



Propuesta de una arquitectura limpia de un sistema basado en crowdsourcing para la búsqueda de personas desaparecidas en México

Proposal of a clean architecture for a crowdsourcing-based system for missing persons search in Mexico

Eduardo Vasquez Conde ✉

Tecnológico Nacional de México, Apizaco, México
m19371076@apizaco.tecnm.mx
ORCID: 0009-0000-9239-7946

Jimmy Vazquez Vasquez

Tecnológico Nacional de México, Apizaco, México
m19371082@apizaco.tecnm.mx
ORCID: 0009-0003-4646-1769

Juan Ramos Ramos

Tecnológico Nacional de México, Apizaco, México
juan.rr@apizaco.tecnm.mx
ORCID: 0009-0004-7440-7257

José Juan Hernández Mora

Tecnológico Nacional de México, Apizaco, México
juan.hm@apizaco.tecnm.mx
ORCID: 0000-0003-2878-7290

María Guadalupe Medina Barrera

Tecnológico Nacional de México, Apizaco, México
guadalupe.mb@apizaco.tecnm.mx
ORCID: 0000-0003-3074-0029



<https://doi.org/10.36825/RITI.14.35.005>

Recibido: Junio 14, 2026

Aceptado: Octubre 02, 2026

Resumen: Los sistemas de *crowdsourcing* aplicados a la búsqueda de personas no localizadas enfrentan exigencias importantes en mantenibilidad, rendimiento y seguridad, especialmente cuando se requiere validación humana de los reportes para mitigar información falsa. Debido a que el sistema aún no ha sido implementado ni sometido a pruebas de estrés en un entorno real, el artículo se limita a una propuesta metodológica de diseño y evaluación arquitectónica previa a su implementación, para gestionar reportes ciudadanos de personas desaparecidas usando georreferenciación. El proceso comienza con tres propuestas arquitectónicas, las cuales se comparan mediante la metodología ATAM (*Architecture Tradeoff Analysis Method*) [1], con el único objetivo de identificar la más adecuada. La evaluación se fundamenta en un árbol de utilidad construido a partir de atributos de calidad priorizados por los *stakeholders* del proyecto. Los resultados indican que *Clean Architecture* ofrece un

mejor aislamiento del dominio y facilita de manera notable la realización de pruebas unitarias y de integración, a cambio presenta una mayor complejidad inicial en la definición de puertos y adaptadores. Además, el trabajo aporta una metodología de evaluación arquitectónica en sistemas de impacto social con limitaciones de equipo reducido y requisitos de validación humana.

Palabras clave: *Arquitectura Limpia, Crowdsourcing, Personas No Localizadas, Evaluación Arquitectónica, ATAM.*

Abstract: Crowdsourcing systems applied to the search for missing persons face significant challenges in terms of maintainability, performance, and security, particularly when human validation of reports is required to mitigate false information. Since the system has not yet been implemented or subjected to stress testing in a real-world environment, this article is limited to a methodological proposal for architectural design and evaluation prior to implementation, aimed at managing citizen reports of missing persons using geolocation. The process begins with three architectural proposals, which are compared using the Architecture Tradeoff Analysis Method (ATAM) [1], with the sole objective of identifying the most suitable alternative. The evaluation is based on a utility tree constructed from quality attributes prioritized by the project stakeholders. The results indicate that Clean Architecture provides better domain isolation and significantly facilitates unit and integration testing; however, it introduces greater initial complexity in the definition of ports and adapters. In addition, this work contributes an architectural evaluation methodology for social-impact systems developed under small-team constraints and requiring human validation.

Keywords: *Clean Architecture, Crowdsourcing, Missing Persons, Architectural Evaluation, ATAM.*

1. Introducción

La desaparición de personas es uno de los problemas más graves que enfrentan las sociedades actuales. El tiempo que transcurre entre el reporte inicial y la localización de la persona es un factor determinante para el éxito de las operaciones de búsqueda [2]. En México, los medios tradicionales de denuncia y difusión de alertas, principalmente a través de redes sociales y canales institucionales, presentan limitaciones muy importantes: generan un gran número de información no verificada, carecen de mecanismos de validación y dificultan la georreferenciación precisa de los avistamientos reportados por la ciudadanía.

Esta situación pone en evidencia la necesidad de diseñar sistemas que aprovechen el potencial del *crowdsourcing* como forma de participación ciudadana masiva, pero que al mismo tiempo incorporen controles de calidad capaces de reducir la propagación de información falsa. El desarrollo de este tipo de sistemas plantea desafíos arquitectónicos significativos. Por un lado, se requiere rendimiento y seguridad debido a la sensibilidad de los datos involucrados y a la necesidad de crecer el sistema de forma controlada. Por otro lado, la presencia de una lógica de negocio compleja como la gestión de reportes ciudadanos y consultas geoespaciales exige una clara separación entre el dominio y las tecnologías de infraestructura.

El trabajo en este artículo es una propuesta metodológica de diseño y evaluación arquitectónica previa a la implementación de producción. De ahí que, se presentan tres propuestas de arquitectura, así como su respectiva evaluación y documentación mediante la aplicación del modelo de vistas 4+1 [3] y los principios de *Clean Architecture* [4]. A través de la comparación de las tres alternativas: Monolito modular, Microservicios y *Clean Architecture*, se analiza el balance entre atributos de calidad como mantenibilidad, usabilidad, rendimiento y seguridad. Con la arquitectura seleccionada se busca garantizar el aislamiento del núcleo de negocio, facilitar la ejecución de pruebas unitarias y permitir la evolución del sistema.

El aporte principal de este trabajo está en la aplicación de métodos de documentación y evaluación arquitectónica a un dominio de alto impacto social, donde la integridad de la información y la capacidad de respuesta ante escenarios de crisis son requisitos principales.

2. Estado del arte

2.1. Sistemas de crowdsourcing en contextos de crisis

El uso de enfoques colaborativos basados en la participación ciudadana ha tomado presencia en la gestión de crisis y emergencias durante la última década. Diversos autores han explorado el paradigma del “humano como sensor” (*Human as a Sensor*), donde los dispositivos móviles de los ciudadanos actúan como nodos distribuidos para la captura de información georreferenciada en tiempo real [5], [6]. Este enfoque, conocido como *Mobile Crowd Sensing* (MCS) ha confirmado ser útil en dominios como transporte inteligente, monitoreo ambiental y respuesta a desastres.

Sin embargo, la mayoría de estas investigaciones se han centrado principalmente en aspectos de usabilidad y motivación de los participantes, sin entrar en detalles sobre las decisiones arquitectónicas que respaldan la mantenibilidad, la seguridad y la escalabilidad de estos sistemas a largo plazo [5], [7]. En el contexto específico de la búsqueda de personas desaparecidas, se han desarrollado propuestas que van desde sistemas gubernamentales centralizados hasta plataformas colaborativas. No obstante, existen limitaciones importantes relacionadas con la calidad de la información, la falta de filtros robustos de validación humana y la insuficiente documentación de las arquitecturas que soporten estos procesos [2], [8].

Aunque existen propuestas de sistemas colaborativos para la atención de problemáticas como la desaparición de personas y el registro de eventos delictivos como el aplicativo desarrollado por Fajardo Toloza [9] en Bogotá, la plataforma Vox de Knoop Silveira *et al.* [10] en Brasil y la solución móvil con GPS de López Vázquez [11] en México, la mayoría de estos trabajos priorizan la funcionalidad y la usabilidad por encima de un diseño arquitectónico bien estructurado. Sobre todo, se observa una insuficiente documentación de procesos formales de evaluación arquitectónica que permitan justificar las decisiones de diseño frente a atributos de calidad relevantes en sistemas que manejan información sensible.

2.2. Evaluación arquitectónica y métodos de análisis de trade-offs

Las decisiones arquitectónicas tomadas en etapas tempranas del desarrollo tienen un gran impacto en la capacidad de un sistema para cumplir con sus atributos de calidad a lo largo de su ciclo de vida [12]. En este sentido, se han propuesto diversos métodos para evaluar arquitecturas de software de manera estructurada. Uno de los enfoques más reconocidos es el método ATAM (*Architecture Tradeoff Analysis Method*), desarrollado por Kazman *et al.* [1], el cual permite identificar compromisos (*trade-offs*) entre atributos de calidad mediante la construcción de escenarios de utilidad priorizados.

A pesar de su utilidad demostrada en diversos dominios, la aplicación del método ATAM en sistemas de *crowdsourcing* orientados a crisis humanitarias sigue siendo limitada en la literatura. La mayoría de los trabajos en este ámbito se enfocan en la funcionalidad y la experiencia de usuario, sin realizar un análisis profundo de las consecuencias que las decisiones arquitectónicas tienen sobre atributos como la mantenibilidad, la testabilidad y la capacidad de evolución del sistema [1], [12].

2.3. Patrones arquitectónicos en sistemas con datos sensibles

En sistemas que manejan información sensible y que requieren altos niveles de mantenibilidad, han surgido enfoques arquitectónicos que promueven una separación entre el núcleo del negocio y las preocupaciones técnicas. La Arquitectura Limpia (*Clean Architecture*), propuesta por Martin [4], y la Arquitectura Hexagonal, desarrollada por Cockburn y Garrido de Paz [13], ambas proponen el aislamiento del dominio mediante la inversión de dependencias, lo que permite probar el sistema con mayor facilidad y la evolucionar cada capa de manera independiente.

Algunos estudios comparativos han mostrado que, en sistemas de tamaño moderado, una arquitectura monolítica modular puede resultar más eficiente en términos de rendimiento y más sencilla de operar que una basada en microservicios [14]. Sin embargo, cuando el sistema requiere alta mantenibilidad, facilidad de prueba y la capacidad de cambiar componentes de infraestructura sin tocar la lógica de negocio como ocurre en sistemas que gestionan datos de personas desaparecidas, los enfoques que priorizan el aislamiento del dominio se vuelven más adecuados [4], [15]. Esta separación resulta importante en contextos donde la integridad y la privacidad de la información son requisitos críticos.

Por su parte, Cambarieri *et al.* [16] comprobaron la viabilidad de transformar una arquitectura tradicional en capas hacia una arquitectura hexagonal combinada con Diseño Dirigido por el Dominio (DDD). Su trabajo muestra que al colocar el modelo de dominio como núcleo central y aplicar el principio de inversión de dependencias, es

factible lograr un alto grado de independencia tecnológica y facilitar la mantenibilidad a largo plazo, especialmente en sistemas donde la lógica de negocio debe evolucionar de forma aislada de los detalles técnicos.

2.4. Brecha de investigación

A modo de síntesis, la Tabla 1 presenta una comparación sistemática de los principales trabajos relacionados en la literatura reciente. Esta comparativa examina las metodologías utilizadas por los autores para evaluar sus diseños y los atributos de calidad en los que enfocaron su arquitectura.

Tabla 1. Comparativa sistemática de metodologías de evaluación y atributos de calidad en trabajos relacionados.

Autor (Año)	Dominio de aplicación	Enfoque del diseño arquitectónico	Metodología de evaluación aplicada	Atributos de calidad priorizados
Fajardo Toloza (2023) [9]	Registro de eventos criminales	No documentado formalmente	Pruebas de funcionalidad de software	Usabilidad y funcionalidad
Knoop Silveira et al. (2016) [10]	Búsqueda de personas desaparecidas	Centralizado (Cliente-Servidor)	Evaluación heurística y de experiencia de usuario.	Usabilidad y accesibilidad.
López Vázquez [11] (2022)	Prevención y rastreo GPS	Aplicación móvil acoplada a hardware	Pruebas de campo (Hardware/Software)	Funcionalidad
Blinowski et al. (2022) [14]	Sistemas empresariales	Monolito vs. Microservicios	Benchmarking y pruebas de estrés.	Rendimiento y escalabilidad
Cambarieri et al. (2020) [16]	Gestión empresarial	Arquitectura Hexagonal y Domain-Drive Design	Análisis cualitativo de dependencias.	Mantenibilidad e independencia tecnológica.

Fuente: Elaboración propia.

Como se puede observar, aunque los sistemas orientados a crisis priorizan metodologías de evaluación basadas en usabilidad y pruebas funcionales, no tienen evaluaciones arquitectónicas estructuradas. Por otro lado, los trabajos que sí aplican métodos de evaluación arquitectónica no se enfocan en ecosistemas de *crowdsourcing* social.

La revisión de la literatura y la comparativa anterior revelan avances significativos en el desarrollo de sistemas colaborativos para la gestión de crisis y en los métodos de evaluación arquitectónica [5], [12]. Sin embargo, permanece una brecha importante: existe poca documentación de procesos completos de evaluación que armonicen un análisis estructurado de *trade-offs* como ATAM con diseños de aislamiento orientados al *crowdsourcing* de impacto social.

3. Materiales y métodos

La metodología de evaluación arquitectónica se fundamentó principalmente en el método ATAM y de forma complementaria con un análisis inicial de viabilidad a través de un *Business Case* (caso de negocio) adaptado de prácticas de gestión de proyectos [17]. Este enfoque unificado permitió, por un lado, permitió filtrar rápidamente alternativas inviables, y por otro, realizar una evaluación técnica completa de los *trade-offs* entre atributos de calidad mediante escenarios ATAM [1].

Para dar validez al proceso de ponderación de los atributos de calidad y la construcción del árbol de utilidad, la evaluación ATAM se estructuró definiendo formalmente a los grupos requeridos por el método. El equipo evaluador estuvo conformado por docentes externos al equipo de desarrollo, quienes aportaron su experiencia y rigor metodológico fungiendo como los principales responsables de la evaluación arquitectónica. Complementariamente, para representar las necesidades reales del dominio, se conformó un panel de *stakeholders* combinando la perspectiva técnica del equipo de arquitectura con las necesidades reales del dominio. Para capturar estas últimas, se diseñó y aplicó un instrumento de consulta dirigido a usuarios clave. El panel estuvo conformado por cuatro perfiles estratégicos: (1) un miembro de una organización formal de búsqueda, (2) un familiar directo

de una persona no localizada, y (3) dos ciudadanos de la sociedad civil (uno con relación indirecta a la problemática).

Mediante el análisis de esta consulta, se derivaron las ponderaciones cualitativas para los escenarios críticos. Como evidencia, los *stakeholders* calificaron con valores altos (9-10/10) la necesidad de proteger los datos sensibles y validar la identidad de colaboradores. Este resultado justificó asignar una prioridad alta a los escenarios de seguridad. Asimismo, la preocupación mayoritaria de los usuarios frente a procesos de reporte largos o complicados justificó la ponderación restrictiva en los escenarios de Usabilidad.

En la primera fase se aplicó una metodología con base en los primeros 4 pasos del *Business Case* [17]: (1) identificación de la necesidad, (2) definición de objetivos SMART, (3) identificación de alternativas, (4) recopilación de datos técnicos y financieros.

Posteriormente, se utilizó un enfoque de Análisis de Costo-Efectividad (ACE) orientado a programas sociales. Bajo este paradigma, la viabilidad técnica evaluó la factibilidad de implementación según los recursos del equipo; después se realizó, la evaluación de costo-efectividad donde se contrastaron los costos de infraestructura y de complejidad arquitectónica frente a indicadores de valor público no monetarios. El objetivo principal de este análisis fue determinar qué arquitectura maximiza la generación de confianza en la sociedad y la facilidad de uso para fomentar la participación ciudadana continua, asegurando al mismo tiempo la resiliencia del sistema.

En una segunda fase, se propusieron diferentes alternativas de solución y como siguiente paso se procedió a su evaluación: tres alternativas propuestas fueron: Monolito Modular con API REST, Arquitectura de Microservicios y *Clean Architecture* con elementos de arquitectura hexagonal. Donde la evaluación se centró en los siguientes cuatro atributos prioritarios derivados del consenso con los *stakeholders*:

1. Rendimiento y fiabilidad bajo concurrencia.
2. Mantenibilidad y adecuación funcional.
3. Seguridad y tratamiento de datos sensibles.
4. Usabilidad y portabilidad.

La ponderación de estos atributos de calidad sirvió como la guía objetiva, para evaluar los compromisos técnicos (*trade-offs*) entre los distintos patrones arquitectónicos. Así permitiendo descartar aquellas que no logran la satisfacción total de los atributos de calidad mencionados.

En una tercera fase, la arquitectura seleccionada se documentó a detalle utilizando como base el modelo de vistas 4+1 de Kruchten: vista lógica (diagrama de clases UML), vista de procesos (diagramas de flujo de casos de uso principales), vista de desarrollo (diagrama de componentes), vista física (diagrama de despliegue) y escenarios (+1) [3].

4. Resultados

4.1. Evaluación arquitectónica mediante ATAM

La selección de la arquitectura para el núcleo de reportes se respaldó con el Método de Análisis de Compensaciones Arquitectónicas (ATAM) [1]. Las restricciones de tiempo y presupuesto del proyecto fueron debidamente identificadas preliminarmente mediante un *Business Case*, se usó exclusivamente como identificador de negocio para deducir los cuatro atributos de calidad críticos: seguridad, usabilidad, accesibilidad y rendimiento.

Para plasmar estos atributos en requisitos evaluables, se construyó un Árbol de Utilidad (véase Figura 1). El análisis técnico priorizó los escenarios con clasificación (A, A) Alta importancia para los *stakeholders* y Alta dificultad arquitectónica, utilizándolos como funciones restrictivas para evaluar tres alternativas: Monolito Modular, Arquitectura de Microservicios y *Clean Architecture* [4] [14].

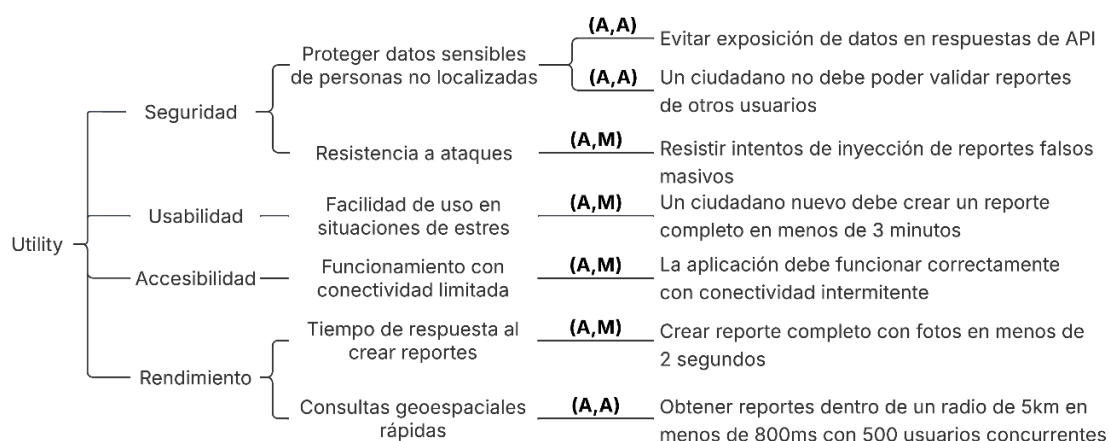


Figura 1. Árbol de utilidad. Fuente: Elaboración propia.

Los resultados de la evaluación técnica frente a los escenarios críticos demostraron los siguientes compromisos:

- Seguridad y Aislamiento (Escenarios A,A): El escenario de "Evitar exposición de datos en respuestas de API" descartó al Monolito Modular. Dicha arquitectura presentó un alto riesgo de acoplamiento, donde las entidades de base de datos podrían filtrarse a la capa de presentación. En contraste, *Clean Architecture* mitigó esta vulnerabilidad mediante el uso estricto de Objetos de Transferencia de Datos (DTOs) y dependencias concéntricas, garantizando el aislamiento de la información de las personas no localizadas.
- Rendimiento Geoespacial (Escenario A,A): Frente al requerimiento de "Obtener reportes en un radio de 5km en menos de 800ms", tanto Microservicios como *Clean Architecture* demostraron viabilidad técnica. Sin embargo, el patrón de puertos y adaptadores de *Clean Architecture* permitió abstraer el motor de base de datos geoespacial como un *plugin* externo, sin tocar la capa de negocio del núcleo.
- Puntos de Sensibilidad en contraste con la Viabilidad Operativa: Aunque la Arquitectura de Microservicios tiene un rendimiento superior en la concurrencia extrema, el análisis determinó que su costo de infraestructura y complejidad de orquestación inicial rompían las restricciones del *Business Case*. Por su parte, el punto de sensibilidad de *Clean Architecture* radicó en una latencia óptima inducida por la transformación de datos entre capas (afectando levemente el escenario A,M de tiempo de respuesta).
- Viabilidad técnica y costo-efectividad social: El análisis demostró que, bajo el enfoque de costo-efectividad, la Arquitectura de Microservicios tiene un costo operativo y de infraestructura desproporcionado frente al beneficio esperado en esta fase del proyecto. Aunque el Monolito Modular presenta el menor costo técnico inicial, su acoplamiento estructural genera un riesgo de inestabilidad y vulnerabilidad de datos; un fallo de esta naturaleza disminuiría la confianza de la sociedad, anulando la efectividad del sistema. Como resultado del paso 4, *Clean Architecture* representó la mejor relación costo-efectividad social: si bien requiere una inversión técnica mayor en su estructuración inicial (costo), este diseño protege el dominio y la información sensible, por lo tanto, es el facilitador principal para crear fiabilidad hacia la ciudadanía (efectividad), permitiendo además aislar la capa de presentación para maximizar la facilidad de uso al reportar.

4.2. Análisis de atributos de calidad mediante ATAM y evaluación de trade-offs

Para cuantificar la viabilidad de las alternativas arquitectónicas frente a los requerimientos del núcleo de reportes, se evaluó el nivel de cumplimiento de los cuatro atributos de calidad principales. Para ello, se definió una escala numérica del 1 al 4 (1 = Bajo/Insatisfactorio, 2 = Medio, 3 = Alto, 4 = Muy Alto). Esta ponderación se derivó directamente del nivel de satisfacción de los escenarios críticos (clasificados como Alta importancia / Alta dificultad [A, A]) en el Árbol de Utilidad. Un valor de 4 indica que la arquitectura resuelve el escenario restrictivo sin degradar otros atributos; un 3 refleja un cumplimiento aceptable con compensaciones manejables; un 2 representa un cumplimiento parcial; y un 1 indica una incapacidad arquitectónica o alto riesgo para satisfacer el requerimiento. La Tabla 2 muestra la evaluación, utilizando el nivel de satisfacción de cada escenario del Árbol de Utilidad.

Tabla 2. Evaluación de atributos de calidad por patrón arquitectónico.

Atributo de Calidad	Monolito Modular	Microservicios	Clean Architecture
Seguridad de datos	1 - Bajo (Acoplamiento)	3 - Alto	4 - Muy Alto (Aislamiento)
Rendimiento geoespacial	2 - Medio	4 - Muy Alto	3 - Alto
Mantenibilidad	2 - Medio	1 - Bajo (Complejo)	4 - Muy Alto
Usabilidad	3 - Alto	2 - Medio	2 - Medio

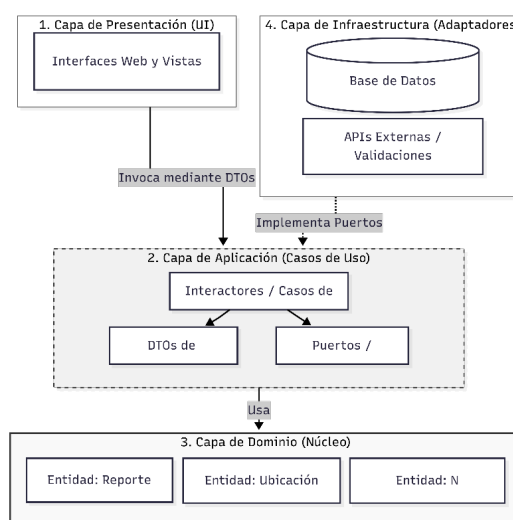
Fuente: Elaboración propia.

El análisis de resultados evidenció compromisos técnicos (*trade-offs*) determinantes que fundamentaron la selección definitiva de *Clean Architecture*:

- Seguridad frente a Simplicidad de Desarrollo: El Monolito Modular facilita una rápida implementación inicial pero su falta de fronteras estrictas entre capas representa un riesgo inaceptable de exposición de datos sensibles. *Clean Architecture* toma el compromiso técnico de requerir mayor escritura de código estructural y una curva de aprendizaje más pronunciada a cambio de garantizar el encapsulamiento total de las reglas de negocio y la información de los reportes.
- Mantenibilidad Operativa frente a Escalabilidad Extrema: Aunque la Arquitectura de Microservicios tiene el nivel óptimo de rendimiento bajo concurrencia masiva, el uso en la fase actual del proyecto frena abruptamente la mantenibilidad debido a la complejidad de orquestación. *Clean Architecture* actúa como un punto de equilibrio ya que impone una separación lógica estricta que facilita la escalabilidad y una migración natural a microservicios en el futuro, sin llegar a costos de infraestructura considerablemente altos.
- Aislamiento Tecnológico frente a Latencia: Para lograr las consultas geoespaciales de alta demanda (escenario A,A), *Clean Architecture* permite la integración de motores especializados como lo son las bases de datos documentales a través del patrón de puertos y adaptadores. El compromiso técnico radica en una latencia de procesamiento ocasionada por el mapeo continuo de datos conversión de objetos de transferencia de datos (*DTOs*) a entidades de dominio entre las distintas capas, un costo operacional considerado aceptable frente a la resiliencia y seguridad obtenidas en el núcleo del sistema.

4.3. Documentación de la arquitectura seleccionada

Para documentar el diseño resultante de la evaluación ATAM, se utilizó la documentación el modelo de vistas 4+1 de Kruchten [3]. La Figura 2 ilustra la topología del sistema, estructurada bajo los principios de *Clean Architecture*.

**Figura 2.** Arquitectura del sistema que apunta al dominio para garantizar la integridad y estabilidad.

Fuente: Elaboración propia.

El diseño se centra en seguir la regla de dependencia concéntrica así se logra que el flujo de control se dirija hacia las reglas de negocio de esta manera se aísla el núcleo de reportes mediante cuatro capas principales:

- **Capa de Dominio (Núcleo Central):** Aloja las entidades puras y agnósticas del dominio de personas desaparecidas (ej. Reporte, Ciudadano, Ubicacion). Al no tener dependencias tecnológicas externas se logra garantizar la integridad y estabilidad de las reglas de negocio críticas.
- **Capa de Aplicación (Casos de Uso):** Funge el rol de orquestador del sistema el cual define los puertos (interfaces abstractas) para la persistencia y notificaciones. Todo el intercambio de información con el exterior se realiza mediante Objetos de Transferencia de Datos (DTOs), disminuyendo el riesgo latente de exponer información ciudadana sensible.
- **Capa de Infraestructura (Adaptadores):** Implementa las interfaces definidas por la capa de aplicación debido a esta separación logra que las operaciones complejas, como son las consultas al motor de base de datos geoespacial puedan funcionar como *plugins* intercambiables.
- **Capa de Presentación:** Interfaz de usuario desarrollada con componentes reactivos los cuales se logran comunicar con el sistema mediante invocaciones de los casos de uso de la capa de aplicación lo que en esencia protege a las entidades centrales de posibles vulnerabilidades o inyecciones desde el cliente web.

4.4. Validación empírica preliminar

Para dar mayor valor cuantitativo a las situaciones críticas identificadas en el método ATAM, se desarrolló un prototipo mínimo viable (MVP) del núcleo de reportes utilizando el *framework* .NET 8 (C#) mediante una arquitectura de Web API. El diseño metodológico de esta validación se sustenta en los principios de testabilidad de la Arquitectura Limpia [4] y el patrón de puertos y adaptadores [13].

De acuerdo con la literatura, la verificación formal de un diseño desacoplado requiere que las reglas de negocio sean puestas a prueba aislando por completo la infraestructura física mediante el uso de objetos simulados (mocks). Asimismo, para evaluar el rendimiento de la arquitectura frente a la concurrencia [12], se emuló la latencia del motor de base de datos. Tomando como base este fundamento teórico, la validación empírica se dividió en dos enfoques complementarios:

- **Validación de arquitectura y aislamiento mediante pruebas con xUnit:** Se ejecutaron pruebas unitarias siguiendo el patrón arquitectónico xUnit para aprobar el escenario de seguridad: "Evitar exposición de datos en respuestas de API". Mediante el uso de mocks en la capa de adaptadores, se comprobó que las entidades del dominio no se filtran hacia la capa de presentación. La ejecución de las pruebas demostró exitosamente el correcto aislamiento de los datos sensibles mediante el uso de DTOs, sin depender de motores de base de datos reales.
- **Validación de rendimiento geoespacial mediante pruebas de carga con k6:** Para verificar el escenario de "Obtener reportes dentro de un radio de 5km en menos de 800ms con 500 usuarios concurrentes", se configuró un testbed utilizando la herramienta de código abierto k6. Se inyectó tráfico concurrente emulando 13,572 peticiones geoespaciales aleatorias durante 30 segundos. Como se detalla en la Tabla 3, los resultados de la telemetría demostraron que el sistema procesó las solicitudes con un tiempo de respuesta promedio de 117.53 ms y un percentil 95 (P95) de 176.75 ms, manteniéndose por debajo del umbral de los 800ms determinado en el árbol de utilidad.

Tabla 3. Resultados de la prueba de carga (k6) para consultas geoespaciales concurrentes.

Métrica evaluada	Resultado obtenido	Umbral de aceptación (ATAM)	Estado
Usuarios concurrentes (VUs)	500 usuarios	500 usuarios	Satisfactorio
Peticiones procesadas	13,572 peticiones	<800 ms	Satisfactorio
Tiempo de respuesta promedio (Avg)	117.53 ms	<800 ms	Satisfactorio
Percentil 95 (P95)	176.75 ms	<800 ms	Satisfactorio

Fuente: Elaboración propia.

Esta evidencia cuantitativa preliminar confirma que los compromisos técnicos (*trade-offs*) asumidos por *Clean Architecture* especialmente la latencia por la conversión de DTOs son aceptables y no impiden el cumplimiento de los atributos de calidad bajo estrés operativo.

5. Conclusiones

El desarrollo de sistemas que se basan en el uso de *crowdsourcing* para una gestión aplicada a la búsqueda de personas desaparecidas necesita de un diseño que garantice la integridad de los datos sensibles sin comprometer la capacidad de respuesta del sistema. Este trabajo sustenta que la selección arquitectónica en dominios de alto impacto social debe basarse en una evaluación estructurada de los atributos de calidad críticos. Mediante la aplicación del método ATAM, y proponiendo una metodología en base a la planteada por IT Madrid [17], remplazando las variables de rentabilidad por un Análisis de Costo-Efectividad (ACE) para priorizar el impacto social y el desarrollo de un prototipo mínimo viable (MVP), se determinó que *Clean Architecture* representa la propuesta metodológica más sólida para establecer el núcleo principal de reportes previo a su implementación.

El análisis de compromisos técnicos resalta que el aislamiento del dominio es un requisito no negociable. El uso del patrón de puertos y adaptadores permite proteger la aplicación contra vulnerabilidades de acoplamiento, protegiendo la información ciudadana y facilitando la integración de motores de persistencia geoespacial como *plugins* externos.

Dado que este estudio se limita a la propuesta de diseño y evaluación arquitectónica, las principales líneas de trabajo futuro se centrarán en el despliegue del sistema y la ejecución de pruebas de estrés operativas. Esto incluirá la implementación de pruebas unitarias, de carga y de accesibilidad en entornos de producción reales para contrastar el rendimiento empírico del sistema frente a los tiempos de respuesta teóricos proyectados en los escenarios críticos del Árbol de Utilidad. Posteriormente, se plantea la integración progresiva de módulos periféricos de validación institucional, monitoreando el acoplamiento arquitectónico para asegurar que el sistema mantenga su resiliencia técnica a medida que escale su despliegue operativo.

6. Agradecimientos

Agradecemos a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) por el apoyo financiero brindado a través de la beca de posgrado, fundamental para el desarrollo de esta investigación. Asimismo, se extiende un sincero reconocimiento a los docentes del Tecnológico Nacional de México/ITApizaco por la orientación técnica y metodológica durante este proyecto.

7. Referencias

- [1] Kazman, R., Klein, M. H., Clements, P. (2000). *ATAM: Method for architecture evaluation* (CMU/SEI-2000-TR-004). Software Engineering Institute, Carnegie Mellon University. <https://www.sei.cmu.edu/library/atam-method-for-architecture-evaluation/>
- [2] Vélez Salas, D. M., Vélez Salas, M. A., Ambrosio Jaramillo, L. J., Guerrero Rojas, N. O., Martínez Medrano, C. I. (2020). *¿Sabemos cuántas personas desaparecidas hay en México? Nota técnica del Registro Nacional de Personas Desaparecidas y No Localizadas*. Observatorio Ciudadano de Seguridad, Justicia y Legalidad. https://onc.org.mx/public/onc_site/uploads/NotaTec-desaparecidos.pdf
- [3] Kruchten, P. B. (1995). The “4+1” view model of software architecture. *IEEE Software*, 12 (6), 42–50. <https://doi.org/10.1109/52.469759>
- [4] Martin, R. C. (2017). *Clean architecture: A craftsman's guide to software structure and design*. Pearson.
- [5] Avvenuti, M., Cimino, M. G. C. A., Cresci, S., Marchetti, A., Tesconi, M. (2016). A framework for detecting unfolding emergencies using humans as sensors. *SpringerPlus*, 5 (43), 1-23. <https://link.springer.com/article/10.1186/s40064-016-1674-y>
- [6] Zhang, C., Zhu, L., Xu, C., Ni, J., Huang, C., Shen, X. (2022). Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing. *IEEE Transactions on Mobile Computing*, 21 (12), 4410–4425. <https://doi.org/10.1109/TMC.2021.3080714>

- [7] Guo, B., Chen, H., Yu, Z., Xie, X., Huangfu, S., Wang, Z. (2014). *FlierMeet: Cross-space public information reposting with mobile crowd sensing*. ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Seattle, Washington. <https://doi.org/10.1145/2638728.2638730>
- [8] Zambrano Vizuite, A. M., Ortiz Mejía, E. L., Zambrano Vizuite, O. M., Calderón Hinojosa, X. A. (2020). An agile real-time location system (RTLS) for missing persons using IoT technology. *Computación y Sistemas*, 24 (2), 845–860. <https://doi.org/10.13053/CyS-24-2-3153>
- [9] Fajardo Toloza, L. S. (2023). *Diseño de un aplicativo crowdsourcing para el registro de eventos criminales* [Trabajo de Grado]. Universidad Distrital Francisco José de Caldas, Bogotá D. C., Colombia. <http://hdl.handle.net/11349/39589>
- [10] Knoop Silveira, P. F., Remor Oliveira, R., Ruas Oliveira, R. (2016). Uma tecnologia social baseada em crowdsourcing para busca de pessoas desaparecidas e cadastro de moradores de rua. En *Anais do XIII Simpósio Brasileiro de Sistemas Colaborativos (SBSC 2016)* (pp. 1437–1448). Sociedade Brasileira de Computação. <https://doi.org/10.5753/sbsc.2016.9501>
- [11] López Vázquez, D. G. (2022). *Aplicación móvil y periférico GPS como herramienta en la prevención de la desaparición involuntaria de personas en México* [Tesis de Licenciatura]. Benemérita Universidad Autónoma de Puebla. México. <https://hdl.handle.net/20.500.12371/17179>
- [12] Bass, L., Clements, P., Kazman, R. (2022). *Software architecture in practice* (4th Ed.). Addison-Wesley Professional.
- [13] Cockburn, A., Garrido de Paz, J. M. (2025). *Hexagonal architecture explained: How the ports & adapters architecture simplifies your life, and how to implement it* (1st Ed.). Humans and Technology Incorporated.
- [14] Blinowski, G., Ojdowska, A., Przybyłek, A. (2022). Monolithic vs. microservice architecture: A performance and scalability evaluation. *IEEE Access*, 10, 20357–20374. <https://doi.org/10.1109/ACCESS.2022.3152803>
- [15] Evans, E. (2003). *Domain-driven design: Tackling complexity in the heart of software*. Addison-Wesley Professional.
- [16] Cambarieri, M., Difabio, F., García Martínez, N. (2020). Implementación de una arquitectura de software guiada por el dominio. En *Actas de las 49 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO), XXI Simposio Argentino de Ingeniería de Software (ASSE 2020)* (pp. 192–209). Sociedad Argentina de Informática e Investigación Operativa. <http://sedici.unlp.edu.ar/handle/10915/115198>
- [17] ITMadrid IT Business School. (2017). *9 pasos para crear un Business Case de éxito*. <https://www.itmadrid.com/wp-content/uploads/2021/03/itmadridd-9pasos-para-un-business-case-de-exito-3.pdf>