

# **Análisis Exhaustivo, Diseño Arquitectónico e Implementación de Ecosistemas de Salud Digital para Geriatría mediante MoAI-ADK**

## **1. Contexto Sociodemográfico y la Necesidad de la Medicina de Precisión Asistida**

La intersección entre la longevidad poblacional y la prevalencia de enfermedades crónico-degenerativas ha generado una crisis silenciosa en los sistemas de salud modernos. En el contexto específico de México y América Latina, la transición epidemiológica ha desplazado a las enfermedades infecciosas, colocando a la diabetes mellitus tipo 2 y la hipertensión arterial sistémica como las principales causas de morbilidad y mortalidad.<sup>1</sup> Este fenómeno no es meramente estadístico; representa una carga operativa inmensa para los cuidadores y el sistema de salud pública, exacerbada por la complejidad inherente al manejo farmacológico en pacientes de la tercera edad. La polifarmacia, común en este grupo demográfico, choca frontalmente con el deterioro cognitivo natural asociado al envejecimiento, resultando en tasas de no adherencia que oscilan peligrosamente, provocando descompensaciones agudas, hospitalizaciones evitables y, en última instancia, un aumento en los costos sanitarios y una disminución en la calidad de vida.

La solicitud de desarrollar una aplicación que no solo gestione recordatorios de medicación, sino que actúe como un nodo central de monitoreo fisiológico y respuesta a emergencias, responde a una necesidad crítica de pasar de una medicina reactiva a una proactiva. Sin embargo, el desarrollo de software médico (Software as a Medical Device - SaMD) conlleva riesgos inherentes que no pueden ser mitigados mediante metodologías de desarrollo ágiles tradicionales o enfoques de "vibe coding" superficiales.<sup>2</sup> La integridad de los datos, la precisión de los algoritmos de detección de anomalías y la fiabilidad de las integraciones con hardware médico exigen un rigor de ingeniería que solo puede ser garantizado mediante marcos de desarrollo agéntico estructurados.

Aquí es donde entra en juego el MoAI-ADK (Agentic Development Kit), un marco de trabajo que redefine la ingeniería de software asistida por inteligencia artificial.<sup>4</sup> A diferencia de los asistentes de codificación convencionales que generan fragmentos de código aislados, MoAI-ADK implementa una metodología "SPEC-First" (Especificación Primero) respaldada por una orquestación de agentes especializados. Este enfoque asegura que cada línea de código escrita para la aplicación —desde la gestión de la pila Bluetooth Low Energy (BLE) hasta los algoritmos de detección de infartos— esté precedida por una especificación inequívoca,

validada por pruebas automatizadas (TDD) y documentada en tiempo real.<sup>4</sup> Para una aplicación destinada a salvar vidas y gestionar datos sensibles bajo normativas como la NOM-024-SSA3-2012 en México, este nivel de rigor no es opcional, es imperativo.<sup>7</sup>

El presente análisis desglosa la arquitectura, el diseño de experiencia de usuario (UX) centrado en geriatría, la integración de hardware IoT y la implementación de protocolos de emergencia, todo bajo la gobernanza del framework MoAI-ADK.

## 2. Metodología de Desarrollo Agentico: El Framework MoAI-ADK

La construcción de sistemas críticos para la seguridad del paciente requiere una trazabilidad absoluta desde el requisito clínico hasta la implementación del código. MoAI-ADK v1.0.0 y versiones superiores ofrecen una infraestructura robusta basada en la orquestación de 26 a 28 agentes de inteligencia artificial especializados, organizados jerárquicamente para simular un equipo de desarrollo de software de alto rendimiento.<sup>4</sup>

### 2.1 Orquestación Jerárquica y Roles de Agentes

El núcleo del sistema es "Mr. Alfred", el orquestador principal que interpreta las intenciones del desarrollador y delega tareas a sub-agentes específicos, asegurando que no haya alucinaciones en la lógica crítica del negocio.<sup>5</sup> Para nuestra aplicación de salud, la estructura operativa se divide en niveles de competencia técnica y de gestión.

Tabla 1: Asignación de Agentes MoAI-ADK para el Desarrollo de la App de Salud

Nivel (Tier)	Agente Especializado	Responsabilidad en el Contexto Médico	Justificación Técnica
Tier 1 (Expertos)	expert-backend	Diseño de esquemas de base de datos PostgreSQL compatibles con HIPAA/NOM-024.	Requiere manejo estricto de integridad referencial y seguridad a nivel de fila (RLS). <sup>10</sup>
	expert-frontend	Implementación de UI de alto contraste	Optimización para usuarios con

		y accesibilidad WCAG 2.1 AA.	presbicia y cataratas. <sup>11</sup>
	expert-security	Auditoría de encriptación y gestión de tokens OAuth2.	Protección contra fugas de datos sensibles (PHI). <sup>4</sup>
<b>Tier 2 (Gestores)</b>	manager-spec	Redacción de especificaciones EARS para lógica de alertas.	Elimina ambigüedad en condiciones críticas (ej. "Si glucosa > 400 mg/dL"). <sup>12</sup>
	manager-tdd	Ciclo Red-Green-Refactor para algoritmos de dosis.	Garantiza que la lógica de medicación no falle matemáticamente. <sup>4</sup>
<b>Tier 4 (Integración)</b>	mcp-integrations	Conexión con SDKs de hardware (BLE) y APIs de telemedicina.	Abstracción de la complejidad de comunicación con dispositivos médicos. <sup>4</sup>

## 2.2 El Ciclo de Vida SPEC-First y Sintaxis EARS

El error más común en el desarrollo de apps de salud es la ambigüedad en los requisitos. MoAI-ADK mitiga esto mediante la sintaxis EARS (Easy Approach to Requirements Syntax), que obliga a definir el comportamiento del sistema bajo condiciones explícitas.<sup>12</sup>

Para el control de la diabetes, un flujo de trabajo típico iniciaría con el comando /moai:1-plan. El agente manager-spec transformaría una solicitud vaga como "alertar si la glucosa es alta" en una especificación técnica rigurosa:

- *Precondición:* Cuando el dispositivo BLE reporta una lectura nueva.
- *Disparador:* Si el valor de glucosa excede el umbral configurado por el médico (ej. 180 mg/dL).
- *Respuesta del Sistema:* La app deberá emitir una alerta sonora de máxima prioridad, enviar un SMS al cuidador y bloquear otras notificaciones no críticas.

Este documento SPEC se convierte en la "fuente única de verdad", evitando que el agente de

desarrollo (manager-tdd) improvise lógica que podría tener consecuencias clínicas fatales.<sup>13</sup>

## 2.3 Garantía de Calidad con Ralph Engine y TRUST 5

La integración del Ralph Engine proporciona un bucle de retroalimentación autónomo que utiliza diagnósticos LSP (Language Server Protocol) y análisis de árbol de sintaxis abstracta (AST-grep).<sup>3</sup> En la práctica, esto significa que si el código generado para la conexión Bluetooth no maneja adecuadamente las excepciones de desconexión —un escenario común y peligroso en monitoreo remoto—, el Ralph Engine detectará la vulnerabilidad y rechazará el código antes de que sea aceptado, forzando una refactorización hasta cumplir con los principios TRUST 5 (Testable, Readable, Unified, Secured, Trackable).<sup>3</sup>

---

## 3. Diseño de Experiencia de Usuario (UX) para la Población Geriátrica

El éxito de una aplicación para el control de enfermedades crónicas en adultos mayores no reside en la complejidad de sus funciones, sino en la accesibilidad de su interfaz. El diseño debe asumir déficits sensoriales y cognitivos como la norma, no como la excepción.<sup>15</sup>

### 3.1 Accesibilidad Visual y Cognitiva: Más allá de lo Estético

La agudeza visual disminuye con la edad, y condiciones como la retinopatía diabética pueden reducir la sensibilidad al contraste. Por lo tanto, el diseño debe adherirse estrictamente a las pautas WCAG 2.1 y 2.2 nivel AA o superior.<sup>11</sup>

- **Tipografía y Legibilidad:** Se debe evitar el uso de fuentes serif o estilizaciones finas. El tamaño base de la fuente no debe ser inferior a 18sp (scale-independent pixels), con capacidad de escalado hasta el 200% sin romper la maquetación.<sup>18</sup>
- **Contraste Cromático:** El ratio de contraste entre texto y fondo debe ser de al menos 7:1, superando el estándar habitual de 4.5:1, para acomodar la pérdida de sensibilidad al contraste.<sup>19</sup> Es crucial no confiar únicamente en el color para transmitir información (ej. un botón rojo de emergencia debe decir también "EMERGENCIA" y tener un icono distintivo) para apoyar a usuarios con daltonismo.<sup>15</sup>
- **Zonas de Interacción (Touch Targets):** Debido a la reducción de la motricidad fina y posibles temblores (Parkinson o debilidad senil), los elementos interactivos deben tener un tamaño mínimo de 44x44 píxeles (preferiblemente 48x48dp en Android), con un espaciado generoso para evitar toques accidentales ("fat finger error").<sup>11</sup>

### 3.2 Interfaz de Voz (VUI) y Retroalimentación Multimodal

La barrera tecnológica más significativa para los adultos mayores es a menudo la entrada de texto. La implementación de una Interfaz de Usuario de Voz (VUI) robusta es fundamental

para la adherencia.<sup>22</sup> Utilizando los agentes mcp-integrations de MoAI-ADK, se pueden orquestar servicios nativos como Android Voice Access o iOS Voice Control para permitir una operación "manos libres" completa.

- **Confirmación de Medicación por Voz:** En lugar de navegar menús, el usuario debería poder decir "Ya tomé mi pastilla" o "¿Qué me toca tomar?". El sistema debe confirmar la acción auditivamente ("Confirmado, tomaste Losartán a las 8:00 AM") para cerrar el bucle de retroalimentación.<sup>21</sup>
- **Alertas Multimodales:** Las notificaciones críticas deben atacar múltiples sentidos simultáneamente: vibración háptica prolongada, señal auditiva de frecuencia baja (más audible para oídos envejecidos que las altas frecuencias) y parpadeo visual en pantalla.<sup>15</sup>

### 3.3 Navegación y Carga Cognitiva

La memoria de trabajo se reduce con la edad, lo que dificulta la navegación en menús profundos o jerárquicos. La arquitectura de la información debe ser plana y lineal.

- **Evitar Gestos Complejos:** Gestos como "deslizar para borrar" o "pellizcar para hacer zoom" son anti-intuitivos y físicamente difíciles para este grupo demográfico. Todas las acciones deben ser visibles a través de botones explícitos con etiquetas de texto claras (ej. "Volver", "Guardar") en lugar de iconos abstractos.<sup>21</sup>
- **Consistencia y Tolerancia al Error:** El sistema debe ser indulgente. Si un usuario marca por error una toma de medicamento, debe haber un mecanismo simple y visible para deshacer la acción ("Deshacer" grande y claro).<sup>25</sup>

---

## 4. Arquitectura Técnica e Integración de Hardware Médico (IoT)

La capacidad de la aplicación para conectarse con dispositivos biométricos transforma el teléfono móvil de un simple recordatorio a un centro de diagnóstico personal. Esta integración se realiza principalmente a través del protocolo Bluetooth Low Energy (BLE) y perfiles GATT (Generic Attribute Profile) estandarizados.

### 4.1 Selección del Framework Móvil: Flutter vs React Native

MoAI-ADK soporta ambos frameworks, pero la elección impacta la facilidad de integración con hardware.<sup>27</sup>

- **Flutter:** Recomendado por su rendimiento nativo y consistencia visual en diferentes dispositivos, crucial para mantener la accesibilidad UI diseñada. La librería flutter\_blue\_plus o flutter\_reactive\_ble ofrece un control granular sobre la conexión BLE, necesario para manejar la reconexión automática con dispositivos médicos que entran y salen del modo de suspensión agresivamente para ahorrar batería.<sup>28</sup>
- **React Native:** Si bien viable con react-native-ble-plx, puede presentar mayores desafíos

de rendimiento en hilos de fondo (background threads) necesarios para el monitoreo continuo de signos vitales sin la interfaz abierta.<sup>31</sup>

Para este proyecto, se sugiere **Flutter** debido a su robustez en el manejo de streams de datos en tiempo real y su capacidad para dibujar interfaces de usuario personalizadas de alto rendimiento, facilitando la implementación de gráficos de tendencias de glucosa fluidos y responsivos.<sup>27</sup>

## 4.2 Protocolos de Comunicación GATT y Decodificación de Datos

La comunicación con glucómetros y tensiómetros no es trivial; requiere la implementación exacta de los perfiles GATT definidos por el Bluetooth SIG.

**Tabla 2: Perfiles GATT Críticos para Monitoreo de Enfermedades Crónicas**

Dispositivo	Servicio UUID	Característica Principal	Perfil Estándar	Desafío de Implementación
Glucómetro	0x1808	Glucose Measurement (0x2A18)	GLS (Glucose Service)	Manejo de "Contexto" (ayuno, postprandial) y seguridad de emparejamiento con PIN. <sup>34</sup>
Tensiómetro	0x1810	Blood Pressure Measurement (0x2A35)	BLS (Blood Pressure Service)	Los datos suelen enviarse como "Indicación" (requiere ACK) en lugar de "Notificación". <sup>36</sup>
Oxímetro / HR	0x180D	Heart Rate Measurement (0x2A37)	HRS (Heart Rate Service)	Decodificación de banderas de 8 vs 16 bits para el valor de ritmo

				cardíaco. <sup>38</sup>
--	--	--	--	-------------------------

El proceso de decodificación de datos es un punto crítico de fallo. Los datos BLE llegan en arrays de bytes (byte arrays), generalmente en formato *little-endian*. El agente expert-backend de MoAI-ADK debe generar parsers que interpreten el primer byte de "Flags" para determinar la presencia de campos opcionales (como timestamp o estado del sensor) antes de leer los valores numéricos.<sup>37</sup>

*Ejemplo Técnico:* Un glucómetro envía una secuencia de bytes. El bit 0 del byte de Flags indica si hay un offset de tiempo. Si se ignora este bit y se lee el offset como parte de la concentración de glucosa, el resultado podría ser un valor astronómicamente incorrecto, disparando una falsa alarma de emergencia. La metodología TDD de MoAI-ADK es esencial aquí: se deben escribir pruebas unitarias con vectores de prueba hexadecimales reales para asegurar que el parser nunca falle ante datos mal formados.<sup>4</sup>

### 4.3 Abstracción a través de Health Connect y HealthKit

Para maximizar la compatibilidad y no depender exclusivamente de la implementación directa de BLE para cada fabricante, la aplicación debe integrarse con **Android Health Connect** y **Apple HealthKit**.<sup>40</sup> Estas plataformas actúan como capas intermedias de abstracción.

- **Health Connect (Android):** Permite leer datos escritos por otras apps (ej. la app propietaria del glucómetro) siempre que el usuario otorgue permisos. Soporta tipos de datos específicos como BloodGlucose, BloodPressure, y HeartRateVariabilityRmssdRecord (crucial para análisis de estrés cardíaco).<sup>40</sup>
- **HealthKit (iOS):** Ofrece funcionalidades similares, pero añade un valor inmenso con la detección de ritmo irregular. La app puede suscribirse a HKCategoryTypeIdentifierIrregularHeartRhythmEvent para recibir notificaciones pasivas si el Apple Watch detecta fibrilación auricular, un precursor común de infartos y accidentes cerebrovasculares.<sup>44</sup>

---

## 5. Detección de Anomalías y Algoritmos de Salud Predictiva

La funcionalidad más avanzada de la aplicación es su capacidad para detectar "datos de infarto" o crisis hipertensivas. Esto implica procesar señales fisiológicas en busca de patrones patológicos.

### 5.1 Algoritmos de Detección de Eventos Cardíacos

La detección de un infarto agudo de miocardio (IAM) o arritmias severas mediante wearables se basa en el análisis de la variabilidad de la frecuencia cardíaca (HRV) y la morfología de la onda ECG (si el dispositivo lo soporta, como Apple Watch o dispositivos con sensores de un

solo canal).<sup>46</sup>

La arquitectura de detección de anomalías propuesta utiliza un enfoque híbrido:

**1. Monitoreo Basado en Reglas (Determinista):**

- Umbrales inmediatos definidos médicamente (ej. Presión Sistólica > 180 mmHg o Glucosa < 50 mg/dL). Estas reglas se ejecutan localmente en el dispositivo para latencia cero.<sup>48</sup>

**2. Monitoreo Basado en IA (Probabilístico):**

- Implementación de modelos de aprendizaje profundo (Deep Learning) ligeros, como Redes Neuronales Convolucionales (CNN) o arquitecturas ResNet-LSTM, entrenados para identificar desviaciones en series temporales de signos vitales.<sup>47</sup>
- Técnicas de *Isolation Forest* o *Autoencoders* LSTM son efectivas para detectar anomalías en datos secuenciales sin necesidad de etiquetas de "enfermedad" previas, aprendiendo simplemente el "estado normal" del paciente y alertando sobre desviaciones significativas.<sup>50</sup>

*Implementación:* El agente mcp-integrations puede facilitar la integración de modelos TensorFlow Lite o PyTorch Mobile directamente en la aplicación Flutter, permitiendo que la inferencia de IA ocurra en el dispositivo ("Edge AI"), preservando la privacidad de los datos y funcionando sin conexión a internet.<sup>53</sup>

## 5.2 Lógica de Alertas Inteligentes y Escalada

Para evitar la "fatiga de alertas", el sistema debe diferenciar entre una lectura anómala aislada y una tendencia peligrosa.

- **Validación de Datos:** Si se detecta un valor crítico, el sistema debe solicitar (si es posible) una re-verificación inmediata para descartar errores de sensor o movimiento.<sup>55</sup>
- **Contextualización:** Una frecuencia cardíaca de 140 lpm es normal durante el ejercicio pero crítica en reposo. La app debe cruzar datos con el acelerómetro del teléfono o reloj para entender el contexto de la actividad.<sup>56</sup>

---

## 6. Protocolos de Respuesta a Emergencias y Telemedicina

Cuando se confirma una anomalía crítica, la aplicación debe transicionar de un modo de "monitoreo" a un modo de "respuesta".

### 6.1 Integración con Servicios de Emergencia (911)

En situaciones de vida o muerte, la app debe facilitar el contacto con servicios de emergencia. En México (y regiones compatibles), esto se puede potenciar mediante la nueva



funcionalidad **Android Emergency Live Video**.<sup>57</sup>

- **Mecanismo:** Al iniciar una llamada al 911 desde la app, esta puede activar automáticamente la transmisión de video en vivo y compartir la ubicación precisa (vía ELS - Emergency Location Service) con el centro de comando (C5), permitiendo a los despachadores ver la condición del paciente en tiempo real.<sup>57</sup>
- **Intent de Android:** La implementación técnica requiere el uso correcto de `Intent.ACTION_DIAL` con el número de emergencia, asegurando que el sistema operativo maneje la llamada con la prioridad adecuada y active los servicios de ubicación de emergencia.<sup>60</sup>

## 6.2 Telemedicina y Consultas Virtuales

Para situaciones no críticas pero urgentes (ej. hiperglucemia sintomática pero estable), la app sugiere una consulta.

- **WebRTC para Video-Consultas:** La integración de un módulo de telemedicina basado en WebRTC (usando paquetes como `flutter_webrtc`) permite la comunicación encriptada peer-to-peer entre el paciente y el equipo médico de la organización.<sup>33</sup>
- **Integración de Datos en Llamada:** Durante la videollamada, la app debe superponer los datos vitales recientes (glucosa, presión, últimas tomas de medicamento) en la pantalla del médico, proporcionando contexto inmediato para la toma de decisiones clínicas.<sup>64</sup>

---

## 7. Backend, Gobernanza de Datos y Cumplimiento Normativo (México)

El almacenamiento y procesamiento de datos de salud en la nube está sujeto a regulaciones estrictas. En México, la **NOM-024-SSA3-2012** y la **Ley Federal de Protección de Datos Personales en Posesión de los Particulares (LFPDPPP)** dictan la arquitectura del backend.<sup>7</sup>

### 7.1 Selección de Backend: Supabase vs Firebase

Aunque Firebase es popular, **Supabase** (basado en PostgreSQL) se perfila como la opción superior para aplicaciones médicas complejas que requieren cumplimiento normativo estricto.

**Tabla 3: Comparativa de Backend para Cumplimiento Normativo**

Característica	Firebase (NoSQL)	Supabase (SQL)	Ventaja para Salud/México
Modelo de Datos	Documentos JSON	Relacional (Tablas)	SQL permite

			integridad referencial estricta para historiales clínicos complejos. <sup>65</sup>
<b>Seguridad de Datos</b>	Reglas en cascada	Row Level Security (RLS)	RLS permite definir políticas de acceso granulares: "Solo el Dr. X puede ver los datos del Paciente Y". <sup>10</sup>
<b>Cumplimiento</b>	Servidores Google (Global)	Auto-hospedable / Región	Posibilidad de hospedar en servidores específicos o configuraciones privadas para cumplir soberanía de datos si fuera necesario. <sup>10</sup>
<b>Interoperabilidad</b>	API propietaria	Estándar SQL	Facilita la exportación de datos en formatos estándar para cumplir con la portabilidad de NOM-024. <sup>7</sup>

## 7.2 Cumplimiento de la NOM-024-SSA3-2012 y LFPDPPP

- **Expediente Clínico Electrónico (ECE):** La norma exige que los registros sean inalterables y auditables. PostgreSQL en Supabase permite crear tablas de auditoría inmutables que registran cada acceso y modificación a los datos de salud, cumpliendo con el requisito de trazabilidad.<sup>7</sup>
- **Identificación Única:** El sistema debe utilizar la CURP como identificador clave para la interoperabilidad con otros sistemas del Sistema Nacional de Salud.<sup>7</sup>
- **Consentimiento y Datos Sensibles:** Bajo la LFPDPPP, los datos de salud son "datos sensibles". La app debe implementar flujos de consentimiento explícito (checkboxes no pre-marcados, firmas digitales) antes de recolectar cualquier dato biométrico.<sup>8</sup> La arquitectura debe garantizar que el usuario pueda ejercer sus derechos ARCO (Acceso,

Rectificación, Cancelación y Oposición) directamente desde la app, por ejemplo, descargando sus datos o solicitando su eliminación.<sup>8</sup>

---

## 8. Hoja de Ruta de Implementación con MoAI-ADK

La ejecución del proyecto sigue el flujo de trabajo "Plan-Run-Sync" de MoAI-ADK, diseñado para minimizar la deuda técnica y maximizar la calidad.<sup>4</sup>

### Fase 1: Inicialización y Especificación (Plan)

#### 1. Configuración del Entorno:

```
Bash
uv tool install moai-adk
moai-adk init salud-aldia-app
moai project # Análisis automático y configuración de estructura [4, 13]
```

#### 2. Definición de Especificaciones (SPEC-First):

Utilizar el comando `/moai:1-plan` para definir los módulos críticos.

- *Prompt*: "Diseñar sistema de recordatorios de medicación con lógica de re-intento y escalada a cuidador si no hay confirmación en 30 min."
- *Resultado (Agente manager-spec)*: Generación de archivo `SPEC-001_Medication_Reminders.md` en formato EARS, detallando estados de la alarma, transiciones de UI y lógica de notificación push.<sup>71</sup>

### Fase 2: Desarrollo Guiado por Pruebas (Run)

1. **Implementación TDD**: Ejecutar `/moai:2-run SPEC-001`. El agente `manager-tdd` creará primero los tests fallidos (RED) para la lógica de horarios (ej. manejo de zonas horarias, años bisiestos) y luego generará el código Dart/Flutter para pasar los tests (GREEN), seguido de una refactorización (REFACTOR) para optimizar la legibilidad.<sup>4</sup>
2. **Integración de Hardware**: Solicitar al agente la creación del módulo BLE: `/moai:2-run SPEC-002_BLE_Integration`. El agente `expert-backend` y `mcp-integrations` generarán el código para escanear UUIDs `0x1808` y `0x1810`, implementando los parsers de bytes little-endian con manejo de errores robusto.<sup>4</sup>

### Fase 3: Sincronización y Documentación (Sync)

1. **Actualización de Documentación**: Al finalizar cada sprint, ejecutar `/moai:3-sync`. El agente `manager-docs` leerá el código nuevo y actualizará los diagramas de arquitectura, la documentación de la API y los manuales de usuario, asegurando que la documentación nunca quede obsoleta.<sup>4</sup>
2. **Validación de Calidad**: El sistema ejecutará automáticamente validaciones de seguridad (buscando hardcoded secrets) y de calidad de código (LSP diagnostics) antes de

permitir cualquier commit al repositorio principal.<sup>3</sup>

## Conclusión

La creación de "SaludAlDía" (nombre tentativo) representa un desafío multidimensional que trasciende la codificación tradicional. Se requiere una simbiosis entre diseño empático para el usuario geriátrico, ingeniería biomédica para la integración de sensores, ciencia de datos para la detección de anomalías y rigor legal para el cumplimiento normativo. El framework MoAI-ADK proporciona la estructura de gobierno necesaria para orquestar estas disciplinas, permitiendo entregar un producto que no solo funciona técnicamente, sino que opera con la fiabilidad clínica necesaria para proteger y mejorar la vida de los pacientes crónicos en México.

## Fuentes citadas

1. ELECTRONIC HEALTH RECORD IN MEXICO - Squarespace, acceso: febrero 6, 2026,  
[https://static1.squarespace.com/static/587fdc951b10e30ca5380172/t/618994ae44eb47201b6d7fa0/1636406447871/The+CIU-Working+Paper+Series+2020-III+Electronic+Health+Record\\_ENG+v24+.pdf](https://static1.squarespace.com/static/587fdc951b10e30ca5380172/t/618994ae44eb47201b6d7fa0/1636406447871/The+CIU-Working+Paper+Series+2020-III+Electronic+Health+Record_ENG+v24+.pdf)
2. Live test: Figma → Claude Code → Mobile app in ONE prompt : r/ClaudeAI - Reddit, acceso: febrero 6, 2026,  
[https://www.reddit.com/r/ClaudeAI/comments/1p03xfb/live\\_test\\_figma\\_claude\\_code\\_mobile\\_app\\_in\\_one/](https://www.reddit.com/r/ClaudeAI/comments/1p03xfb/live_test_figma_claude_code_mobile_app_in_one/)
3. MoAI-ADK (Agentic Development Kit) has already been updated to version 0.41.2. - Reddit, acceso: febrero 6, 2026,  
[https://www.reddit.com/r/ClaudeAI/comments/1q8agql/moaiadk\\_agentic\\_development\\_kit\\_has\\_already\\_been/](https://www.reddit.com/r/ClaudeAI/comments/1q8agql/moaiadk_agentic_development_kit_has_already_been/)
4. MoAI-ADK v0.30.2 Released: AI Agent-Powered SPEC-First TDD Development Framework : r/ClaudeAI - Reddit, acceso: febrero 6, 2026,  
[https://www.reddit.com/r/ClaudeAI/comments/1p7o4zf/moaiadk\\_v0302\\_released\\_ai\\_agentpowered\\_specfirst/](https://www.reddit.com/r/ClaudeAI/comments/1p7o4zf/moaiadk_v0302_released_ai_agentpowered_specfirst/)
5. MoAI-ADK v1.0.0 Released! - Open Source Agentic Development Kit for Claude Code with One-Line Install : r/ClaudeAI - Reddit, acceso: febrero 6, 2026,  
[https://www.reddit.com/r/ClaudeAI/comments/1qad36n/moaiadk\\_v100\\_released\\_open\\_source\\_agentic/](https://www.reddit.com/r/ClaudeAI/comments/1qad36n/moaiadk_v100_released_open_source_agentic/)
6. moai-adk · PyPI, acceso: febrero 6, 2026, <https://pypi.org/project/moai-adk/0.25.9/>
7. NORMA Oficial Mexicana NOM-024-SSA3-2010, Que establece los objetivos funcionales y funcionalidades que deberán, acceso: febrero 6, 2026,  
[https://www.ucol.mx/content/cms/13/file/NOM/NOM-024-SSA3-2010\\_SistemasE CE.pdf](https://www.ucol.mx/content/cms/13/file/NOM/NOM-024-SSA3-2010_SistemasE CE.pdf)
8. Ley Federal de Protección de Datos Personales en Posesión de los Particulares - Cámara de Diputados, acceso: febrero 6, 2026,  
<https://www.diputados.gob.mx/LeyesBiblio/pdf/LFPDPPP.pdf>
9. GOOS (u/Goos\_Kim) - Reddit, acceso: febrero 6, 2026,

- [https://www.reddit.com/user/Goos\\_Kim/](https://www.reddit.com/user/Goos_Kim/)
10. Supabase vs Firebase 2025 Guide for Mobile Apps - Netclues Technologies, acceso: febrero 6, 2026, <https://www.netclues.com/blog/supabase-vs-firebase-baas-comparison-guide>
  11. Does WCAG Apply to Mobile Apps? - AudioEye, acceso: febrero 6, 2026, <https://www.audioeye.com/post/does-wcag-apply-to-mobile-apps/>
  12. MoAI Workflow SPEC | Claude Code Skill for Requirements - MCP Market, acceso: febrero 6, 2026, <https://mcpmarket.com/es/tools/skills/moai-workflow-spec>
  13. modu-ai/moai-adk: MoAI - Agentic Development Kit - GitHub, acceso: febrero 6, 2026, <https://github.com/modu-ai/moai-adk>
  14. moai-workflow-spec - Skill | Smithery, acceso: febrero 6, 2026, <https://smithery.ai/skills/hongcha-poodle/moai-workflow-spec>
  15. Guide to Healthcare App Design for Elderly Users | Orangesoft, acceso: febrero 6, 2026, <https://orangesoft.co/blog/guide-to-designing-healthcare-apps-for-seniors>
  16. UX for Elderly Users: How to Design Patient-Friendly Interfaces - Cadabra Studio, acceso: febrero 6, 2026, <https://cadabra.studio/blog/ux-for-elderly/>
  17. Accessibility of Web Content and Mobile Apps Provided by State and Local Government Entities: A Small Entity Compliance Guide | ADA.gov, acceso: febrero 6, 2026, <https://www.ada.gov/resources/small-entity-compliance-guide/>
  18. A Guide To Interface Design for Older Adults - Adchitects, acceso: febrero 6, 2026, <https://adchitects.co/blog/guide-to-interface-design-for-older-adults>
  19. Elderly-friendly Website/Mobile Application Design Guide - Digital Policy Office, acceso: febrero 6, 2026, [https://www.digitalpolicy.gov.hk/en/our\\_work/digital\\_government/digital\\_inclusion/accessibility/promulgating\\_resources/application\\_design\\_guide/doc/elderly\\_friendly\\_design\\_guide\\_eng.pdf](https://www.digitalpolicy.gov.hk/en/our_work/digital_government/digital_inclusion/accessibility/promulgating_resources/application_design_guide/doc/elderly_friendly_design_guide_eng.pdf)
  20. How Do I Design Apps for Elderly People?, acceso: febrero 6, 2026, <https://thisisglance.com/learning-centre/how-do-i-design-apps-for-elderly-people>
  21. Design Guidelines of Mobile Apps for Older Adults: Systematic Review and Thematic Analysis - PMC, acceso: febrero 6, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC10557006/>
  22. Best Communication Apps and Devices for Older Adults - Technology | Medical Alert Blog, acceso: febrero 6, 2026, <https://www.medicalguardian.com/medical-alert-blog/technology/best-communication-apps-and-devices-for-older-adults>
  23. 5 Senior-Friendly Voice-Controlled Apps to Support Independent Living - Arbor Company, acceso: febrero 6, 2026, <https://www.arborcompany.com/locations/new-jersey/teaneck/blog/5-senior-friendly-voice-controlled-apps-to-support-independent-living>
  24. Seniors & Technology: 5 Apps for Older Adults - SageAge, acceso: febrero 6, 2026, <https://www.sageage.com/blog/seniors-and-technology/>
  25. Optimizing mobile app design for older adults: systematic review of age-friendly design, acceso: febrero 6, 2026,

- <https://pmc.ncbi.nlm.nih.gov/articles/PMC12350549/>
26. Older Users and Web Accessibility: Meeting the Needs of Ageing Web Users - W3C, acceso: febrero 6, 2026, <https://www.w3.org/WAI/older-users/>
  27. Skill for modu-ai/moai-rank - Skills.sh, acceso: febrero 6, 2026, <https://skills.sh/modu-ai/moai-rank>
  28. Top Flutter Bluetooth, NFC, Beacon & Bluetooth Low Energy (BLE) packages - Flutter Gems, acceso: febrero 6, 2026, <https://fluttergems.dev/bluetooth-nfc-beacon/>
  29. How to Integrate ANY Wearable with Your React Native App - Blog, acceso: febrero 6, 2026, <https://blog.xmartlabs.com/blog/wereables-react-native-integration/>
  30. Flutter + Wearables: Integrating Smartwatches and Sensors for Real-Time Fitness Tracking, acceso: febrero 6, 2026, <https://touchlane.com/flutter-wearables-integrating-smartwatches-and-sensors-for-real-time-fitness-tracking/>
  31. react-native-ble-plx 3.3.0 | Documentation, acceso: febrero 6, 2026, <https://dotintent.github.io/react-native-ble-plx/>
  32. Cardiac Monitoring Platform for Wearable Health Technology - Binary Studio, acceso: febrero 6, 2026, <https://binary-studio.com/case-studies/wearable-health-software-development/>
  33. A Comprehensive Guide to Flutter WebRTC - 100ms.live, acceso: febrero 6, 2026, <https://www.100ms.live/blog/flutter-webrtc>
  34. Glucose Profile | Bluetooth® Technology Website, acceso: febrero 6, 2026, <https://www.bluetooth.com/specifications/specs/glp-1-0-1/>
  35. How to get glucose meter data in flutter BLE using flutter\_blue - Stack Overflow, acceso: febrero 6, 2026, <https://stackoverflow.com/questions/63953601/how-to-get-glucose-meter-data-in-flutter-ble-using-flutter-blue>
  36. Monitor Blood Pressure , Heart Rate & Sugar Levels BLEHealth+, acceso: febrero 6, 2026, <https://www.blehealthplus.com/>
  37. get weight from GATT characteristics value in react-native-ble-plx - Stack Overflow, acceso: febrero 6, 2026, <https://stackoverflow.com/questions/70952070/get-weight-from-gatt-characteristics-value-in-react-native-ble-plx>
  38. An Android-Based Heart Monitoring System for the Elderly and for Patients with Heart Disease - PMC - NIH, acceso: febrero 6, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC4152930/>
  39. How to decode response from Glucose Measurement Characteristic (Bluetooth), acceso: febrero 6, 2026, <https://stackoverflow.com/questions/63336244/how-to-decode-response-from-glucose-measurement-characteristic-bluetooth>
  40. Health Connect | Android health & fitness - Android Developers, acceso: febrero 6, 2026, <https://developer.android.com/health-and-fitness/health-connect>
  41. Protecting access to user's health data - Apple Support, acceso: febrero 6, 2026, <https://support.apple.com/guide/security/protecting-access-to-users-health-dat>

- [a-sec88be9900f/web](#)
42. HeartRateVariabilityRmssdRecord | API reference - Android Developers, acceso: febrero 6, 2026,  
<https://developer.android.com/reference/android/health/connect/datatypes/HeartRateVariabilityRmssdRecord>
  43. Health Connect data types | Android health & fitness - Android Developers, acceso: febrero 6, 2026,  
<https://developer.android.com/health-and-fitness/health-connect/data-types>
  44. irregularHeartRhythmEvent | Apple Developer Documentation, acceso: febrero 6, 2026,  
<https://developer.apple.com/documentation/healthkit/hkcategorytypeidentifier/irregularheartrhythmevent>
  45. HKCategoryTypeIdentifierIrregular, acceso: febrero 6, 2026,  
<https://developer.apple.com/documentation/healthkit/hkcategorytypeidentifier/irregularheartrhythmevent?language=objc>
  46. An AI tool detected structural heart disease in adults using a smartwatch, acceso: febrero 6, 2026,  
<https://newsroom.heart.org/news/an-ai-tool-detected-structural-heart-disease-in-adults-using-a-smartwatch>
  47. Wearable ECG Device and Machine Learning for Heart Monitoring - PMC - PubMed Central, acceso: febrero 6, 2026,  
<https://pmc.ncbi.nlm.nih.gov/articles/PMC11244216/>
  48. Real-Time Remote Patient Monitoring and Alarming System for Noncommunicable Lifestyle Diseases - PMC, acceso: febrero 6, 2026,  
<https://pmc.ncbi.nlm.nih.gov/articles/PMC10681793/>
  49. Detecting anomalies in smart wearables for hypertension: a deep learning mechanism, acceso: febrero 6, 2026,  
<https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2024.1426168/full>
  50. Anomaly Detection in Elderly Health Monitoring via IoT for Timely Interventions - MDPI, acceso: febrero 6, 2026, <https://www.mdpi.com/2076-3417/15/13/7272>
  51. Time Series Anomaly Detection Tutorial with PyTorch in Python | LSTM Autoencoder for ECG Data - YouTube, acceso: febrero 6, 2026,  
<https://www.youtube.com/watch?v=qN3n0TM4Jno>
  52. Introduction to Anomaly Detection in Python with PyCaret | by Moez Ali - Medium, acceso: febrero 6, 2026,  
<https://medium.com/data-science/introduction-to-anomaly-detection-in-python-with-pycaret-2fec7144f87>
  53. README.md - Loai-Houmane/2D-Anomaly-Detection - GitHub, acceso: febrero 6, 2026,  
<https://github.com/Loai-Houmane/2D-Anomaly-Detection/blob/main/README.md>
  54. This project detects anomalies in 2D data using PyTorch for model training and Flutter for a cross-platform application. Key features include pre-trained models, a Flutter mobile app that shows heat maps, a Flask server backend, and a  Tkinter desktop app. - GitHub, acceso: febrero 6, 2026,



- <https://github.com/Loai-Houmane/2D-Anomaly-Detection>
55. How Mobile Apps Power Smart ECG Monitors and Wearable Heart Sensors, acceso: febrero 6, 2026, <https://www.sidekickinteractive.com/uncategorized/how-mobile-apps-power-smart-ecg-monitors-and-wearable-heart-sensors/>
  56. Anomaly-based threat detection in smart health using machine learning - PMC - NIH, acceso: febrero 6, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC11577804/>
  57. Google Gives Android More 911 Call Center Responsibility - Government Technology, acceso: febrero 6, 2026, <https://www.govtech.com/biz/google-gives-android-more-911-call-center-responsibility>
  58. New Android Emergency Services Feature! HUGE Update: Android 8+ Newer! - YouTube, acceso: febrero 6, 2026, <https://www.youtube.com/watch?v=-Gi8o9Y9g00>
  59. Motorola Solutions and Google Let Android Users Share Live Video with 911 in Seconds, acceso: febrero 6, 2026, <https://www.motorolasolutions.com/newsroom/press-releases/msi-and-android-users-share-live-video-with-911-in-seconds.html>
  60. 911 CDMX - Apps on Google Play, acceso: febrero 6, 2026, [https://play.google.com/store/apps/details?id=com.desarrollo.c5.app\\_066\\_android\\_2](https://play.google.com/store/apps/details?id=com.desarrollo.c5.app_066_android_2)
  61. Emergency numbers and emergency calling | Android Open Source Project, acceso: febrero 6, 2026, <https://source.android.com/docs/core/connect/emergency-call>
  62. Build a Video Chat App in Flutter – Real-Time Calling with WebRTC - Meet Hour, acceso: febrero 6, 2026, <https://meethour.io/blog/video-conference-integration/how-to-build-a-video-calling-app-in-flutter-using-meet-hour-integration>
  63. Flutter-WebRTC Guide: Build your first Flutter WebRTC app | by Video SDK - Medium, acceso: febrero 6, 2026, <https://medium.com/video-sdk/flutter-webrtc-guide-build-your-first-flutter-webrtc-app-927501f8d75d>
  64. Clone MedTalk: HIPAA-Ready Video and Chat Consultations in Flutter, acceso: febrero 6, 2026, <https://getstream.io/blog/flutter-telehealth-app/>
  65. Supabase vs Firebase, acceso: febrero 6, 2026, <https://supabase.com/alternatives/supabase-vs-firebase>
  66. Firebase vs Supabase vs Custom Backend – Which To Choose? - Mobitouch, acceso: febrero 6, 2026, <https://mobitouch.net/blog/firebase-vs-supabase-vs-custom-backend-which-backend-should-you-choose>
  67. DOF-30NOV12-NOM-024-SSA3-2012.pdf - DIRECCIÓN GENERAL DE INFORMACIÓN EN SALUD, acceso: febrero 6, 2026, <http://www.dgjis.salud.gob.mx/descargas/normatividad/normas/DOF-30NOV12-NOM-024-SSA3-2012.pdf>
  68. LEY DE PROTECCIÓN DE DATOS PERSONALES EN POSESIÓN DE SUJETOS



OBLIGADOS DE LA CIUDAD DE MÉXICO - Congreso de la CDMX, acceso: febrero 6, 2026,

<https://www.congresocdmx.gob.mx/archivo-a8b6a4657686ea6f01e7a024377b814e0664d634.pdf>

69. Ley General de Protección de Datos Personales en Posesión de Sujetos

Obligados - Cámara de Diputados, acceso: febrero 6, 2026,

<https://www.diputados.gob.mx/LeyesBiblio/pdf/LGPDPPSO.pdf>

70. moai-adk · PyPI, acceso: febrero 6, 2026, <https://pypi.org/project/moai-adk/0.32.5/>

71. r/moai\_adk - Reddit, acceso: febrero 6, 2026, [https://www.reddit.com/r/moai\\_adk/](https://www.reddit.com/r/moai_adk/)

72. moai-foundation-core by modu-ai/moai-adk - Skills.sh, acceso: febrero 6, 2026,

<https://skills.sh/modu-ai/moai-adk/moai-foundation-core>