2 Arquitectura offline-first

Como ya se explicó, *offline-first* es una característica de las PWA, y también es una característica importante en el proyecto de titulación. Aplicando esta metodología, se puede hacer que los usuarios con nula o baja conectividad puedan seguir usando el proyecto de titulación.

Este paradigma permite a los desarrolladores priorizar el funcionamiento sin conexión mediante el uso de motores de almacenamiento integrados en el navegador web (como *IndexedDB*, el que se va a usar en el sistema). Esto resulta especialmente relevante para mercados emergentes donde la conectividad a internet es escasa y costosa (Biørn-Hansen et al., 2018).

Un concepto importante en la arquitectura *offline-first* es el *Store-and-Forward* (almacenar y reenviar). Este método captura y almacena los datos en tiempo real en los dispositivos móviles de forma local y retrasa la sincronización (subida de esa información) hasta el momento que se detecta que existe conexión (McIntosh et al., 2016). Esto ayuda a usuarios que no cuentan con internet y quieren reportar. Cuando el usuario hace el reporte, se guarda en el almacenamiento del dispositivo hasta que se detecte que existe internet, y en ese momento se sube a la base de datos.

La independencia de conectividad es uno de los conceptos centrales que diferencian a una PWA de un sitio web convencional. La tecnología PWA facilita esta arquitectura al implementar *service workers*; esto se logra mediante la combinación de scripts de intercepción de red (*service workers*) y cachés locales de datos y recursos estáticos (Biørn-Hansen et al., 2018).

Por último, en países con ingresos medios o bajos donde las conexiones a internet fallan o son deficientes, esta arquitectura destaca, mejorando la movilidad y el alcance, flexibilizando los métodos de recolección de datos sobre el terreno (McIntosh et al., 2016). Esta realidad es directamente aplicable al DMQ, donde existen zonas con conectividad intermitente que, sin esta arquitectura, quedarían excluidas de la plataforma.

También se explica en la siguiente tabla las tecnologías que usamos y detallamos en el diagrama de arquitectura.

Tabla 5
Tecnologias Usadas

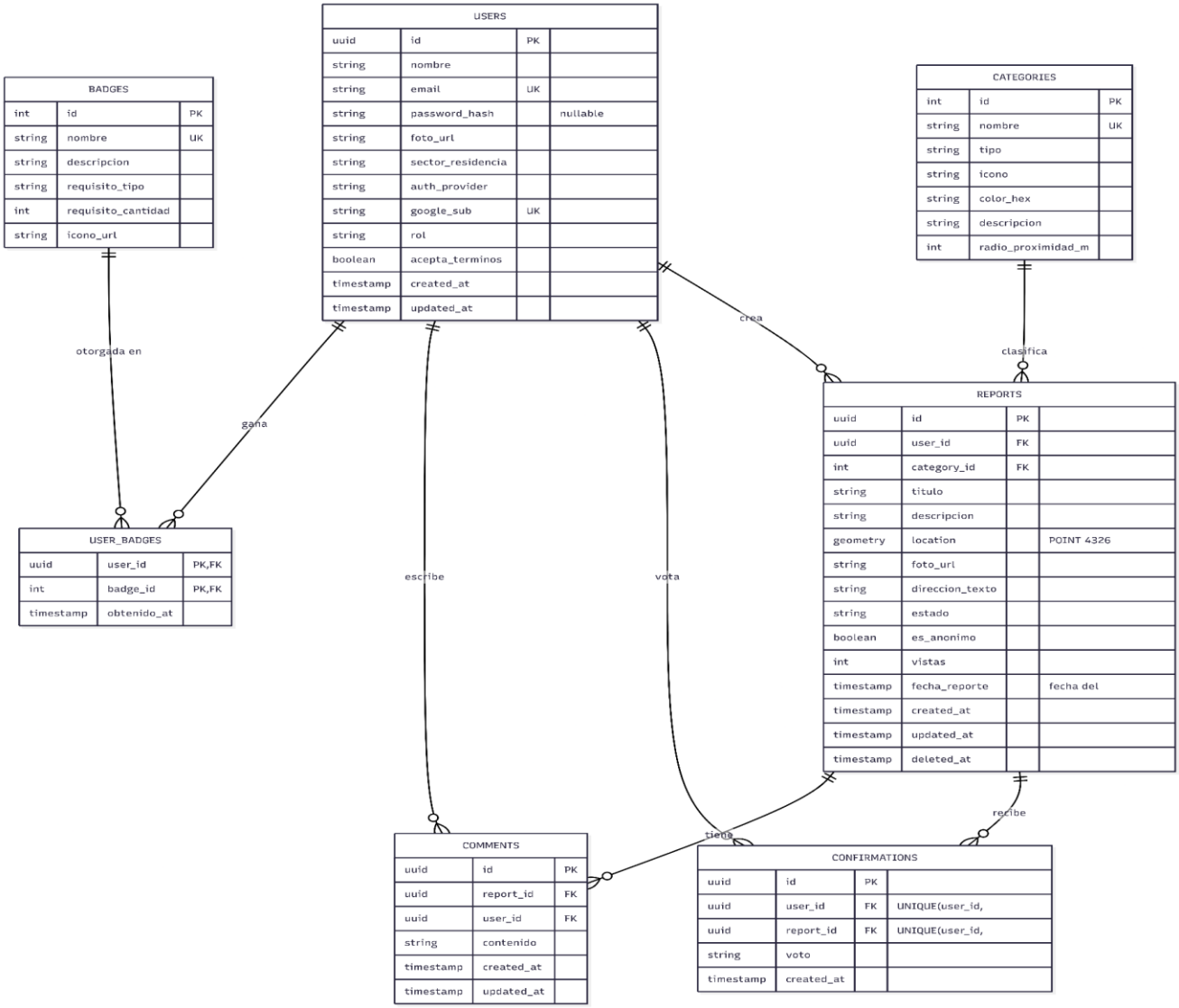
Tecnología	Para qué se usó
App web instalable (PWA, React)	La app que usa el ciudadano; offline

Mapa interactivo (MapLibre + Protomaps)	Mapa de Quito fluido y sin conexión
Validación en el dispositivo (Turf.js)	Comprobar al instante que el punto esté en Quito
Servidor (FastAPI, Python)	Recibir, validar y procesar; aplicar reglas
Base de datos con mapas (PostgreSQL + PostGIS)	Almacenar y responder preguntas geográficas
Almacenamiento (Supabase Storage)	Guardar las fotos
Acceso (Google OAuth 2.0)	Iniciar sesión con Google
Direcciones (Nominatim / OSM)	Traducir coordenadas a una dirección

3.7 Diseño de la Base de Datos

La información del sistema se diseñó siguiendo un modelo relacional que se presenta en el siguiente diagrama. Se guarda en 7 entidades: usuarios, categorías, reportes, confirmaciones, comentarios, insignias e insignias obtenidas por cada usuario.

Figura 3
Modelo de Base de Datos



Relaciones:

- Un usuario crea muchos reportes
- Una categoría clasifica muchos reportes
- Un reporte recibe muchas confirmaciones y comentarios
- Un usuario vota una sola vez cada reporte

- Usuarios e insignias es muchos-a-muchos (vía user_badges).

Decisiones de datos clave:

- **Reporte anónimo:** En la aplicación se muestra como "Ciudadano Anónimo" para conservar la privacidad frente a otros usuarios, pero la BD conserva internamente el vínculo real (respaldo).
- **Borrado lógico:** Los reportes no se borran de manera irreversible, en cambio, se marcan con fecha de eliminación, todo esto siguiendo el derecho al olvido de la LOPDP, y dejan de mostrarse.
- **BD con mapas (PostGIS):** Gracias a PostGIS, se pueden hacer diferentes consultas rápidas de datos geográficos, lo que ayuda a la eficiencia en consultas como, qué hay cerca de cierto punto, o si cierto punto está dentro del límite permitido de Quito

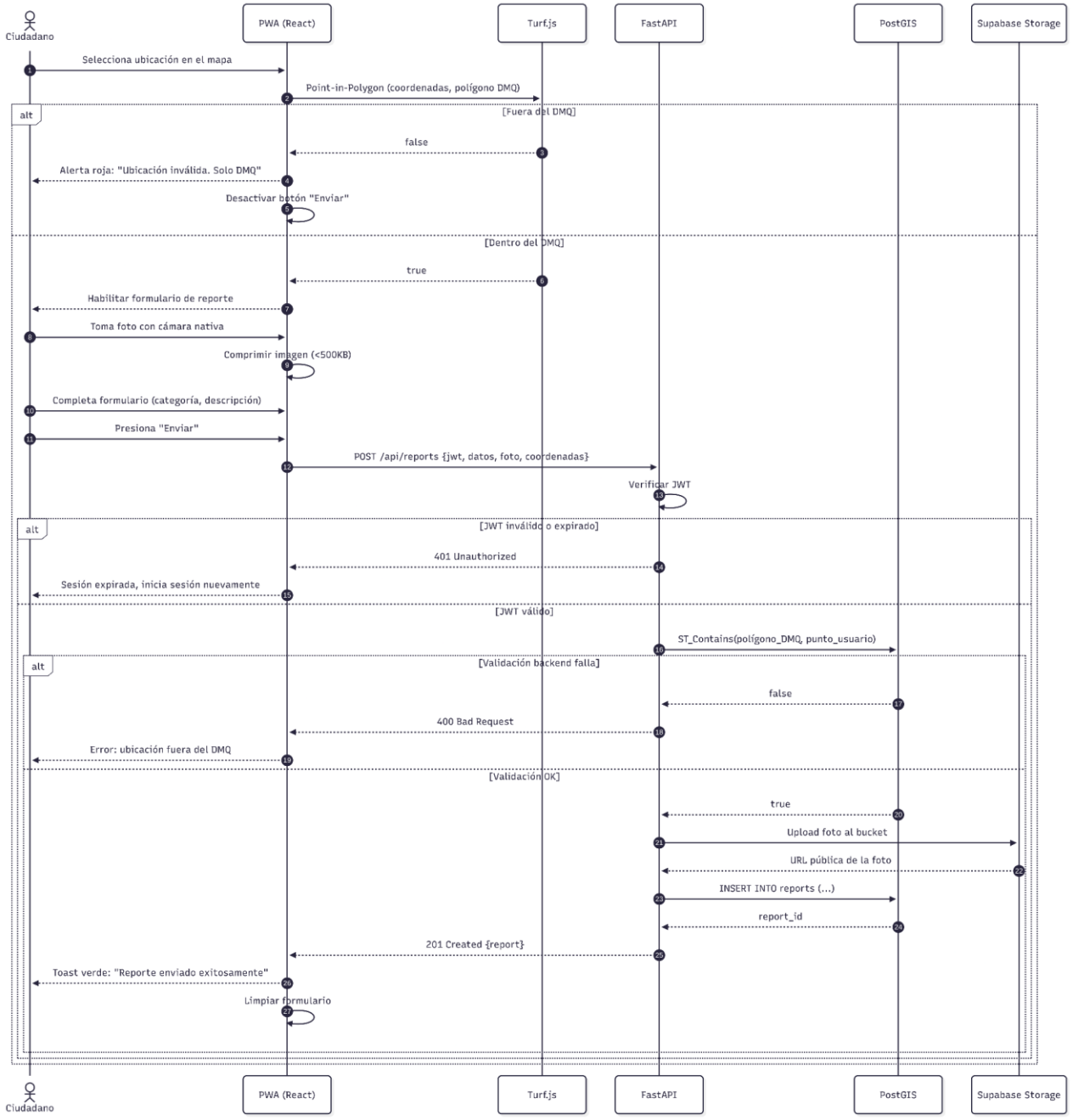
3.8 Diagramas de Secuencia

Los últimos diagramas para realizar fueron los diagramas de secuencia. Estos permiten plasmar todas las acciones detalladas anteriormente en la dimensión del tiempo. Con ello se observa a más detalle cómo se ejecutan los procesos, en qué orden, y quienes interactúan. A continuación, se muestran tres flujos: crear un reporte, guardar sin internet y confirmar un reporte.

3.8.1 Crear un reporte (con validación de ubicación)

Figura 4

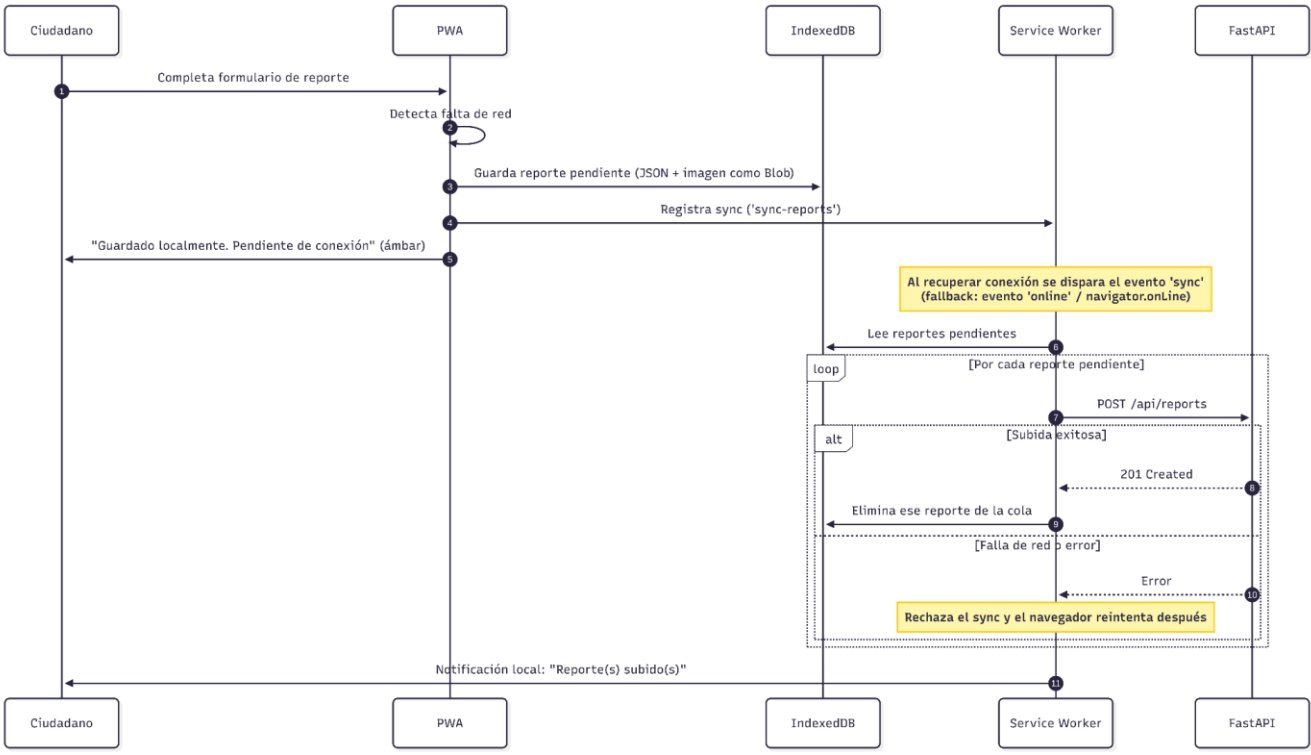
Diagrama de Secuencia (Crear Reporte)



3.8.2 Guardar sin internet y enviar al reconectar

Sin red, el reporte se guarda en el dispositivo; al volver la conexión, el Service Worker lo envía solo.

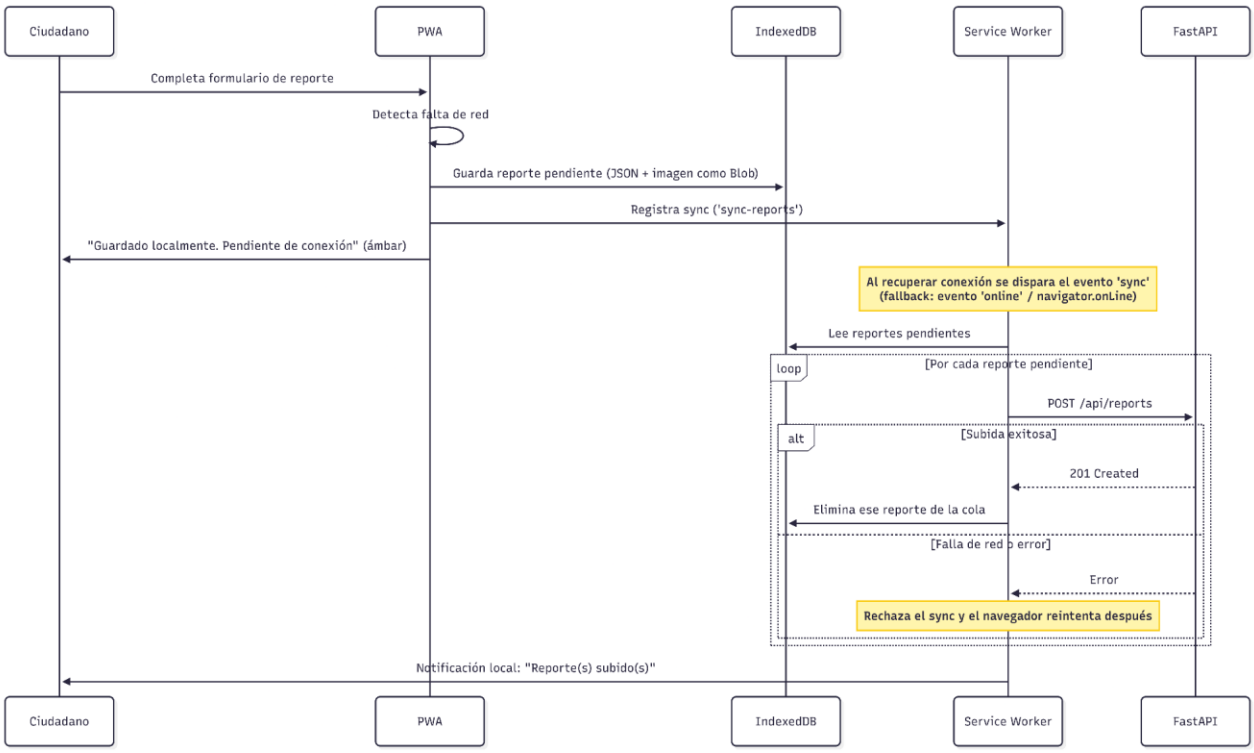
Figura 5
Diagrama de Secuencia (Offline)



3.8.3 Confirmar un reporte (validación ciega)

Los conteos se mantienen ocultos hasta que el ciudadano vota; el servidor evita que vote el autor o que vote dos veces.

Figura 6
Diagrama de Secuencia (Confirmar Reporte)



3.9 Diseño de la Interfaz de Usuario (UI/UX)

El diseño de la interfaz nace de ciertos conceptos como el Diseño Centrado en el Usuario (DCU) y de criterios de usabilidad y heurísticas específicas para mapas web. Por lo que primero los vamos a explorar a fondo, y en base a ello vamos a sacar diferentes ideas y conclusiones para nuestra aplicación

Diseño Centrado en el Usuario (DCU)

El diseño centrado en el usuario (DCU) describe un enfoque temprano y activo en las necesidades del usuario al conceptualizar e implementar una interfaz, con énfasis en el refinamiento iterativo de la facilidad de uso y la utilidad de la interfaz (Roth et al., 2015).

El proyecto de titulación emplea el DCU, el cual es clave para que la plataforma no sea solamente funcional, sino que responda a todas las necesidades de los usuarios que se han mencionado. Y aunque no se aplique el ciclo completo, sí se toma en cuenta a la hora de desarrollar el sistema.

Roth et al. (2015) dividen el DCU (explícitamente en cartografía) en lo que llaman las 3 U (usuario, utilidad y usabilidad). Para ello se pueden seguir los siguientes pasos que los autores proponen:

Usuarios. Primero se debe conocer a los usuarios y para ello se proponen axiomas, de los cuales se destacan tres para el presente proyecto:

Primero, los expertos del dominio no representan a los usuarios típicos (Roth et al., 2015). Estos son expertos en tecnología (programadores de *software*, administradores de bases de datos) o funcionarios del DMQ y administradores o representantes de comunidades. Ellos entienden perfectamente las tecnologías que se están usando o, por el contrario, pueden tener un gran conocimiento en los incidentes y eventos que suceden en su entorno. Por lo tanto, no se debe diseñar un sistema para ellos, con coordenadas exactas en la pantalla o funciones avanzadas, sino diseñar el proyecto de titulación para los usuarios típicos.

Segundo, los usuarios rara vez saben lo que quieren al principio, por lo que el equipo debe traducir sus peticiones abstractas en requisitos concretos (Roth et al., 2015). Muchos usuarios solamente quieren una aplicación para avisar rápido sobre algo que pasa a su alrededor. Sin embargo, para el proyecto se debe ir más allá y ver todos los problemas y obstáculos que se tienen, y cómo los usuarios pueden llegar a tener mejor experiencia. Entre esos obstáculos puede estar la baja conectividad, poco tiempo a la hora de avisar sobre un reporte o limitación,

limitaciones en el *hardware* del celular, entre otros obstáculos que el usuario no conoce, pero le afectan.

Respecto al último axioma, los usuarios exhiben una gran diversidad en características, habilidades y necesidades (Roth et al., 2015). Esto indica que los usuarios son diversos, algo fundamental en el proyecto, el cual tiene como objetivo abarcar a una gran cantidad de ciudadanos dentro del DMQ. Por lo cual, se debe pensar en todos ellos, y tratar de no dejar a ningún grupo ciudadano por fuera. Pensar en que todos tienen habilidades tecnológicas diferentes, necesidades diferentes y objetivos diferentes, por lo cual la plataforma debe ser simple para los menos expertos, fácil de usar para quienes no tienen mucho conocimiento tecnológico, ofrecer una amplia gama de categorías para la variedad de necesidades de cada uno y contemplar alternativas para las diferentes limitaciones de tecnología que se tienen en el DMQ.

Utilidad. Describe la pertinencia de la interfaz para que el usuario complete sus objetivos deseados (Roth et al., 2015). En este sentido, el proyecto de titulación debe resolver realmente los objetivos propuestos, los cuales puedan ser logrados mediante una correcta UI/UX. No se debe enfocar solamente en una interfaz llamativa, sino en una interfaz útil; cada elemento visual tendrá un objetivo y por lo tanto será funcional. La tercera U, correspondiente a la usabilidad, se desarrolla a continuación.

3.9.1 Usabilidad: medidas de Nielsen

Siguiendo con la misma línea de Roth et al. (2015), se describe a la usabilidad como la facilidad con la que se puede utilizar una interfaz para completar el conjunto de objetivos deseados por el usuario. La plataforma, además de útil, debe ser fácil de usar. Como se mencionó, se tienen diferentes tipos de usuarios, y el proyecto de titulación debe ser accesible para todos, eliminando la fricción mencionada anteriormente.

Nielsen (1993, como se citó en Roth et al., 2015) enumera cinco medidas de usabilidad, de las cuales solo se verán cuatro (se excluye la satisfacción subjetiva, ya que su evaluación requiere pruebas con usuarios finales fuera del alcance del presente proyecto), y aunque no se aplique a usuarios en pruebas, sí se pueden tomar en cuenta cuando se diseñe el proyecto de titulación.

1. **Facilidad de aprendizaje (*Learnability*):** Rapidez con la que los usuarios comprenden la interfaz sin haberla utilizado previamente (Nielsen, 1993, como se citó en Roth et al., 2015). Los usuarios deben ser capaces de crear reportes y ver reportes de manera fácil, sin fricción.

2. **Eficiencia (*Efficiency*):** Velocidad con la que los usuarios interactúan con la interfaz para completar las tareas deseadas una vez aprendida (Nielsen, 1993, como se citó en Roth et al., 2015). Este aspecto es muy importante en el proyecto de titulación, donde la rapidez al reportar es una prioridad. Los usuarios deben poder interactuar con la plataforma con pocos clics y lograr sus objetivos de la manera más rápida posible.

3. **Recordabilidad (*Memorability*):** Capacidad de los usuarios para regresar a la interfaz y retomar sus actividades con facilidad (Nielsen, 1993, como se citó en Roth et al., 2015). Se emplean elementos de *affordance* para hacer que a los usuarios se les haga fácil y sencillo recordar cómo usar la interfaz.

Frecuencia y severidad de errores (*Errors*): Frecuencia con la que los usuarios cometen errores y el nivel de gravedad o impacto de los mismos (Nielsen, 1993, como se citó en

Roth et al., 2015). Se debe entender que muchas veces los usuarios pueden llegar a cometer errores, pero se debe hacer el proyecto de titulación lo suficientemente intuitivo y fácil de usar para que no los cometan, y si los cometen, se debe procurar que no sean errores graves. Esto se puede lograr con un buen diseño de base de datos no destructivo, y con confirmaciones en acciones de gran impacto como confirmar para asegurarse de que el usuario sí quería hacer eso.

3.9.2 Heurísticas para mapas web interactivos

Marques et al. (2024) comentan que las heurísticas de Nielsen son importantes, pero para los mapas interactivos modernos, proponen 12 nuevas heurísticas que mejoran y adaptan los criterios clásicos a la tecnología y necesidades modernas. Para ello lo dividen en tres categorías:

3.9.2.1 Categoría A: Verificación del mapa

- **M1 — Visibilidad de los datos en diferentes escalas (*zoom*):** El sistema debe garantizar que los elementos espaciales y la densidad de información se adapten de forma legible e intuitiva cada vez que el usuario acerca o aleja el mapa (Marques et al., 2024). Se debe asegurar que la letra y todos los elementos que indican los reportes sean legibles y fáciles de leer, aun cuando se acerque o aleje el mapa. Para ello, se toma en cuenta que, si el usuario aleja mucho el mapa, los elementos pueden verse muy juntos y no se pueden identificar muy bien. Por lo que se deberían agrupar estos reportes en grupos para que no exista mucha densidad de elementos en pantalla que dificulten la legibilidad.

- **M2 — Herramientas de manipulación del mapa (*pan*/mover):** Las funciones para desplazarse de forma horizontal y vertical por el mapa deben ser fluidas, reactivas y fáciles de operar (Marques et al., 2024). Al igual que en la mayoría de mapas modernos,

se debe asegurar que al arrastrar el mapa, este responda fluidamente a la acción y haga exactamente lo que se quiere que haga sin movimientos extraños o lentitud.

- **M3 — Permanencia del diseño del mapa:** La estructura del mapa base, los controles de navegación y los paneles principales deben mantener una consistencia visual fija para no desorientar al usuario (Marques et al., 2024). La consistencia visual en el proyecto de titulación es clave, y para ello se debe hacer un diseño consistente desde un inicio. Se debe usar una misma paleta de colores, mismo diseño de modales, botones, títulos.
- **M4 — Visualización de la escala gráfica:** El mapa debe incluir siempre una escala gráfica visible y dinámica que se actualice de forma matemática exacta según el nivel de *zoom*, permitiendo al usuario calcular distancias reales de manera rápida (Marques et al., 2024). En el proyecto de titulación, al tratarse de un mapa orientado a reportes urbanos y no a mediciones geográficas precisas, la escala gráfica se incluye como elemento informativo complementario.
- **M5 — Convenciones cartográficas y simbología:** Los íconos, colores y capas deben respetar los estándares cartográficos universales y las expectativas del mundo real (por ejemplo, usar azul para cuerpos de agua) (Marques et al., 2024). Esto resulta importante para que los usuarios puedan visualizar de manera correcta el mapa. Se debe asegurar que los colores que se usen en el mapa, tanto para el modo claro como para el modo oscuro, sean estándares y no usar ciertos colores no estandarizados por una interfaz más llamativa.

3.9.2.2 Categoría B: Verificación de la información

- **M6 — Visibilidad de la leyenda:** La leyenda del mapa debe estar disponible de forma clara, explicando sin ambigüedades qué significa cada símbolo, línea o color presente en la pantalla (Marques et al., 2024). En el proyecto de titulación, al no tener muchos elementos, y al no contar con mucho espacio en un dispositivo móvil para mostrar una leyenda, se decidió no mostrarla. Y para identificar los reportes, se cuentan con colores y emojis, que se usarán en todo el sistema para que ayuden a identificar de manera visual sin la necesidad de una leyenda.
- **M7 — Control y consistencia de las capas de información:** El usuario debe tener la libertad de activar, desactivar o reordenar las capas de datos geográficos con controles sencillos e independientes (Marques et al., 2024). El sistema no tiene muchas capas por mostrar, solo la capa del mapa, la capa de los reportes y una capa 3D. Las primeras dos capas son partes clave del proyecto de titulación por lo que no se pueden ocultar o mover. En el caso de la capa 3D, aquí sí se puede contar con un botón para activarla y desactivarla.
- **M8 — Búsqueda de ubicaciones:** El sistema debe integrar un motor de búsqueda o caja de texto (*geocoder*) eficiente que reconozca nombres de lugares, direcciones o coordenadas, llevando al usuario al punto exacto con una retroalimentación visual clara (Marques et al., 2024). El proyecto de titulación, al basarse en reportes que se ven en tiempo real, no necesita una búsqueda al momento de crear los reportes, solamente se usará el GPS del usuario para que pueda hacer el reporte rápidamente desde donde se encuentra. Y en el caso de visualizar los reportes, tampoco se recomienda hacer una búsqueda, para evitar los temas hablados sobre estigmatización social y privacidad. El

objetivo de este proyecto de titulación es únicamente informar a los ciudadanos de su entorno inmediato.

- **M9 — Información contextual mediante ventanas emergentes (*pop-ups*):** Al hacer clic o pasar el cursor sobre un elemento del mapa, el sistema debe desplegar cuadros de información complementaria breves y limpios, evitando saturar la pantalla (Marques et al., 2024). Esto es muy importante a la hora de mostrar los reportes. Cuando los usuarios hagan clic en un reporte se debe mostrar un *pop-up* con información valiosa, sin llegar a ser demasiado grande para evitar saturar la pantalla.

- **M10 — Historial de navegación espacial (Ir al inicio / Vista por defecto):** Debe existir un botón directo que actúe como *home* para que el usuario pueda restablecer el mapa a su extensión o vista geográfica inicial si se llega a perder navegando (Marques et al., 2024). En el caso del proyecto de titulación, al solo tener una pantalla principal, no se necesita un botón *home*, pero sí se necesita un botón que pueda regresar al usuario a su ubicación actual cuando esté navegando por el mapa y quiera volver.

3.9.2.3 Categoría C: Verificación del sistema

- **M11 — Nivel de interactividad:** La interfaz debe reaccionar de manera inmediata ante las interacciones del usuario (como arrastrar el mapa o abrir herramientas de medición), ofreciendo indicadores de carga si el renderizado de datos geográficos toma tiempo (Marques et al., 2024). Al ser un mapa *offline* (según la arquitectura PWA detallada en la

sección 3.6), las interacciones del usuario se dan de manera inmediata. Pero, en el caso que el mapa aún no se haya descargado, se debería mostrar un indicador de carga si es que se tiene mucha demora.

- **M12 — Compatibilidad y adaptabilidad (Diseño responsivo):** La aplicación de mapas interactivos debe funcionar de manera óptima y fluida a través de diferentes navegadores web y adaptarse correctamente a pantallas de diversos tamaños, especialmente dispositivos móviles (Marques et al., 2024). La plataforma debe ser responsiva para todos los tipos de dispositivos que existen, asegurando así incluir a toda la variedad de usuarios existentes en Quito. Por lo que antes de finalizar con el proyecto de titulación, se debe probarlo en diferentes tamaños y navegadores distintos.

En resumen, de las doce heurísticas vistas, la plataforma sigue fielmente a: la legibilidad por escala (M1), la manipulación del mapa (M2), la consistencia del diseño (M3), las convenciones cartográficas (M5), los pop-ups contextuales (M9), la interactividad (M11) y el diseño responsivo (M12). Mientras que otras, se adaptan a las necesidades del proyecto: la escala gráfica (M4) se conserva sólo como elemento informativo y el control de capas (M7) se limita a la activación de la vista 3D, dado que las capas de mapa y de reportes son esenciales. Por último, tres heurísticas se descartan de forma justificada: la leyenda (M6), sustituida por colores y emojis ante la limitación de espacio en móvil; el buscador de ubicaciones (M8), omitido para evitar riesgos de estigmatización y privacidad y porque el reporte se ancla al GPS del usuario; y el botón de vista inicial (M10), innecesario en una interfaz de pantalla única.

3.9.3 Proceso de exploración y selección del diseño

Antes de fijar una interfaz definitiva, y siguiendo el enfoque iterativo del DCU, se optó por explorar diferentes opciones que tenían puntos de vista distintos y eran válidas para las

características de nuestro proyecto. A continuación, se van a presentar estas alternativas, tanto las descartadas como la versión final, y la justificación personal de las elecciones.

Para evaluar estas propuestas me basé principalmente en cuatro características principales que necesitaba tener la aplicación: la empatía (la interfaz debe sentirse humana), la inclusión (legibilidad para adultos mayores y usuarios con baja alfabetización digital, conforme a la WCAG 2.1 AA), la urgencia (reportar un incidente debe ser posible en pocos toques, incluso bajo incidentes rápidos) y la confiabilidad (el usuario debe percibir que los reportes aquí tienen información confiable).

Sobre estos pilares se exploraron tres direcciones estéticas. La primera, de carácter cívico e institucional, inspirada en servicios digitales gubernamentales modernos, ofrecía una jerarquía clara y una buena accesibilidad, pero a su vez resultaba fría. Para ello se optó por una segunda opción, cálida y comunitaria, lograba el tono más cercano y empático, aunque con ello no se sentía muy confiable, al asemejarse en exceso a una red social. Por último se diseñó una tercera, orientada a la naturaleza de urgencia en los reportes, con ello se creó una interfaz que alcanzaba la máxima legibilidad en condiciones adversas mediante un tema oscuro de alto contraste, pero su estética era un poco adversa y no se sentía muy empática, dejando de lado un aspecto importante.

Figura 7

Dirección A UI

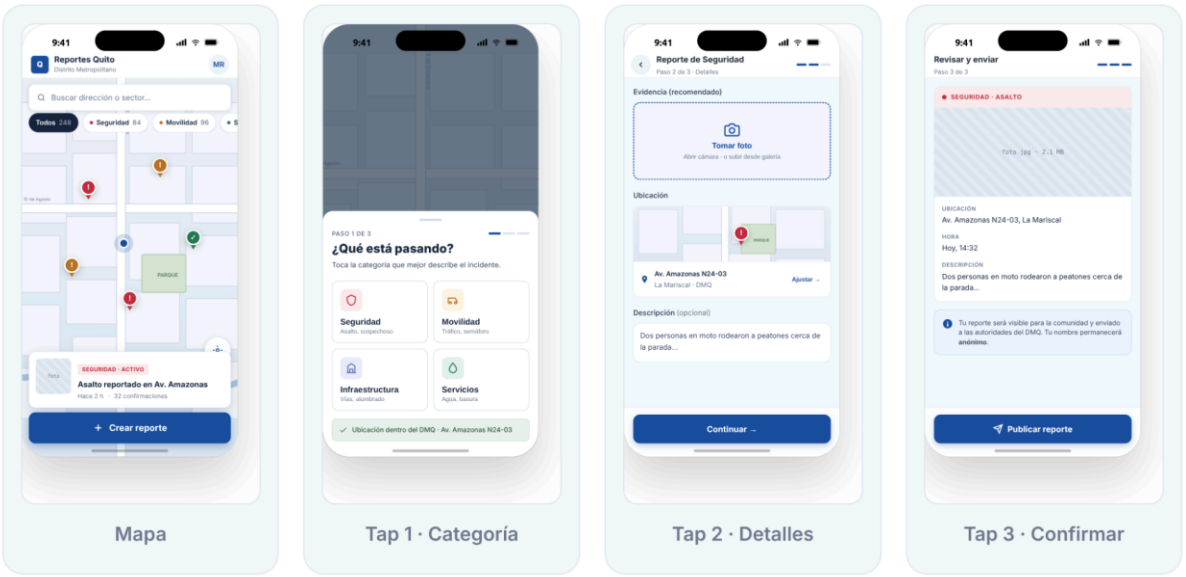


Figura 8

Dirección B UI

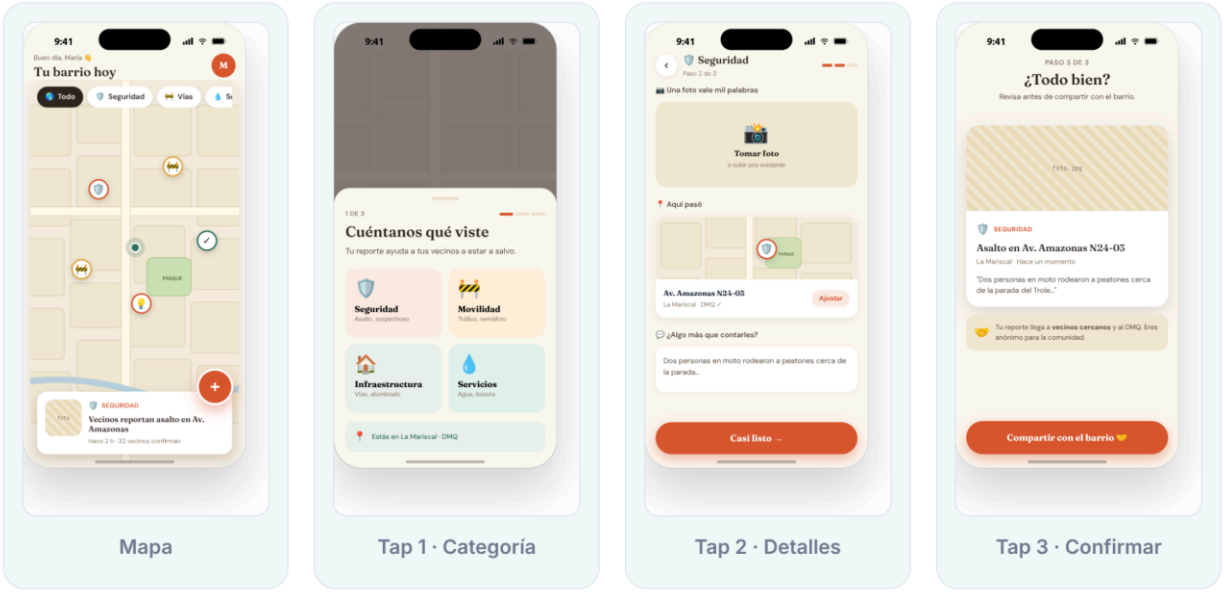
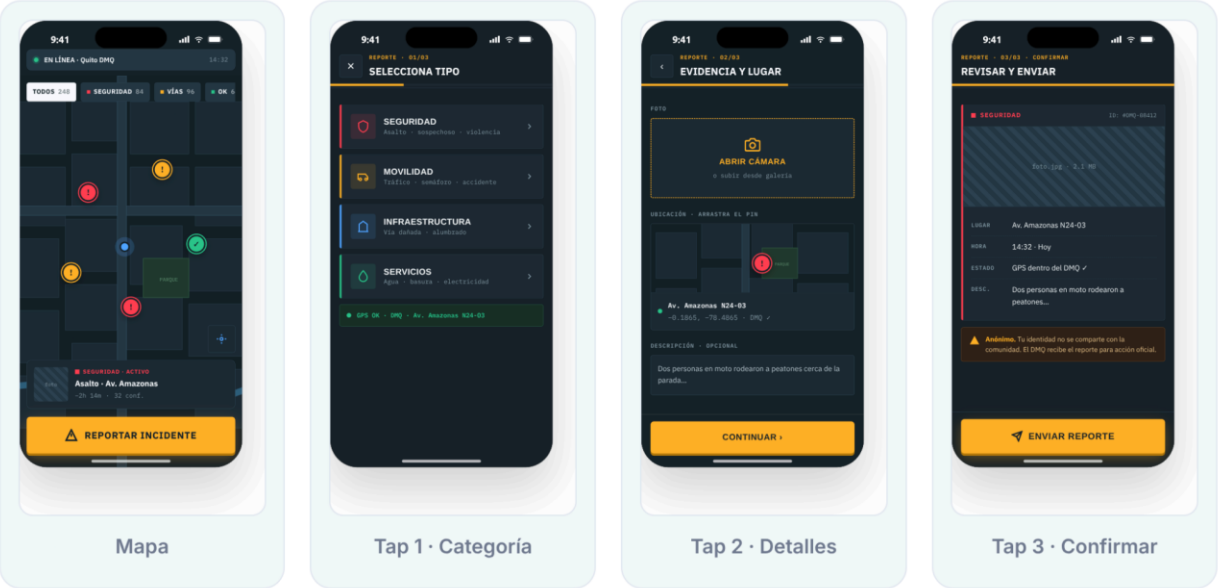


Figura 9
Dirección C UI



Al evaluar todas estas opciones, se vieron los defectos y ventajas que tenía cada uno y al final se optó por recopilar todos los elementos más significativos. Se tomó la primera opción como base, ya que era lo más cercano a lo que se quería (una aplicación confiable y legible), y a esta opción se agregó ciertos mensajes y lenguaje cercano que tenía la segunda opción, y también se mejora la legibilidad, contraste y un modo oscuro de la tercera opción.

La versión resultante, cívica pero cercana, es la que fundamenta los principios y lineamientos que se detallan a continuación.

3.9.4 Principios y lineamientos de interfaz

Los tres marcos anteriores (el DCU, las medidas de usabilidad de Nielsen y las heurísticas para mapas de Marques et al.) no se aplican de forma aislada, sino que se traducen en un conjunto de principios operativos que guían cada decisión de la interfaz. Así, la diversidad de usuarios y la prioridad de rapidez identificadas en el DCU derivan en un enfoque mobile-first y

de pocos pasos; las medidas de eficiencia y prevención de errores se concretan en controles amplios y confirmaciones; y las heurísticas cartográficas se reflejan en la consistencia visual y el tratamiento del mapa. A continuación se enuncian estos principios y, al cierre de la sección, se sintetiza su trazabilidad con los fundamentos teóricos.

Los principios de diseño que rigen la interfaz son los siguientes:

- **Diseño *mobile-first*:** concebido para dispositivos móviles desde un ancho mínimo de 320 px y para la operación con una sola mano.
- **Economía de interacciones:** cada función se resuelve en el menor número de toques posible, reduciendo la fricción de uso en toda la aplicación.
- **Controles accesibles:** botones de tamaño amplio ubicados dentro del área de alcance del pulgar.
- **Modo claro y oscuro:** claro por defecto, dada la frecuencia de uso en exteriores, con el mapa adaptándose de forma coherente al tema.
- **Lenguaje claro:** ausencia de tecnicismos y mensajes de error orientados a la solución antes que a culpar al usuario.
- **Retroalimentación constante:** un aviso visible durante al menos tres segundos tras cada acción relevante.

Como aplicación de estos principios, la creación de un reporte se organiza en cuatro pasos secuenciales (categoría, fotografía, detalles y confirmación) dentro de un panel que emerge sobre el mapa; esta segmentación evita la carga cognitiva de un formulario extenso.

Las siguientes figuras presentan el estado actual de la interfaz implementada, que materializa los principios y lineamientos descritos anteriormente. Se seleccionó un conjunto

representativo de pantallas que abarca el mapa principal, el flujo de creación de un reporte, la ficha de detalle y el comportamiento del tema visual; las pantallas restantes (inicio de sesión, perfil con insignias y panel de filtros) se incluyen en los anexos.

Figura 10

Pantalla Principa Modo Oscuro Modo Clarol

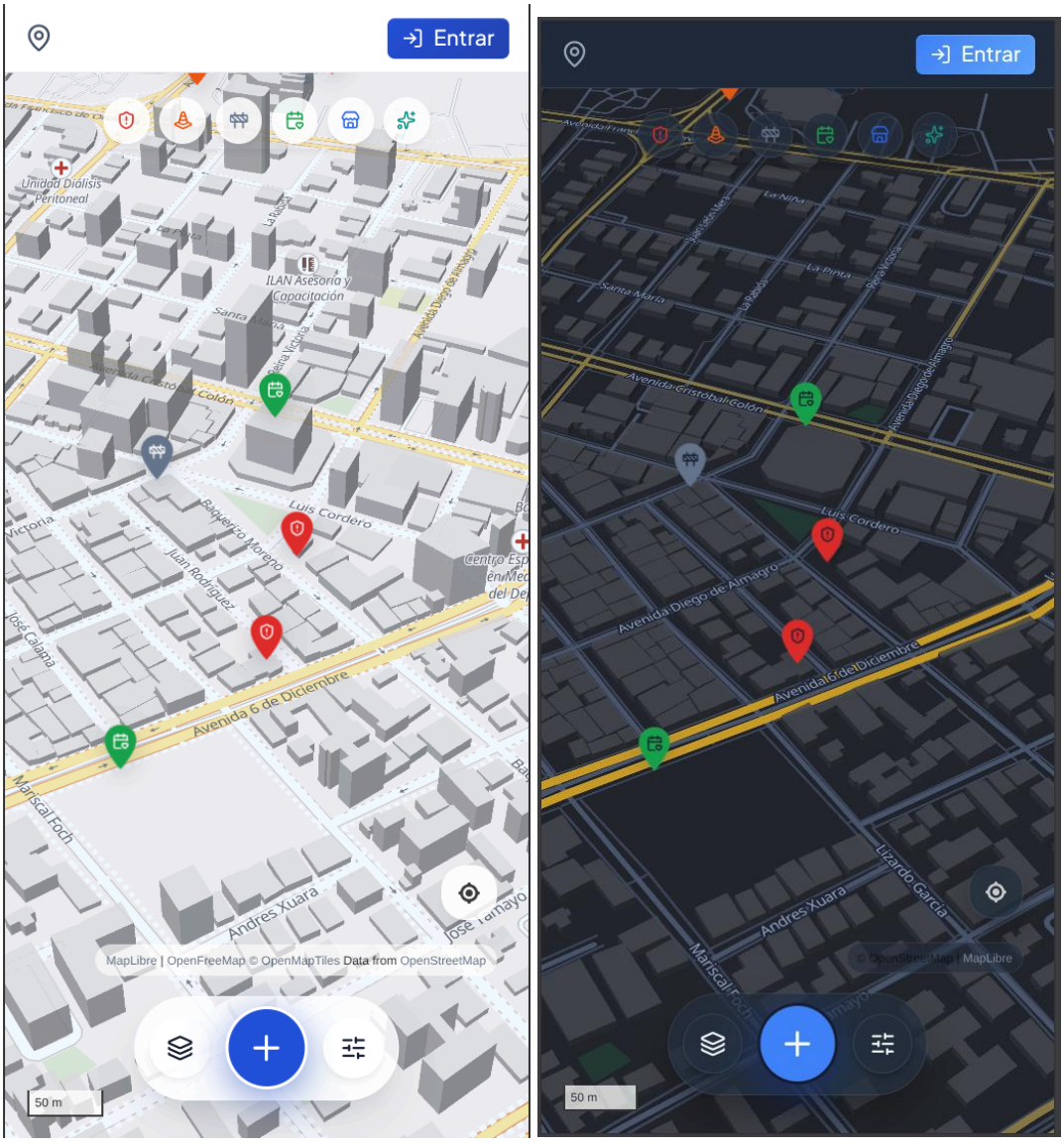


Figura 11

Creacion de Reporte UI

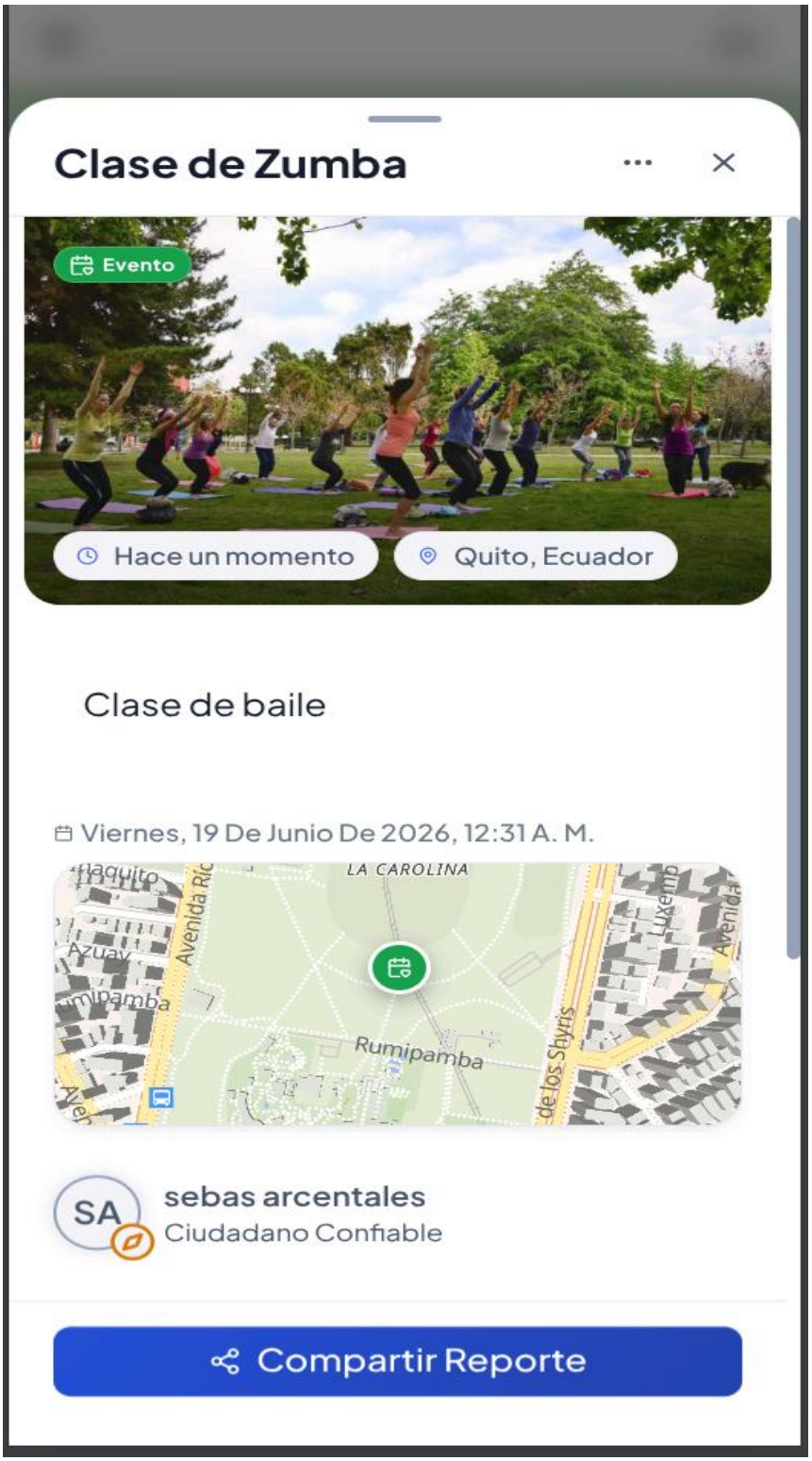


Figura **12**

Creación de Reporte 2 UI



Figura 13
Detalle de Reporte UI



Como se observa en la Figura 3.X, la pantalla principal prioriza la visualización geográfica de los reportes; el flujo de creación confirma la economía de pasos, y la comparación entre los modos claro y oscuro evidencia la coherencia visual de la interfaz.

Paleta: color primario azul #1d4ed8; cada categoría tiene su color fijo en el mapa:

Tabla 6
Paleta de Colores

Categoría	Hex
Seguridad	#E53E3E
Movilidad	#ED8936
Infraestructura	#A0AEC0
Eventos comunitarios	#48BB78
Ferias barriales	#4299E1
Limpieza comunitaria	#38B2AC

Tipografía: fuente Plus Jakarta Sans, base 16px para el cuerpo (nunca menos de 14 px en lo interactivo).

Para cerrar la sección, conviene evidenciar que cada lineamiento responde a un criterio de diseño y no a una preferencia arbitraria. La diversidad de usuarios planteada por el DCU (Roth et al., 2015) fundamenta el enfoque *mobile-first* desde 320 px y un lenguaje sin tecnicismos, de modo que la plataforma sea accesible para usuarios con distinto *hardware* y nivel técnico. Las medidas de eficiencia y prevención de errores de Nielsen se concretan, respectivamente, en la reducción del número de interacciones con controles al alcance del pulgar y en la división del formulario por pasos con confirmación en las acciones de mayor impacto, lo que reduce tanto el

tiempo de uso como el abandono. Finalmente, las heurísticas cartográficas de Marques et al. (2024) se reflejan en la consistencia de la paleta, la tipografía y los componentes (M3), en el color fijo por categoría coherente entre los modos claro y oscuro (M5) y en una ficha emergente breve junto con la agrupación por *clustering* (M9 y M1), que ofrecen una identificación visual inmediata sin saturar la pantalla.

3.10 Decisiones de Diseño Relevantes

Decisiones más importantes (contexto, opción, beneficio):

- **Tecnología de mapas (la más relevante):** en vez de mapas de imagen servidos por terceros, se usaron teselas vectoriales en un único archivo propio. Beneficio: el mapa funciona sin internet, es fluido y protege la privacidad (no envía datos del usuario a servidores externos).
- **Foto opcional:** una foto da validez, pero exigir frenaría reportes urgentes. Se dejó opcional (optimizada al subirla). Beneficio: reportar es rápido y nadie deja de avisar por no tener foto.
- **Reporte por pasos:** un formulario largo abruma. Se dividió en 4 pasos. Beneficio: menos carga para el usuario y menos abandono.
- **Funcionamiento sin internet (PWA):** Quito tiene zonas con mala señal. La app es instalable, consulta el mapa offline y envía los reportes al reconectar. Beneficio: sigue siendo útil sin conexión y no se pierde el reporte.

4 CAPÍTULO 4: Desarrollo e implementación

Ahora se aplicará la tercera fase de AUP y la más extensa. Aquí aplicaremos todo lo visto en el capítulo anterior y lo convertiremos en software que funciona.

Para hacer este desarrollo, analizando los requerimientos y las historias de usuario, se optó por organizar todo en 8 iteraciones, cada una con un objetivo que terminaba en un incremento probado e integrado.

Pero antes de comenzar el desarrollo de las iteraciones, se tuvieron que preparar dos cosas transversales, que pertenecen a la disciplina de AUP y que son importantes en el desarrollo: el entorno de trabajo, y los estándares de código y control de versiones.

4.1 Entorno y herramientas de trabajo (Disciplina de Entorno)

En esta disciplina de AUP, se busca dar herramientas y espacios de trabajo para hacer un desarrollo óptimo, reproducible, limpio y en orden. Para esto, algo importante es que el host (mi computadora) no tiene instaladas las dependencias de las herramientas directamente, sino que todas estas están totalmente aisladas, lo que ayuda a hacer un sistema reproducible y no hay conflictos de versiones. Aquí se presentará un resumen de las herramientas utilizadas, pero estarán todas listadas en anexos, al igual los motivos y decisiones documentadas (ADR) para elegir las herramientas.

4.1.1 Herramientas principales

Aquí se detallan las herramientas principales de cada capa mencionada en la arquitectura de nuestro sistema (BD, servidor, cliente)

- **Podman**: ejecuta la base de datos (PostgreSQL 16 + PostGIS 3.4) en un contenedor aislado.

- **Micromamba**: entorno aislado para el backend (Python 3.12).
- **pnpm**: gestor de paquetes del frontend (más ligero y rápido que npm), donde se usó React 19 + Vite 6.

  Y para correr todos estos módulos a la vez, se creó un script:
- **Script dev.sh**: arranca todo el sistema con un solo comando y maneja los logs de cada una. Se creó para reducir la fricción de arrancar manualmente

4.2 Buenas prácticas y organización del código *(Entorno + Gestión de Configuración)*

Esta otra disciplina trata sobre fijar qué estándares debemos seguir a la hora de desarrollar nuestro código, y la forma de versionar el trabajo, dos aspectos clave a la hora de tener buena calidad y control en nuestro desarrollo.

4.2.1 Principios de ingeniería

Aquí se detallan ciertas reglas que se seguirán en todo el desarrollo para asegurar la calidad de este, y evitar problemas en un futuro.

- **Separación de responsabilidades**: cada parte del sistema tiene una sola función clara (rutas, lógica, datos), lo que facilita mantenerlo y probarlo.
- **No repetir código (DRY)**: componentes reutilizables y estilos centralizados. Esto nos facilita al momento de hacer cambios y mantener una misma identidad visual
- **Manejo de errores en un solo lugar**: evita repetir lo mismo en cada pantalla.
- **Accesibilidad**: contraste, navegación por teclado, respeto a preferencias de movimiento.
- **Seguridad desde el diseño**: contraseñas cifradas, control de acceso, borrado lógico que conserva el historial.
- **Trazabilidad de decisiones** : todo cambio importante se registró como decisión documentada (ADR).

4.2.2 Estándares de codificación

También, se siguieron guías de estilos reconocidas, para hacer que el código siempre sea uniforme.

- Backend (Python): se siguió PEP 8, con formateo y linter automáticos.
- Frontend (JavaScript/React): convenciones idiomáticas de React + linter (ESLint).
- Por último todos los nombres de archivos, carpetas, variables, clases, componentes, funciones... Siguen los mismos principios.

4.2.3 Gestión de Configuración (control de versiones)

Para el control de versiones, se usó GIT. Diferentes ramas según el propósito(estable, integración, trabajo actual). Cuando se hacía un commit, los mensajes seguían Conventional Commits para mantener un mismo formato y tener un historial legible.

Con todo esto cada cambio que se hizo era seguro y reversible, El uso de Git junto con estas reglas ayudó a tener un historial ordenado, y facilitó mucho el encontrar y revertir versiones en casos de error, lo que facilitó la realización de las 8 iteraciones

4.3 Iteraciones de Construcción

Ahora sí, dadas las reglas y configuraciones previas, se continúa con el desarrollo de la aplicación. Como ya se comentó, existieron 8 iteraciones, cada una se completó siguiendo el mismo proceso. Comenzando con la planificación, mejorando continuamente todos los análisis que vimos en el capítulo anterior, luego se pasa al desarrollo basándose principalmente en el desarrollo con pruebas (TDD) y al completarse se pasa a la integración y de ser el caso, aplicar pruebas finales (TAD)(Se omitió el último paso de cierre, el cual está centrado en desarrollo en equipo) Para más detalle se explican los procesos más importantes en las iteraciones:

En la planificación: se seleccionan las historias de usuario más relevantes a completar y se las agrupa para asignarlas al incremento actual. Se hacen correcciones si es necesario (todos los documentos cambian a lo largo del desarrollo)

Desarrollo: Se aplica principalmente el desarrollo con pruebas (TDD): Primero se escribe una prueba automatizada que falla, luego se escribe el código para pasar la prueba y por último se refactoriza el código para limpiarlo

Y para documentarlo se usó la siguiente plantilla, la cual también se usará para mostrar lo hecho en cada iteración: **(1)** objetivo, **(2)** historias de usuario seleccionadas, **(3)** implementación, **(4)** resultado/incremento, **(5)** reto principal (cuando aplica). Estos pasos corresponden a las prácticas de AUP, exceptuando por el 5, el cual lo usaremos únicamente de forma narrativa. Las pruebas, aunque se hacían al mismo tiempo que las iteraciones, las pondremos en un capítulo aparte por temas de legibilidad y orden.

4.3.1 4.3.1. Iteración 1 Fundación

Esta es la primera iteración en nuestro proyecto. Antes de desarrollar las funcionalidades, primero se debe empezar con la base técnica sobre la que se desarrollarán. Aquí es donde levantaremos nuestra arquitectura.

4.3.1.1 (1) Objetivo de la iteración

Se debe dejar el proyecto corriendo de extremo a extremo (*frontend* + *backend* + base de datos) pero sin lógica de negocio todavía.

Se debe desarrollar y configurar el entorno de trabajo, estructura de carpetas, herramientas, bases datos, conexiones entre capas, ..etc. (todo lo explicado anteriormente)

4.3.1.2 (2) Historias seleccionadas

A diferencia de las siguientes iteraciones, la actual, no toma ninguna historia de usuario, ya que es una iteración únicamente técnica y no implementa ninguna función

4.3.1.3 (3) Implementación:

- Tres frentes en paralelo: frontend, backend y base de datos, más la configuración general del repositorio.
- En esta iteración casi no hubo ciclo TDD, porque no hay reglas de negocio que probar. Únicamente se probaron los criterios de arranque y que las capas estén correctamente configuradas y comunicadas entre sí.

4.3.1.4 (4) Resultado / incremento entregado :

- Sistema base funcionando y verificable con criterios concretos que demuestran que todo el sistema puede funcionar. Algunos de estos criterios son el levantamiento del *frontend* y visualización, levantamiento del *backend* y rutas saludables, PostGIS funcionando con 3 tablas básicas, entre otros.
- Aún no hay nada visible para el usuario final, pero la parte técnica funciona.

4.3.2 4.3.2. Iteración 2 Identidad y Acceso

Aquí se encuentra la primera iteración con funcionalidad real para el usuario. Ya hay ciertas reglas de negocio que probar por lo que ya se pueden hacer pruebas. Esta iteración se hace al principio ya que es un prerrequisito para casi todas las funcionalidades siguientes

4.3.2.1 (1) Objetivo de la iteración :

El objetivo es que nuestro sistema tenga un mecanismo completo de autenticación: registro tradicional, inicio de sesión con Google, manejo de la sesión (JWT), control de acceso a funciones protegidas y perfil de usuario.

4.3.2.2 (2) Historias seleccionadas:

Se tomaron las historias de usuario como el registro tradicional, google SSO, el modo invitado, cierre de sesión, recuperación de contraseña y eliminación de la cuenta. Para más detalle ver el anexo de historias de usuario

4.3.2.3 (3) Implementación:

- Backend: *endpoints* de registro, *login*, Google, perfil (/me), recuperación y eliminación de cuenta; emisión y validación de la sesión; middleware que protege rutas.
- *Frontend*: pantallas de *login*/registro, manejo del estado de sesión, modo invitado y vista "Mi Perfil".
- TDD: lógica de autenticación (validar credenciales, *hashing*, expiración de sesión, rechazo sin token)
- Seguridad aplicada: sanitización de inputs (XSS/SQL Injection), mensaje de login genérico, rate limiting, hashing de contraseñas.

4.3.2.4 (4) Resultado / incremento entregado

Con este incremento un usuario ya puede crear cuenta, entrar (tradicional o Google), navegar como invitado y salir; las rutas protegidas rechazan accesos sin sesión (el usuario puede ver el mapa pero no puede crear reportes)

4.3.2.5 (5) Reto principal

Una limitación del sistema actual, es mandar correos personalizados a los usuarios, usado en registro y recuperación de contraseña. No se encontró una vía gratuita y eficiente para realizar esta tarea por lo que quedó excluida y actualmente solo se simula.

Otra limitación tiene que ver con el inicio de sesión con Google. Para abrir la aplicación en el teléfono se usó un "túnel" (un servicio que expone temporalmente la app local bajo una dirección web pública que cambia en cada sesión). El problema es que Google, por seguridad, solo permite el inicio de sesión desde direcciones web previamente autorizadas y fijas. Es una restricción impuesta por las políticas de seguridad de Google, no un defecto de la implementación.

4.3.3 Iteración 3 Core Geoespacial (Mapa)

4.3.3.1 Objetivo de la iteración

En esta iteración se construyó todo el núcleo geoespacial de la aplicación. Lo que vendría a ser el mapa interactivo de Quito de nuestra aplicación. Aquí se debe poder ubicar al usuario por GPS, restringir reportes fuera de Quito, permitir marcar puntos en el mapa, traducir coordenadas a direcciones legibles, entre otras funciones.

4.3.3.2 Historias seleccionadas

Se seleccionaron varias historias de usuario el renderizado del mapa ("Como usuario, quiero ver un mapa centrado en Quito al abrir la app para ubicarme"), localización GPS continua ("Como ciudadano en movimiento, quiero ver mi ubicación actualizándose sin pulsar nada.") y selección manual de coordenadas ("Como reportador, quiero poder marcar en el mapa una ubicación exacta diferente a la de mi ubicación actual, por si el reporte se encuentra a algunos metros de distancia.")

4.3.3.3 Implementación

Aquí se implementó el uso de teselas vectoriales auto hospedadas(MapLibre GL + Protomaps PMTiles). Estas teselas nos ayudan a mostrar el mapa con mejor rendimiento, menor uso de memoria y mayor rapidez, ya que, a diferencia de otras alternativas, esta tecnología no usa imágenes sino que dibuja ciertos polígonos los cuales son muy livianos en comparación con las imágenes convencionales. Además, esto también nos ayuda en una siguiente iteración donde haremos que este mapa, gracias a su poco peso de almacenamiento, se pueda usar de manera offline.

Dentro del mapa, existen varias opciones de visualización. La primera opción es elegir entre: Modo Claro y Modo Oscuro, cada una con colores distintos y adaptados a la visualización de cada modo para un mejor contraste. La segunda opción es poder elegir entre dos tipos de mapas, uno de poco peso que en un futuro se podrá ver de manera offline y otro mapa más detallado que siempre necesitará de conexión a internet.

También existe una capa 3D que ayuda a los usuarios a identificar los lugares gracias a la visualización de los edificios.

Otra capa por encima del mapa, se agregaron los marcadores de los reportes, y el *clustering*(grupos de reportes) que se activan cuando existen muchos reportes en un área. Todos estos con una visualización de cada categoría al que pertenecen.

También se implementó:

- **Doble geofencing:** como ya se mostró en los diagramas, se valida que el punto esté dentro del DMQ tanto en el cliente (respuesta instantánea) como en el servidor (por seguridad).

- Geocodificación inversa (convertir ubicaciones en direcciones legibles): esto se aplica con un servicio externo (Nominatim), respetando su política de uso (esperar entre consultas y cachear resultados).
- Filtrado por categorías: Existen chips encima del mapa, para poder filtrar los reportes por categorías elegidas. Esto se hace sin recargar el mapa y es accesible por el teclado

4.3.3.4 Resultado / incremento entregado

Mapa plenamente funcional: el usuario se ve en el mapa, puede cambiar el estilo y detalle del mapa, puede elegir un punto válido dentro de Quito, ve su dirección , los reportes y los marcadores agrupados y filtrables.

4.3.4 Iteración 4 Reportes Ciudadanos

Con esta iteración ya podemos completar el flujo principal de la aplicación. El funcionamiento principal del sistema estaría cubierto al completar esta iteración gracias a la ayuda de los incrementos anteriores

4.3.4.1 Objetivo de la iteración

- Implementar el flujo completo de creación de un reporte: formulario (categoría, título, descripción), captura y compresión de la foto, validación de que el punto esté dentro de Quito, guardado, y una ficha para ver el reporte.
- Integración con los incrementos anteriores, usando la identidad del usuario del incremento 2 y el mapa del incremento

4.3.4.2 Historias seleccionadas

Algunas de las historias de usuario seleccionadas son: Formulario de reportes:"Como usuario, quiero llenar un formulario con categoría y descripción de lo que veo, para clasificar y

detallar mi aporte". Captura de fotografía: "Como testigo de un problema, quiero tomar una foto con mi celular y adjuntarla, para dar evidencia real".

4.3.4.3 Implementación

- *Backend*: un *endpoint* para crear el reporte que vuelve a validar el *geofencing* en el servidor (autoridad final), sube la foto al almacenamiento externo y guarda el reporte con su punto geográfico.

  Frontend: formulario con selector de categoría y contador de caracteres; captura de foto desde la cámara; compresión de la imagen en el propio navegador antes de subirla; aviso de éxito y aparición inmediata del nuevo marcador en el mapa.

4.3.4.4 Resultado / incremento entregado

El ciudadano ya puede completar todo el flujo: elegir punto válido, llenar formulario, adjuntar foto , enviar, ver su reporte en el mapa al instante. El invitado queda bloqueado del formulario y lo redirige al *login*.

4.3.5 Iteración 5 Visualización y Exploración

Ahora que ya cumplimos con todo el flujo principal del sistema. Debemos enfocarnos en mejorar la experiencia del usuario. En esta iteración mejoraremos la consulta y visualización de los reportes

4.3.5.1 Objetivo de la iteración

Optimizar cómo se cargan y exploran los reportes: traer solo los de la zona visible del mapa (no todos), filtros multicriterio (categoría, antigüedad, impacto), ficha de detalle completa, compartir reportes y una vista "Mis Reportes"(con el historial de reportes del usuario).

4.3.5.2 Historias seleccionadas

Aquí se seleccionaron las historias de usuario que cumplían el objetivo de la iteración: Carga dinámica ("Como usuario con plan de datos limitado, quiero que el mapa solo descargue reportes de la zona que estoy mirando, para no consumir mis megas."), Ficha detallada del reporte ("Como usuario, quiero tocar un marcador para ver foto, descripción y fecha, para saber qué pasó.). Filtrado multicriterio ("Como usuario, quiero filtrar por categoría, antigüedad y tipo de impacto, para ver sólo lo que me interesa en mi vida diaria."). Historial de aporte ("Como ciudadano comprometido, quiero ver la lista de todos mis reportes, para llevar control de mi participación."). Y por último también existieron otras historias de usuario importantes como Compartir Reporte y Eliminar Mis Reportes

4.3.5.3 Implementación

1. Carga por "ventana visible" (bounding box): en lugar de descargar todos los reportes de la ciudad, el mapa pide solo los que caben en la zona que el usuario está viendo. Decisión hablada en la fundamentación teórica, debido a la limitación de planes de datos de los usuarios
2. Filtros que responden al instante: categoría, antigüedad (24 h / semana / histórico) e impacto (negativo/positivo), aplicados sin recargar la página.
3. Eliminar / borrado lógico: cuando un usuario borra su reporte, se retira del mapa de inmediato, pero el dato se conserva internamente (preservar el linaje/integridad). Conecta con "seguridad por diseño" de §4.2.

4.3.5.4 Resultado / incremento entregado

La exploración de datos queda completa y eficiente: el usuario ve solo lo relevante de su zona, filtra a gusto, abre el detalle con foto, comparte y gestiona su propio historial. Todo esto con una UI y UX que va acorde a lo planificado y dando la mejor impresión al usuario.

4.3.6 4.3.6. Iteración 6 PWA y Experiencia Offline

En esta iteración, convertimos lo que actualmente era solo una página web, en una PWA (Una página instalable en celular) y con las funcionalidades offline que esta pueda otorgar.

4.3.6.1 Objetivo de la iteración

- Transformar la app en una PWA real: que se pueda instalar como una app nativa, que funcione sin conexión (ver reportes guardados, crear reportes offline) y que sincronice automáticamente lo pendiente al volver la red.
- Además, también se incluye un objetivo aparte de privacidad: el modo anónimo para reportar temas sensibles sin exponer la identidad.

4.3.6.2 Historias seleccionadas

Entre las historias de usuario existen: Instalación PWA ("Como usuario frecuente, quiero instalar la app sin pasar por Play Store, para abrirla como app nativa). Guardado de reporte offline ("Como usuario sin mucho acceso a internet, quiero escribir mi reporte y que la app lo guarde, para no perder mi esfuerzo y realizar reportes en cualquier lugar "). Sincronización automática ("Como usuario que recuperó señal, quiero que mis reportes se suban solos en segundo plano, para no acordarme de enviarlos"). Y una historia de usuario del objetivo de privacidad mencionado: Reporte en modo anónimo ("Como ciudadano que quiere denunciar algo sensible, quiero reportar sin exponer mi identidad ni dónde tomé la foto, para participar sin miedo a represalias o comprometer mi información personal.")

4.3.6.3 Implementación

1. Instalable y con caché inteligente: la app se instala como nativa y guarda en el dispositivo lo necesario (interfaz, mapa, datos) para arrancar rápido incluso en redes lentas.
2. Funciona offline: guarda los últimos reportes consultados localmente y, si no hay red, los muestra en el mapa con un aviso de "modo fuera de línea".
3. Sincronización de reportes pendientes: un reporte creado sin señal se guarda en una cola local y se envía sólo cuando vuelve la conexión.
4. Modo anónimo: El modo anónimo no elimina la relación que existe en la base de datos entre el usuario y su reporte. Esto se conserva internamente para poder mostrarlos en "Mis reportes" y tener un linaje del dato. Lo que se hizo es un pseudo anonimato, solamente se oculta esta información de los demás usuarios.

4.3.6.4 Resultado / incremento entregado

La app ya es instalable, abre como nativa, funciona sin conexión y sincroniza sola.

4.3.6.5 Reto principal

1. Sincronización offline multiplataforma: descubrir que la tecnología "ideal" (Background Sync) no funciona en iPhone obligó a rediseñar la sincronización para que funcionara en cualquier dispositivo, disparándole desde varios momentos en vez de depender de una sola API.
2. Anonimato honesto: lograr que un reporte sensible no exponga al autor de verdad no solo se debe eliminar el nombre en los reportes, hubo que eliminar los metadatos GPS de la foto, que de otro modo delataría la ubicación.

4.3.7 Iteración 7 Geocolaboración y Calidad del Dato

Esta iteración hace que nuestro sistema deje de ser un simple repositorio de reportes, para pasar a ser una inteligencia colectiva (vista en la fundamentación teórica) , donde existe validación entre vecinos, control de duplicado y una sección de comentarios

4.3.7.1 Objetivo de la iteración

Habilitar la interacción y los mecanismos que garantizan que los datos sean confiables: avisar de posibles duplicados, dejar que los vecinos confirmen o disputen un reporte, comentarlo y mostrar su "historia" (linaje).

4.3.7.2 Historias seleccionadas

Las historias de usuario seleccionadas son : Sugerencia de reporte cercano("Como ciudadano que va a reportar, quiero que la app me avise si ya existe un reporte parecido cerca, para confirmarlo en vez de duplicarlo, sin que me frene"), Confirmación entre pares ("Como usuario, quiero confirmar o disputar un reporte sin ver antes los votos de otros, para que mi valoración sea independiente y no me deje llevar por la mayoría"), Comentarios ("Como usuario con conocimiento local, quiero comentar un reporte para aportar contexto, corregir o avisar que ya se resolvió, para que la comunidad tenga voz"). Y por último Linaje del Dato ("Como usuario que consulta un reporte, quiero ver su historia (creación, si fue editado, cuántas confirmaciones tiene), para confiar en la información")

4.3.7.3 Implementación

A diferencia de iteraciones previas (flujo individual), aquí el foco es la capa social: más que pantallas, son decisiones de diseño para dejar participar sin que el sistema se manipule.

- Confirmación "ciega": se aplicó la votación a los reportes, cuando un usuario quiere dar su voto, este no ve los votos de los demás, todo esto para evitar lo que se hablaba sobre

inteligencia colectiva (la sabiduría de la multitud requiere juicios independientes) y no dar el "arrastre social" mencionado

- Mecanismo anti-duplicado: se creó este mecanismo de detección de duplicados cerca del sitio. Esto solo se usará como sugerencia, para que el usuario revise rápidamente si realmente es un duplicado. Para esto, hay dos tipos de duplicados, uno de advertencia de duplicado, que muestra al usuario una pantalla a la hora de crear el reporte y pone su atención al reporte duplicado, para verificar si es o no es. Y otro de sugerencia, que toma en cuenta ciertos reportes con poca probabilidad de duplicado(no coinciden mucho los títulos, ubicación lejana...), el cual solo se muestra como sugerencia y no interviene en el flujo de creación de reporte.
- Comentarios: los vecinos aportan contexto/correcciones; con límite de longitud y filtro de palabras vetadas (español ecuatoriano, texto normalizado).
- Linaje del dato: en la ficha, creación + indicador "editado" + confirmaciones reales. Honestidad de alcance: sin auditoría completa, se arma con los datos ya existentes (JBGE).

4.3.7.4 Resultado / incremento entregado

La plataforma deja de ser un repositorio individual y pasa a ser colaborativa: la comunidad valida, comenta y reduce duplicados. Todo lo visto aquí, son decisiones tomadas de la fundamentación teórica, desde la parte de la confiabilidad: cada reporte muestra evidencia de su confiabilidad (linaje, confirmaciones reales); hasta otros conceptos como la inteligencia colectiva y arrastre social aplicados en mecanismos que quedan blindados contra manipulación: voto ciego, disputas que no borran reportes y sugerencias que no deciden automáticamente.

4.3.8 Iteración 8 Gamificación y Confianza Ciudadana

La última iteración, aquí se aplican los conceptos vistos en motivación, donde damos incentivo para que el usuario aporte, además de mejorar la credibilidad de los reportes al tener un mecanismo de reputación, tomando funcionalidades de la iteración anterior.

4.3.8.1 Objetivo de la iteración

Aquí se aplica la confianza y reputación del ciudadano mediante un perfil que demuestre su participación histórica, un sistema de insignias que premia diferentes logros y métricas de impacto de sus reportes.

4.3.8.2 Historias seleccionadas

Aquí existe tres historias de usuario seleccionadas:

- Reputación: ("Como ciudadano activo, quiero ver mi participación histórica (antigüedad, reportes activos, confirmaciones emitidas y recibidas), para entender mi reputación.")
- Obtención de logros e insignias: (*"Como usuario frecuente, quiero ganar insignias al alcanzar hitos, para sentirme reconocido y motivado a aportar con calidad."*)
- Visualización del impacto de mis reportes: (*"Como creador de un reporte, quiero ver cuántas personas lo han visto y cuántas lo han confirmado, para sentir que mi aporte tiene impacto real."*)

4.3.8.3 Implementación

- Perfil de confianza: muestra antigüedad, reportes activos, confirmaciones emitidas y reputación (confirmaciones recibidas).
- Insignias automáticas: al alcanzar un hito (p. ej. 10 confirmaciones emitidas)

- Impacto del reporte: un contador de visualizaciones (visible para todos) y las confirmaciones recibidas (que el autor ve).

4.3.8.4 Resultado / incremento entregado

El ciudadano ve su reputación construida con datos reales, gana insignias por participar con calidad y comprueba el impacto de sus reportes. Con esto el MVP queda completo y estable

5 CAPÍTULO 5: Análisis de Pruebas

5.1 Estrategia de Pruebas

Como ya se mencionó, en AUP, las pruebas no son un hecho aislado, sino que son parte de una Disciplina, la cual se aplica a lo largo del ciclo de vida (FLOOT), no solamente al final. Pero por propósitos narrativos, se decidió ponerlas en una sección aparte.

Para las pruebas realizadas en este proyecto, se siguieron dos escenarios distintos:

- **Pruebas durante el desarrollo:** mediante el mencionado ciclo TDD, el desarrollo y las pruebas se hacían al mismo tiempo y tenían una gran conexión. Esto aseguraba que cada cosa que desarrollemos en cada iteración sea probado y validado antes de darlo por terminado.
- **Pruebas integrales finales:** aunque se ejecutarán las pruebas en el desarrollo, se vio necesario hacer pruebas integrales, que verifiquen tanto ciertas funcionalidades específicas que no se habían probado, como pruebas a la aplicación completa, que de otro modo, no se habría podido probar.

Debido a la flexibilidad y adaptación propias de la naturaleza de la aplicación, se tuvieron que hacer pruebas diferentes para cada caso. Un aspecto importante para probar fue su apartado responsive, se probó que la app funcionará y se viera correctamente tanto en computadora (>= 1024px) y celulares con pantallas pequeñas. Pero también se requirió probar ciertas funcionalidades en casos específicos, como puede ser la funcionalidad offline, el uso de la app como invitado (sin cuenta), o la aplicación instalada como PWA. Todos estos apartados sirvieron para probar la aplicación en todos los casos posibles, probando y corrigiendo ciertos escenarios que solo existían con aspectos concretos.

5.2 Pruebas Durante el Desarrollo (por iteración)

Las pruebas que aquí se van a mostrar se hacían al acabar una funcionalidad con casos definidos, siguiendo el ciclo mencionado anteriormente. Al acabar de probar se registró la prueba con su resultado para su documentación posterior, que es lo que vamos a comentar aquí. Cabe aclarar que aquí no se repetirá lo hecho en cada iteración y que sólo se mostrarán las pruebas hechas.

Para una correcta verificación, se hicieron dos tipos de pruebas dependiendo de su naturaleza. Por una parte, se hicieron pruebas técnicas (sin interfaz), que probaban el funcionamiento interno, como un correcto funcionamiento de la BD y que las reglas del negocio se cumplieran. Por otro lado, existían pruebas de pantallas, donde se verifica que la interfaz fuera correcta y funcionara correctamente desde la aplicación.

5.2.1 Resumen de lo verificado en las iteraciones principales

Aquí se van a ver ciertas pruebas que se hicieron durante el desarrollo, para esto se escogieron las iteraciones más relevantes que nos van a servir de ejemplo de cómo se desarrollaron estas pruebas.

5.2.1.1 Iteración 2 Identidad y Acceso

Comenzando con la iteración 2, la cual como ya explicamos en el desarrollo, la primera que aplicó realmente pruebas durante el desarrollo ya que aquí se desarrollaron las primeras funcionalidades: quién puede entrar, con qué datos y en qué condiciones.

Para cumplir con los objetivos de esta iteración se probaron principalmente 3 funcionalidades: que el registro acepte solo datos válidos, que el acceso sea seguro y que la recuperación de contraseña no afecte a la seguridad. En total existieron 33 casos, de los cuales solo existió un error que posteriormente fue corregido.

Además, paralelo a estas funcionalidades, también se probó que no existan vulnerabilidades a ataques comunes como inyección de comando o manipulación de credenciales. Y ciertos casos atípicos como servidores caídos, o sesión expirada, para que la app, aún con fallos, nunca deje de funcionar o tenga errores.

5.2.1.1.1 Pruebas más relevantes:

- Solo se permitió crear una cuenta con datos correctos y el consentimiento de protección de datos aceptado; los registros con email duplicado, contraseña débil o campos malformados se rechazaron correctamente.
- Ante una clave equivocada y ante un usuario inexistente el sistema devolvió el mismo mensaje, sin delatar si el correo estaba registrado, todo esto con fines de seguridad.
- Tras cinco intentos fallidos en un minuto el acceso quedó bloqueado temporalmente.
- Los intentos de inyección de comandos en el correo y los enlaces de sesión manipulados fueron rechazados, y el sistema siguió funcionando incluso con el servidor caído o la sesión expirada.
- Una contraseña excesivamente larga (de cien caracteres) fue rechazada para evitar saturar el proceso de cifrado y degradar el servicio.
- Un usuario autenticado pudo eliminar su propia cuenta, cumpliendo el derecho al olvido que exige la Ley de Protección de Datos Personales.
- La recuperación de contraseña tuvo enlaces de un solo uso y con el mismo mensaje para correos existentes e inexistentes, impidiendo averiguar qué correos tienen cuentas registradas.

5.2.1.2 Iteración 3 Core Geoespacial

Pasamos con la siguiente iteración, donde se llevó a cabo el desarrollo del mapa. Aquí se probaron diferentes funcionalidades como el geofencing (validar que un punto esté dentro de los límites de Quito), tanto en el celular como en el servidor. También se probó el comportamiento ante los diferentes permisos, como el acceso a la ubicación, donde el usuario es libre de elegir compartir o no compartir su ubicación y la app debe seguir funcionando correctamente. Visualmente también se probó que el mapa cargará correctamente el modo claro y oscuro, y los diferentes tipos de mapas, con sus detalles, colores y marcadores correctamente visualizados.

5.2.1.2.1 Pruebas más relevantes:

- El mapa de Quito cargó sin errores en modo claro y oscuro, mostrando visible el límite del Distrito.
- La aplicación reaccionó correctamente tanto si el usuario concede su ubicación (centraba el mapa en él) como si la negaba (volvía a Quito sin bloquearse).
- Un punto dentro de Quito fue aceptado y uno fuera fue rechazado, comprobando de forma independiente en el celular y en el servidor.
- Al intentar arrastrar el pin fuera de los límites del Distrito, el mapa lo devolvía automáticamente hacia adentro, impidiendo físicamente fijar un reporte fuera de zona como capa adicional de protección.
- Cuando varios reportes quedaban cerca, se agrupaban en un marcador con forma de gráfico de pastel que distinguía las categorías por color, y al tocarlo el mapa hacía un acercamiento suave hacia el grupo.
- El botón para crear un reporte distinguió la sesión: a un invitado le mostró el acceso para iniciar sesión, mientras que a un usuario autenticado le permitió continuar.

- Al elegir un punto, el sistema muestra su dirección legible y reutiliza las consultas repetidas de la misma coordenada.

5.2.1.3 Iteración 5 Visualización y Filtros

Pasamos a otra iteración con pruebas importantes. En esta iteración nos enfocamos en probar principalmente dos funciones: que los filtros elegidos por el usuario funcionen correctamente, y que la app consuma pocos datos móviles cargando únicamente los datos necesarios. También se probaron otras funciones como el historial y el borrado correcto de reportes propios.

5.2.1.3.1 Pruebas más relevantes:

- El mapa descargó únicamente los reportes del área visible, evitando gastar el plan de datos del usuario.
- Los filtros por categoría y por rango de tiempo actualizaron la vista al instante.
- Al tocar un reporte en el mapa se abrió su ficha de detalle con la información completa: la foto, el tiempo transcurrido desde su creación y su estado.
- El historial personal sólo fue accesible con sesión iniciada; sin cuenta el acceso quedó denegado.
- Cada usuario puede eliminar sus propios reportes, pero no los ajenos, y todo borrado exige una confirmación previa para evitar accidentes.

5.2.1.4 Iteración 7 Geocolaboración y Calidad del Dato

En esta iteración, como ya comentamos se agregaron las reglas sociales del sistema, como son los votos, comentarios, anti duplicados. Esta fue la iteración más difícil de probar ya que se tuvo que simular cómo interactúan las personas y probar todos los casos posibles. Aquí se

probaron las funcionalidades mencionadas como la confirmación ciega, moderación, centrándonos tanto en lo que se podía permitir, como en lo que no se podría permitir

5.2.1.4.1 Pruebas más relevantes:

- Cuando se veía un reporte ajeno, no se podía ver los votos de los demás hasta emitir el suyo, ni votar dos veces, ni valorar su propio reporte (confirmación ciega).
- Al fijar el pin cerca de un reporte parecido, el sistema sugirió sumarse en vez de duplicar, sin frenar el flujo.
- El control de cercanía no solo detectó reportes parecidos, sino que clasificó la coincidencia como de confianza alta o baja: si era alta, propuso directamente "¿es el mismo?", y si era baja, mostró un listado de reportes cercanos para que el usuario los revisará y decidiera.
- La ficha mostró el nombre real del autor del reporte o, cuando este era anónimo, la etiqueta "Ciudadano Anónimo".
- El lenguaje ofensivo en los comentarios fue censurado, incluso cuando se intentaba disfrazar con tildes para evadir el filtro.
- Un reporte que acumuló tres o más disputas pasó automáticamente a revisión
- El historial mostró su creación, edición y confirmaciones reales.

5.2.1.5 Iteración 8 Gamificación y Confianza

La última iteración está centrada en la motivación, reputación y confiabilidad. Aquí el principal objetivo era que las reglas impuestas fueran precisas y que los logros y reputación reflejaran la realidad. Entre las principales funcionalidades se probó que la reputación contará correctamente las confirmaciones reales, que las insignias no se dupliquen, y que el contador de visitas sea exacto. Aquí se verificó también, de forma medible, el pseudo anonimato: una

confirmación sobre un reporte anónimo sí suma a la reputación de su autor, porque el vínculo interno persiste, aunque el nombre no se muestre.

Pruebas más relevantes:

- La reputación se calculó sólo a partir de confirmaciones reales recibidas, excluyendo disputas y reportes borrados.
- Las insignias se otorgaron una sola vez por logro, sin duplicarse al repetir la acción, y el contador de visitas aumentó de forma exacta.
- La antigüedad y el número de reportes activos se redactaron correctamente, ignorando los reportes eliminados.
- Una confirmación recibida en un reporte anónimo sí sumó a la reputación de su autor, porque el vínculo interno se conserva aunque el nombre no se muestre a los otros usuarios (pseudo anonimato).

5.2.2 Resumen de pruebas (todas las iteraciones):

Tabla 7
Resumen de Pruebas por Iteración

Iteración (módulo)	Casos probados	Resultado
It. 1 — Fundación	3	✅ Todas
It. 2 — Identidad y Acceso	33	✅ Todas
It. 3 — Core Geoespacial	17	✅ Todas
It. 4 — Reportes	14	✅ Todas
It. 5 — Visualización y Filtros	13	✅ Todas

It. 6 — PWA y Resiliencia	12	✅ Todas
It. 7 — Geocolaboración y Calidad	13	✅ Todas
It. 8 — Gamificación y Confianza	6	✅ Todas

5.3 Pruebas Integrales Finales

Ahora vamos a hablar sobre este otro tipo de pruebas, las cuales no se realizaron durante el desarrollo, sino que se probaron al final para dar un recorrido completo de la aplicación, de principio a fin, pudiendo probar así todos los flujos completos.

Aunque en AUP no existe una sección de pruebas finales, sí menciona las pruebas de aceptación (UAT) que debe hacer el usuario final o cliente en la fase de Transición. Debido a la naturaleza de nuestro proyecto, se adaptó esta sección para hacer pruebas de aceptación del sistema.

Para probarlo se analizaron 8 flujos principales. Todos ellos probados en distintas condiciones de uso (PC / celular, con / sin internet, con / sin cuenta)

1. Identidad y Acceso: Cubren las funcionalidades de registro, login, sesión, invitado y derecho al olvido
2. Mapa, GPS y geofencing: Aquí se probó la funcionalidad de carga del mapa, ubicación y validación de zonas dentro de Quito
3. Creación de reporte: Se probó que la creación, con el formulario, foto, categoría, descripción, entre otras, funcionen correctamente con datos válidos y en diferentes casos como el modo anónimo o situación offline
4. Visualización y filtros: Se comprobó que los reportes se cargarán por zona, funcionamiento correcto de los filtros, historial y panel en computadoras

5. PWA, offline y sincronización: Se verificó el correcto funcionamiento del PWA, al instalarse, al usarse sin conexión, y su sincronización al recuperarla con la cola y reenvío de los datos.
6. Geocolaboración: Aquí se verifica que el detalle del reporte se vea bien e interactúe correctamente, un buen funcionamiento de la confirmación ciega, comentarios y el linaje del reporte.
7. Gamificación y confianza: Se prueba todo lo relacionado al perfil, como la antigüedad de la cuenta, reputación, insignias, vistas...
8. Rendimiento y Visualización: Aquí se comprobó que toda la aplicación cumpliera con los requisitos visuales y de experiencia del usuario, además de tener un rendimiento adecuado

Tabla 8
Resumen de Pruebas Finales

Flujo	Qué cubre	Casos
F1 · Identidad y acceso	Registro, login, sesión, invitado, derecho al olvido	15
F2 · Mapa, GPS y geofencing	Carga del mapa, ubicación, validación dentro/fuera de Quito	11
F3 · Crear reporte	Formulario, foto, categoría, fuera de Quito, anónimo, offline	14
F4 · Visualización y filtros	Carga por zona, filtros, historial, panel	11
F5 · PWA, offline y sincronización	Instalar, mapa sin internet, cola y reenvío al reconectar	12

F6 · Geocolaboración	Detalle, confirmación ciega, comentarios, linaje	14
F7 · Gamificación y confianza	Perfil, antigüedad, reputación, insignias, vistas	12
F8 · Transversal	Diseño adaptable, accesibilidad, rendimiento	12
Total		**101**

Al final se aprobaron 99 casos, de los cuales las fallas encontradas se corrigieron correctamente. Los dos casos que no se pudieron corregir fueron por depender de servicios externos: el login con Google y el envío real de correos. Aunque la implementación de estos fuera correcta, los errores surgieron de dependencias externas, por motivos que ya se explicaron.

Para dar un ejemplo de estas pruebas integrales se describen tres casos representativos: en el primero la aplicación protege la calidad del dato rechazando ubicaciones fuera de Quito; en el segundo la aplicación funciona correctamente sin internet, guardando el reporte y enviándolo al recuperar la señal; y en el tercero se verifica la imparcialidad de la validación, que oculta los votos previos hasta que el usuario emite el suyo.

- **F3-07 · Reportar fuera de Quito → rechazado.** Al fijar un punto fuera del DMQ, el botón de enviar se bloquea en el celular (antes de mandar nada) y el servidor también lo rechaza (doble validación). *Esperado: alerta "Ubicación inválida. Solo DMQ".*
- **F5-06/F5-07 · Crear reporte sin internet → queda en cola → se envía al reconectar.** Sin red, el reporte se guarda localmente con aviso "se enviará al

reconectar"; al volver la señal, se sincroniza solo y aparece en el servidor. *(Probar sobre la versión de producción.)*

- **F6-06 · Confirmar un reporte sin ver antes los votos de otros (confirmación ciega).** Un usuario que no es el autor no ve los conteos hasta emitir su propio voto, para no dejarse llevar por la mayoría.

5.4 Resultados, Problemas Encontrados y Correcciones

Acabada la totalidad de las pruebas, ahora podemos mostrar los resultados de las mismas, además de mostrar ejemplos de qué problemas existieron y cómo se solucionaron

5.4.1 Resultados

5.4.1.1 Resumen de los resultados de las pruebas:

Tabla 9
Resumen Resultados Pruebas

Iteración	Casos	Aprobados	Observaciones	Pendientes
It. 2 Identidad	~33	~33	0	0
It. 3 Geoespacial	17	17	0	0
It. 5 Visualización	13	13	0	0
It. 7 Geocolaboración	13	13	0	0
It. 8 Gamificación	6	6	0	0

5.4.1.2 Pasada integral (8 flujos, 101 casos):

Flujo	Casos	Aprobados	Observaciones	Bloqueados (externo)

F1 Identidad y acceso	15	13	0	2
F2 Mapa, GPS y geofencing	11	11	0	0
F3 Crear reporte	14	14	0	0
F4 Visualización y filtros	11	11	0	0
F5 PWA, offline y sincronización	12	12	0	0
F6 Geocolaboración	14	14	0	0
F7 Gamificación y confianza	12	12	0	0
F8 Transversal	12	12	0	0
Total	**101**	**99**	**0**	**2**

5.4.2 Problemas encontrados y correcciones

Durante las pruebas, existieron ciertos errores que debían ser corregidos. La mayoría se corrigen de forma rápida y sin documentación por la metodología TDD, pero algunos importantes sí llegaron a documentarse. Para un ejemplo de esto, aquí se presentarán algunos problemas relevantes con sus respectivas soluciones

- **Bug de recuperación de contraseña**: debido a un error de tipo de dato en el id del usuario lo que ocasiona se cayera con un error 500; para corregirlo se corrigió el problema técnico y se agregó un manejo de errores que muestra "token inválido o expirado"

- **El sistema confundía "contraseña incorrecta" con "sesión expirada”:** En la aplicación, se trataba cualquier respuesta 401 como sesión caducada, así que al equivocarte de contraseña en el *login* te cerraba la sesión y te redirige en vez de solo avisar del error. Se corrigió excluyendo los *endpoints* de *login*/registro de esa lógica.
- **Marcadores duplicados en los bordes del mapa**: cuando dos zonas vecinas del mapa se juntaban, un mismo grupo de reportes llegaba a dibujarse dos veces. Para resolverlo se añadió un control para que cada grupo apareciera una sola vez.
- **Colores poco legibles en modo oscuro**: en las pruebas de pantalla, se detectó que, en el tema oscuro, las calles y los parques casi no se distinguían del fondo, por lo que se ajustaron los colores para que tuvieran buen contraste.
- **Conflicto entre *cluster* y colocación de pin**: Al tocar un grupo de reportes(*cluster*), se colocaba el pin para crear el reporte; para corregirlo se detuvo la propagación del evento antes de que llegara al mapa
- **Modales apilados**: Al abrir modales como los filtros, o detalle del reporte, entre otros, estos se podían superponer uno con otros, creando una mala experiencia en el usuario; para resolverlo a la hora de abrir un modal, el anterior se cierra automáticamente

5.5 Verificación de Calidad (Requisitos No Funcionales)

Como se ha mencionado, la calidad del sistema no se da solo por el correcto funcionamiento de sus funciones, sino también por sus cualidades: que cargue rápido, que funcione sin señal, que cualquier persona pueda usarlo y que se adapte a pantallas pequeñas. Todas estas cualidades se verificaron y se mostrarán a detalle a continuación:

- Rapidez. Para esta cualidad se debe tomar en cuenta que la aplicación está pensada para usarse desde el celular y con datos móviles. Por lo que la rapidez es un aspecto muy importante, para ello, una funcionalidad que comentamos y es muy importante en este sentido, es que las fotografías de los reportes se comprimieran antes de enviarse, manteniéndose por debajo del medio megabyte para no saturar la conexión. Además, también se verificó que la primera vista de la pantalla apareciera de forma casi inmediata.
- Funcionamiento sin internet. Se verificó que el mapa de Quito se cargará desde el almacenamiento del propio dispositivo aun sin señal y que un reporte creado sin conexión quedará guardado en una cola local y se reenviará automáticamente al recuperar el internet, sin que el usuario perdiera su información.
- Accesibilidad. Se revisó que la aplicación pudiera usarse por personas con distintas capacidades: contraste de color suficiente para leer con comodidad, botones e iconos con un nombre para que un lector de pantalla lo pueda leer y la posibilidad de manejar toda la interfaz con el teclado.
- Diseño adaptable. Se comprobó que la interfaz se viera y funcionara correctamente desde pantallas muy pequeñas hasta el computador, revisando en anchos de 320, 414 y 1024 píxeles, sin que ningún elemento se desbordará o quedará inutilizable.
- Seguridad. Se confirmó que las contraseñas se guardan cifradas, que el acceso a las zonas privadas exigiera una sesión válida y que los intentos de inyección o de manipular la sesión fueran rechazados, apoyándose en las pruebas de seguridad y robustez realizadas en la Iteración 2.

5.6 Pruebas de usabilidad

5.6.1 Introducción y objetivos

Para validar que el sistema es usable realmente, era necesario hacer un conjunto de pruebas que demostraran, con usuarios reales, que la aplicación es apta para cada tipo de usuario, y poder obtener un feedback externo. Esto complementa las pruebas anteriores, las cuales solamente valida que el sistema funcione correctamente.

Para estas pruebas se evalúan cuatro de las cinco medidas de usabilidad de Nielsen (1993): la facilidad de aprendizaje (learnability), la eficiencia (efficiency), los errores (errors) y la satisfacción subjetiva (satisfaction), esta última mediante el cuestionario SUS. La memorización (memorability) no se evalúa porque requiere una segunda sesión con los mismos usuarios tras un periodo sin uso, lo cual queda fuera del alcance. Las pruebas se realizan con 5 usuarios, número que según Nielsen (1993) permite identificar alrededor del 85% de los problemas de usabilidad de una interfaz.

5.6.2 Metodología

5.6.2.1 Perfil de participantes

- Ciudadanos del DMQ, no expertos en tecnología ni en el sistema
- Criterios mínimos de diversidad (al menos uno por grupo):
    - Un usuario mayor de 40 años (para probar learnability en usuarios con menos experiencia digital)
    - Un usuario con smartphone de gama media o baja (para probar rendimiento real)
    - Un usuario que nunca ha usado una app de reportes ciudadanos

- Criterio de exclusión: personas involucradas en el desarrollo o evaluación del proyecto

Tabla 10
Participantes Pruebas Usabilidad

Participante	Rango de edad	Dispositivo	¿Usó antes una app de reportes?
P1 — Paulina Arciniega	46–55 (50)	Smartphone gama media	No
P2 — Alejandra Medina	18–25 (21)	Smartphone gama alta	Sí
P3 — Gustavo Jara	18–25 (21)	Smartphone gama media	Sí
P4 — Dominique Verkade	18–25 (22)	Smartphone gama alta	Sí
P5 — Marco Tello	36–45 (44)	Smartphone gama baja	No

La muestra cumple los tres criterios mínimos de diversidad: P1 (50 años) y P5 (44 años) aportan el perfil de mayor edad y menor experiencia digital; P5 además usa un dispositivo de gama baja, lo que permite observar el rendimiento real de la aplicación bajo restricciones de hardware; y tanto P1 como P5 nunca habían utilizado una aplicación de reportes ciudadanos. Ningún participante estuvo involucrado en el desarrollo ni en la evaluación del proyecto.

5.6.2.2 Tareas

Tabla 11
Tareas Pruebas de Usabilidad

N°	Tarea	Flujo que cubre	Métrica principal
T1	"Crea una cuenta en la plataforma"	Autenticación	Learnability, errores
T2	"Reporta un bache que encontraste cerca de donde estás"	Creación de reporte	Efficiency, errores

T3	"Encuentra reportes de seguridad en tu zona y filtra para ver sólo esa categoría"	Visualización y filtros	Efficiency, learnability
T4	"Abre el detalle de cualquier reporte y deja tu opinión sobre si es verídico"	Geo colaboración (voto)	Learnability, errores
T5	"Entra a tu perfil y dime cuántos reportes llevas y si ganaste alguna insignia"	Gamificación / perfil	Learnability

5.6.2.3 Instrumentos de evaluación

- Protocolo think-aloud: el participante narra en voz alta lo que piensa mientras usa la app
- Planilla de observación: una tabla donde el evaluador anota por tarea si se completó, cuánto tardó, cuántos errores cometió y dónde se bloqueó
- Cuestionario SUS (System Usability Scale; Brooke, 1996) post-prueba: estándar de 10 preguntas en escala Likert 1-5, demora 2 minutos, produce una puntuación de 0 a 100

5.6.2.4 Métricas para registrar por tarea

- Tasa de completitud: completó / no completó / completó con ayuda
- Tiempo de completitud (en segundos)
- Número de errores (toques incorrectos, acciones equivocadas antes de completar)
- Puntos de bloqueo: dónde se detuvo o expresó confusión

5.6.3 Procedimiento

Descripción del protocolo de la sesión (el orden exacto de lo que pasa):

1. Presentación e introducción (explicar que se prueba la app, no al usuario)
2. Firma del consentimiento informado (obligatorio, incluir que la sesión puede grabarse)

3. Contexto breve: qué es la plataforma, sin explicar cómo se usa
4. Ejecución de tareas en orden, con think-aloud activo
5. El evaluador solo interviene si el participante lleva más de 3 minutos bloqueado
6. Cuestionario SUS al finalizar todas las tareas
7. Pregunta abierta final: "¿Qué fue lo más difícil? ¿Qué cambiarías?"

5.6.4 5.6.4 Resultados

5.6.4.1 Tabla de completitud por participante y tarea

Leyenda: ✅ completó solo / ⚠️ completó con dificultad / ❌ no completó

Participante	**T1**	**T2**	**T3**	**T4**	**T5**
P1	⚠️	⚠️	⚠️	⚠️	✅
P2	✅	✅	✅	✅	✅
P3	✅	✅	✅	✅	✅
P4	✅	⚠️	✅	⚠️	✅
P5	⚠️	⚠️	⚠️	⚠️	✅

Las cinco tareas fueron completadas por los cinco participantes (tasa de completitud del 100 %); ninguna terminó en abandono. La dificultad se concentró en T2 (creación de reporte) y

T4 (detalle y voto), donde los dos usuarios de mayor edad y sin experiencia previa (P1 y P5) necesitaron varios intentos. T5 (perfil y gamificación) fue resuelta sin fricción por todos.

5.6.4.2 Tabla de tiempos promedio por tarea

Tabla 12
Tabla de Tiempos Pruebas de Usabilidad

Participante	T1	T2	T3	T4	T5
P1	120	180	110	240	30
P2	20	70	45	90	20
P3	25	65	50	85	25
P4	18	85	55	120	22
P5	95	150	100	200	40
Promedio	55.6	110.0	72.0	147.0	27.4

5.6.4.3 Tabla de errores por tarea

Tabla 13
Errores por Tarea Pruebas de Usabilidad

Participante	T1	T2	T3	T4	T5
P1	3	4	2	5	0
P2	0	1	0	1	0
P3	0	1	1	1	0
P4	0	2	1	2	0
P5	2	3	2	3	1
Total	5	11	6	12	1

La mayor cantidad de errores se acumuló en T2 (11) y T4 (12). En T2 los errores provinieron de buscar el botón de "usar mi ubicación “y de la imprecisión del punto geolocalizado. En T4 se debieron a la dificultad para mover el mapa y a la búsqueda del botón de voto dentro de un detalle que solo mostraba el resumen. T5 fue prácticamente libre de errores (1 en total).

5.6.4.4 Puntuaciones SUS

Referencia (Bangor et al., 2008): ≥ 68 = por encima del promedio · ≥ 80 = buena · ≥ 90 = excelente

Tabla 14
Puntaje SUS Pruebas de Usabilidad

Participante	Puntaje SUS (0–100)
P1	67.5
P2	90.0
P3	87.5
P4	85.0
P5	70.0
Promedio	80.0

El promedio de 80.0 ubica al sistema en el rango de usabilidad **buena** según la referencia de Bangor et al. (2008), claramente por encima de la media de la industria (≈68). Los participantes con experiencia previa (P2, P3, P4) puntuaron entre 85 y 90 (rango "bueno" a casi "excelente"), mientras que los usuarios de mayor edad y sin experiencia (P1 y P5) puntuaron 67.5 y 70, en torno al umbral del promedio. Esta diferencia es coherente con la fricción observada en T2 y T4 y señala que el principal margen de mejora está en el primer reporte y en la exploración del mapa para usuarios noveles.

5.6.4.5 Problemas encontrados

Severidad: **Alta** = impide completar la tarea o causa pérdida de datos · **Media** = genera confusión o demora notable pero el usuario completa · **Baja** = detalle cosmético o molestia menor

Tabla 15 Problemas Encontrados Pruebas de Usabilidad

Problema observado	Tarea	Frecuencia (nº usuarios)	Severidad
Algunos usuarios no notan que pueden ver el detalle completo	T4	3	Media

del reporte y creen que solo se muestra el resumen (problema de percepción)			
El campo de título del reporte no abre el teclado al tocarlo (no se puede escribir)	T2	2	Alta
No se encuentra el botón para "usar mi ubicación actual" al crear el reporte	T2	3	Media
La ubicación obtenida por GPS no es del todo precisa respecto al lugar real	T2	3	Media
El reporte siempre se ancla en la ubicación del usuario y no permite marcar el punto exacto del bache	T2	2	Media
La pantalla de selección de "sector" no se entiende con claridad	T1	2	Media
No queda claro que se deben aceptar los términos y condiciones; el enlace no es visible	T1	2	Media
La hoja del incidente tapa los iconos y controles inferiores del mapa	T3	2	Media
Dificultad para desplazarse y hacer zoom en el mapa	T4	2	Media
Al aplicar un filtro no se muestran de inmediato los incidentes existentes de esa categoría	T3	2	Media

5.6.5 Correcciones aplicadas

A partir de los problemas detectados se priorizaron primero los de severidad alta y luego los de severidad media con mayor frecuencia. Las correcciones aplicadas tras las sesiones fueron:

- **Teclado del campo título (Alta, T2):** se corrigió el bug que impedía abrir el teclado al enfocar el campo de título; el campo ahora recibe el foco y abre el teclado automáticamente al tocarlo.

- **Botón de ubicación (Media, T2):** se hizo el botón "Usar mi ubicación actual" más grande y vistoso, y se le añadió un efecto de titileo (parpadeo) cuando la ubicación aún no está activa, de modo que llame la atención del usuario y resulte fácil de encontrar.
- **Pin y bottom sheet sobre el mapa (Media, T3):** ahora, cada vez que el usuario mueve el mapa, el pin junto con su hoja inferior (bottom sheet) se ocultan automáticamente para no estorbar la exploración de la zona.
- **Términos y condiciones (Media, T1):** se hizo visible la casilla de aceptación de términos y condiciones con su enlace, requerida antes de completar el registro.
- **Selección de sector (Media, T1):** la selección del sector ahora se realiza directamente sobre un mapa, en lugar de una lista o un campo de texto, lo que la hace más clara e intuitiva.

Algunos problemas observados no requirieron cambios. El detalle del reporte ya mostraba toda la información del incidente desde su implementación; la dificultad registrada fue de percepción (algunos usuarios no notaron que podían desplazarse para ver el contenido completo), pero como la vista es entendible y la información ya está disponible, no se aplicaron modificaciones. Por otro lado, ya se tenía implementado una manera de poner la ubicación exacta por si el gps tenía cierto fallo, por lo que tampoco requirió cambios. Por último, la fluidez del desplazamiento del mapa (T4) y la actualización inmediata del filtro (T3) se documentan como mejoras menores cuya validación definitiva requeriría una segunda ronda de pruebas.

5.6.6 Conclusiones de las pruebas de usabilidad

- **Puntaje SUS y significado:** el sistema obtuvo un SUS promedio de 80.0 / 100, que según Bangor et al. (2008) lo ubica en el rango de usabilidad buena, claramente por encima de la media de la industria (68). Los usuarios más

familiarizados con plataformas similares puntuaron en el rango "bueno" a casi "excelente" (85–90) y los usuarios nuevos, de mayor edad tuvieron un promedio (67.5 - 70), que nos comunica dónde debemos mejorar.

- **Medidas de Nielsen bien resueltas:** la eficiencia quedó validada en las tareas de visualización y perfil (T3 y T5, con tiempos promedio de 72 s y 27.4 s y casi sin errores), y la facilidad de aprendizaje se evidenció en que las cinco tareas se completaron al 100 % sin abandonos, incluso por usuarios que nunca habían usado una app de reportes.
- **Medidas con fricción:** los errores y la eficiencia mostraron fricción en la creación del reporte (T2) y en el detalle/voto (T4), concentrando el 79 % de los errores totales. Las causas principales fueron el campo de título que no abría el teclado correctamente en ciertas situaciones y el no encontrar el botón de ubicación, también debido a su poco uso en otras aplicaciones como google maps. Ambas fueron corregidas y la percepción de que el detalle del reporte solo mostraba el resumen, problema que no requirió cambios al tratarse de una vista que ya muestra la información completa.
- **Cumplimiento del objetivo:** considerando los resultados mostrados de la tasa de completitud del 100 % y un SUS por encima del promedio en una muestra que incluyó usuarios de gama baja, de mayor edad y sin experiencia previa, se concluye que el sistema es usable para ciudadanos con distinto nivel técnico, aunque la curva de aprendizaje del primer reporte es más pronunciada para usuarios no familiarizados con la tecnología.

- **Problemas sin corregir:** la precisión de la geolocalización depende del dispositivo del usuario, por lo que queda fuera del alcance y para mitigarlo se usa la ya existente función de marca una ubicación manualmente. La navegación del mapa en dispositivos de gama baja y la validación de las mejoras de filtro requieren una segunda ronda de pruebas, lo cual excede el alcance de esta evaluación.

6 CAPÍTULO 6: Conclusiones y recomendaciones

6.1 Conclusiones

6.1.1 Verificación de objetivos

- Objetivo 1 (Fundamentación teórica): Se construyó un marco que cubre VGI, crowdsourcing, calidad del dato (medidas ISO + indicadores abstractos), sesgos, motivación y privacidad. Todas las decisiones de diseño nacieron del marco teórico
- Objetivo 2 (Análisis de requerimientos): Se crearon 41 requisitos funcionales y 17 requisitos no funcionales que definen claramente todo lo que hacía el sistema y como lo hacía, los cuales fueron útiles en las 8 iteraciones de desarrollo
- Objetivo 3 (Diseño UI/UX) Se creó un diseño del sistema, tomando de base la fundamentación teórica. Se aplicó correctamente el DCU tomando como referencia las 5 medidas de Nielsen, las 12 heurísticas para mapas. Para la decisión de diseño final se pasaron por varias alternativas y se tomaron en cuenta varios principios como la economía de interacciones, lenguaje claro, retroalimentación constante, entre otras.
- Objetivo 4 (Implementación): Para la implementación se realizó un MVP funcional dividido en 8 iteraciones, cada iteración agregando funciones integradas y probadas. Entre estas iteraciones se cubrieron funcionalidades como la autenticación, mapa offline, reportes, geo colaboración, gamificación y privacidad, dejando así un sistema completo y funcional
- Objetivo 5 (Pruebas): Se realizaron pruebas técnicas durante el desarrollo y pruebas integrales al finalizar todo el sistema, en total se crearon 101 casos de aceptación de los cuales aprobaron 99 de ellos, los restantes se debieron a limitaciones de servicios externos.

6.1.2 6.1.2 Conclusiones generales

6.1.2.1 Sobre la viabilidad tecnológica

La elección tecnológica en el proyecto fue correcta, y se adaptó a la necesidad que existía. La elección de PWA con *offline-first* fue una decisión acertada al poder captar la mayoría de usuario, eliminando la barrera de instalación y un funcionamiento correcto en dispositivo con hardware y conexión limitada. Además, el uso de teselas vectoriales ayudó a que la descarga y visualización del mapa mejorará considerablemente, hubiera mayor detalle en el mapa, además de poder alojarla en el propio dispositivo para su uso offline y no depender de servicios externos.

6.1.2.2 Sobre la aplicación del marco teórico

Para asegurar la calidad de los datos en el sistema, se investigaron y utilizaron diferentes mecanismos como la votación ciega, anti duplicados, *geofencing*, gamificación, reputación, entre otros. Para la privacidad, además de aplicar el diseño de minimización de datos (no recopilar lo que no se necesita) ,se implementó el modo anónimo, que en verdad es un pseudo anonimato, el cual oculta la identidad a los otros usuarios, tanto para no revelar información crítica como su ubicación como para no exponerse en reportes con información sensible. Para la motivación, se implementan diferentes estrategias como la gamificación, el cual está compuesto de insignias, reputación y visualizadores de impacto que responde a los conceptos vistos de factores extrínsecos identificados

6.1.2.3 Sobre Aup

La metodología AUP, fue ideal para el desarrollo del actual proyecto, siendo muy ágil y rápida, pero con su componente documental que se pedía en trabajos como el actual. Además como ya se comentó, esta metodología cumplía con una estructura similar al pedido en el actual trabajo por lo que su implementación no necesito muchas adaptaciones. Por último, el desarrollo

por iteraciones y la forma de trabajo que esta metodología requiere, como el ciclo TDD, iteraciones…, facilito mucho el desarrollo; permitió tener una guía clara del desarrollo, una forma de probar que la funcionalidad fuera correcta y un camino para agregar funcionalidades

6.1.3 Limitaciones

Estas son las limitaciones honestas que deben reconocerse:

- Escala no probada: el sistema fue probado con datos de prueba, no con carga real de usuarios concurrentes o miles de reportes
- Login con Google en desarrollo: la restricción de Google OAuth para URLs dinámicas impide probar SSO en entorno local con dispositivo móvil real
- Correos electrónicos simulados: el flujo de registro y recuperación de contraseña simula el envío sin entregar correos reales
- Retención dependiente de resolución: una de las mayores motivaciones vistas para los usuarios a largo plazo es ver sus reportes resueltos, lo cual requiere integración con entidades externas fuera del alcance actual
- Sesgo de participación inevitable: como predice la regla del 1%, la plataforma necesita una gran cantidad de usuarios para que los datos sean representativos del DMQ

6.2 Recomendaciones

6.2.1 Trabajo futuro técnico

Para un trabajo futuro se deben completar funcionalidades críticas que esta aplicación necesita como el servicio real de envío de correos y un correcto funcionamiento del *login* con Google. Además, se deberían hacer pruebas más profundas como Auditoría formal de accesibilidad (WCAG 2.1 AA) y pruebas de carga y rendimiento con datos reales.

Otras mejoras, no tan importantes, sería la implementación de notificaciones push reales mediante Web Push API (ya tiene la infraestructura PWA), un módulo de administración con interfaz gráfica para moderación manual de contenido sensible, un sistema de detección de comportamiento automatizado y antispam (*bots*, ataques similares al descrito por Zhou et al. en Waze), validación asistida por IA para clasificación de categorías o detección de contenido inapropiado en imágenes

6.2.2 Trabajo futuro de adopción

El mayor desafío de aplicación con datos voluntarios es la adopción, por lo que para que esta adopción sea correcta, se deben seguir ciertas recomendaciones. Estrategia de lanzamiento focalizada, no tener campañas masivas genéricas, sino aplicar el lanzamiento en comunidades ya organizadas y con gran identidad local. Además, es importante identificar al 1% de usuarios generadores (administradores de grupos, líderes comunitarios), los cuales generan los datos para atraer a más usuarios. Si esto último no es suficiente, se deberían crear datos iniciales para llenar la aplicación de reportes que motiven a los demás usuarios a participar. Otro aspecto importante a comunicar es diferenciar a nuestra aplicación al agregar todo tipo de reportes, como eventos, los cuales diferencia de otras aplicaciones y eliminan ciertas estigmatizaciones en los usuarios Por último, también se comentaron que estrategias como el uso de redes sociales, solo son válidas para dar un pico de interés inicial y se deben combinar con otras estrategias de retención.

7 Referencias

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction*, *24*(6), 574–594. https://doi.org/10.1080/10447318.2008.10470776

Biørn-Hansen, A., Majchrzak, T. A. y Grønli, T. M. (2018). Progressive Web Apps for the Unified Development of Mobile Applications. En T. A. Majchrzak, T. M. Grønli y A. Biørn-Hansen (Eds.), *Web Information Systems and Technologies* (pp. 64-86). Springer. https://doi.org/10.1007/978-3-319-93527-0_4

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. En P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry (pp. 189–194). Taylor & Francis.

Dunkel, A., Löchner, M., & Burghardt, D. (2020). Privacy-aware visualization of volunteered geographic information (VGI) to analyze spatial activity: A benchmark implementation. *ISPRS International Journal of Geo-Information*, *9*(10), Artículo 607. https://doi.org/10.3390/ijgi9100607

Flanagin, A. J., & Metzger, M. J. (2008). The credibility of volunteered geographic information. *GeoJournal*, *72*(3-4), 137-148. https://doi.org/10.1007/s10708-008-9188-y

Flora, H. K., & Chande, S. V. (2014). A systematic study on agile software development methodologies and practices. International Journal of Computer Science and Information Technologies, 5(3), 3626-3637.

Fritz, S., See, L., & Brovelli, M. (2017). Motivating and sustaining participation in VGI. En G. Foody, L. See, S. Fritz, P. Mooney, A.-M. Olteanu-Raimond, C. C. Fonte, & V. Antoniou (Eds.), *Mapping and the citizen sensor* (pp. 93–117). Ubiquity Press. https://doi.org/10.5334/bbf.e

Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal*, *69*(4), 211-221. https://doi.org/10.1007/s10708-007-9111-y

Marques, A. P. S., Meirelles, A. A. T. y Silva, J. F. C. (2024). Validation and Refinement of Usability Heuristics for Interactive Web Maps. *Proceedings of the 20th International Conference on Web Information Systems and Technologies*. https://doi.org/10.5220/0012691300003690

McIntosh, S., Pérez-Ramos, J., Demment, M. M., Morris, A., Glaros, A., Lomonaco, C., Hazuda, H. y Fischer, L. (2016). Development and Implementation of Culturally Tailored Offline Mobile Health Surveys. *JMIR Public Health and Surveillance*, *2*(1), e28. https://doi.org/10.2196/publichealth.5408

Mooney, P., Olteanu-Raimond, A.-M., Touya, G., Juul, N., Alvanides, S., & Kerle, N. (2017). Considerations of privacy, ethics and legal issues in volunteered geographic information. En G. Foody, L. See, S. Fritz, P. Mooney, A.-M. Olteanu-Raimond, C. C. Fonte, & V. Antoniou (Eds.), *Mapping and the citizen sensor* (pp. 119–135). Ubiquity Press. https://doi.org/10.5334/bbf.f

Nielsen, J. (1993). *Usability Engineering*. Academic Press.

Pak, B., Chua, A., & Vande Moere, A. (2017). *FixMyStreet* Brussels: Socio-Demographic Inequality in Crowdsourced Civic Participation. *Journal of Urban Technology*, *24*(2), 65-87. https://doi.org/10.1080/10630732.2016.1270047

Riddell, A. (2023). Intersecting positionalities and the unexpected uses of digital crime and safety tracking in Brooklyn. *Social Inclusion*, *11*(3), 30–40. https://doi.org/10.17645/si.v11i3.6615

Roth, R. E., Ross, K. S. y MacEachren, A. M. (2015). User-Centered Design for Interactive Maps: A Case Study in Crime Analysis. *ISPRS International Journal of Geo-Information*, *4*(1), 262-301. https://doi.org/10.3390/ijgi4010262

See, L., Mooney, P., Foody, G., Bastin, L., Comber, A., Estima, J., Fritz, S., Kerle, N., Jiang, B., Laakso, M., Liu, H.-Y., Milčinski, G., Nikšič, M., Painho, M., Pődör, A., Olteanu-Raimond, A.-M., & Rutzinger, M. (2016). Crowdsourcing, Citizen Science or Volunteered Geographic Information? The Current State of Crowdsourced Geographic Information. *ISPRS International Journal of Geo-Information*, *5*(5), 55. https://doi.org/10.3390/ijgi5050055

Senaratne, H., Mobasheri, A., Ali, A. L., Capineri, C., & Haklay, M. (Muki). (2017). A review of volunteered geographic information quality assessment methods. *International Journal of Geographical Information Science*, *31*(1), 139-167. https://doi.org/10.1080/13658816.2016.1189556

Stober, T., & Hansmann, U. (2010). Agile software development: Best practices for large software development projects. Springer. https://doi.org/10.1007/978-3-540-70832-2

Tandel, S. S. y Jamadar, A. (2018). A Review on Features, Advantages of Progressive Web Apps (PWA). *International Journal of Innovative Research in Computer and Communication Engineering*, *6*(2), 1091-1095.

Trottier, D. (2017). Digital vigilantism as weaponisation of visibility. *Philosophy & Technology*, *30*, 55–72. https://doi.org/10.1007/s13347-016-0216-4

Zhou, J., Jin, X., Yu, L., Xue, L., & Ren, Y. (2021). TruthTrust: Truth Inference-Based Trust Management Mechanism on a Crowdsourcing Platform. *Sensors*, *21*(8), 2578. https://doi.org/10.3390/s21082578