

UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional de Córdoba

Ingeniería en Sistemas de Información
Proyecto Final



Grupo N 12

Romero Plaza, Mateo

Kilic Aslan, Haik Martín

Rodeyro Contarino, Nicolás

Maldonado Gómez, Jeremías Daniel

Escudero Marusich, Octavio Matias

Legajos

83009

85246

86537

86291

89866

Docentes

Ing. Quinteros, Sergio

Mgter Ing. Santos, Virginia

Curso

5K3

AÑO 2026



AURORA
Luz que acompaña



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL CÓRDOBA
INGENIERÍA EN SISTEMAS DE INFORMACIÓN
CÁTEDRA: PROYECTO FINAL

Sistema de acompañamiento domiciliario para personas con pérdida de memoria

AÑO DE CURSADO
2026

Necesidad

Cuando la memoria falla, las rutinas cotidianas dejan de ser seguras.



Pérdida de Autonomía



Cuidado Continuo



Soluciones Separadas

Objetivos



ACOMPañAR



ORIENTAR



ANTICIPAR

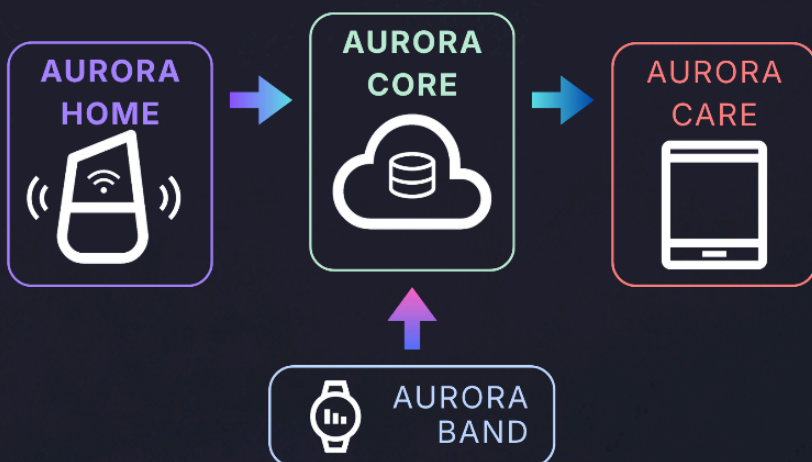


CONECTAR



PROTEGER

Ecosistema y Arquitectura



Resultados y Beneficiarios



Flujo de Alerta: Recordatorio de Medicación



Una rutina programada activa el recordatorio



Aurora Home avisa por comando de voz



Aurora Core evalúa la respuesta del paciente



Aurora Care notifica al cuidador

Herramientas y Tecnologías Aplicadas



Autores

Escudero Octavio
octavioescudero448@gmail.com

Kilic Haik
kilichaik@gmail.com

Maldonado Gomez Jeremias
jmalonadogomez21@gmail.com

Rodeyro Contarino Nicolas
nicolasrodeyro@gmail.com

Romero Plaza Mateo
mateoromero plaza12@gmail.com

Docentes

• Ing. Quinteros, Sergio
• Mgter Ing. Santos, Virginia

Curso: 5K3 - Año: 2026



Aurora: Sistema Inteligente de Acompañamiento y Monitoreo Domiciliario para Pacientes con Pérdida de Memoria

Escudero Marusich, Octavio Matías

Kilic Aslan, Haik Martín

Maldonado Gómez, Jeremías Daniel

Rodeyro Contarino, Nicolás

Romero Plaza, Mateo

Universidad Tecnológica Nacional, Facultad Regional Córdoba — Ingeniería en Sistemas de Información

Abstract

Se desarrolló un ecosistema tecnológico integral denominado Aurora, orientado al acompañamiento y monitoreo domiciliario de personas con pérdida de memoria. El diseño transitó desde un esquema de cuidado reactivo hacia un modelo proactivo basado en inteligencia artificial, interacción por voz y monitoreo biométrico continuo. La arquitectura se compuso de cuatro subsistemas: Aurora Home (asistente de voz en Raspberry Pi con Whisper.cpp), Aurora Band (pulsera IoT), Aurora Care (aplicación móvil con React Native) y Aurora Core (orquestración con Django y PostgreSQL). Se adoptó un patrón de microservicios con comunicación REST y Redis Pub/Sub sobre infraestructura híbrida. Se implementó un modo degradado para operación sin conectividad, garantizando recordatorios críticos y buffers de eventos. Los resultados arquitectónicos constituyeron la base para el Producto Mínimo Viable.

Palabras Clave: Alzheimer, asistencia domiciliar, inteligencia artificial, interacción por voz, Internet de las Cosas, monitoreo biométrico, terapia de reminiscencia.

Clasificación: Producto

Introducción

La enfermedad de Alzheimer afecta a más de 55 millones de personas a nivel mundial, con proyecciones de 139 millones para 2050 [1]. En Argentina, aproximadamente 1,2 millones padecen demencia, generando una carga significativa para el sistema de salud y las familias [2]. Las personas con deterioro

cognitivo presentan dificultades para mantener rutinas, tomar medicación y realizar actividades cotidianas, afectando su autonomía e incrementando el estrés del cuidador [3]. Las soluciones existentes suelen centrarse en aspectos parciales como localización GPS o recordatorios genéricos [4], sin integrar interacción por voz, monitoreo biométrico e inteligencia artificial. Aurora es una solución integral para pacientes con deterioro menor a la etapa 5 de la escala FAST [5], combinando interacción conversacional, monitoreo continuo y automatización de rutinas.

Elementos del Trabajo y Metodología

La arquitectura de Aurora se modeló mediante el marco C4 [6] en tres niveles: Contexto, Contenedores y Despliegue.

En el Nivel 1, el sistema interactuó con el paciente (voz) y el cuidador (aplicación). Se establecieron conexiones con servicios externos de LLM, síntesis de voz, mensajería y telefonía de emergencia.

En el Nivel 2 (Contenedores), el ecosistema se descompone en ocho componentes desplegables: Aurora Home (Raspberry Pi con Whisper.cpp), Aurora Core (Django REST Framework + Channels [8]), Worker IA (FastAPI), Aurora Band Gateway (DRF), Base de Datos (Supabase PostgreSQL + pgvector),

Redis (caché + pub-sub), n8n (orquestador de flujos) y Aurora Care (React Native + Expo mobile app).

En el Nivel 3 (Despliegue), la infraestructura se distribuyó de manera híbrida: los servicios backend se desplegaron mediante Docker Compose en un VPS de Hostinger [14]; la base de datos se alojó en Supabase Cloud; y el frontend de Aurora Care se desplegó como aplicación distribuida en Play Store de Google. La Raspberry Pi en el hogar del paciente ejecutó Aurora Home con caché Redis local.

Las decisiones técnicas se formalizaron mediante registros ADR [7]. Se seleccionó Python y Django para el backend (ADR-001) y React Native para la aplicación móvil (ADR-002). La persistencia se delegó en Supabase (ADR-003). El procesamiento STT se ejecutó localmente en Raspberry Pi con Whisper.cpp para priorizar la privacidad (ADR-006).

Se adopta OAuth 2.0 / OpenID Connect (OIDC) como mecanismo de autenticación para los cuidadores, utilizando un proveedor externo. Aurora Core mantiene la relación entre los cuidadores y las cuentas DOT mediante asociaciones en la base de datos, garantizando un acceso seguro y desacoplado.

El desarrollo siguió la metodología Scrum con sprints de tres semanas. Se utilizó capacity planning para un equipo de cinco integrantes con un total de 84 horas semanales [12]. Se aplicaron políticas de revisión cruzada de código y uso asistido de inteligencia artificial con supervisión humana obligatoria [13].

El desarrollo siguió la metodología Scrum con sprints de tres semanas. Se utilizó capacity planning para un equipo de cinco integrantes con un total de 84 horas semanales [12]. Se aplicaron políticas de revisión cruzada de código y uso asistido de inteligencia artificial con supervisión

humana obligatoria [13].

Resultados

La arquitectura de microservicios se desplegó exitosamente con Docker Compose, validando tolerancia a fallos parciales. El modelo de datos en PostgreSQL con pgvector incluyó tablas para pacientes, cuidadores, eventos, alertas, rutinas, medicación y memoria contextual, habilitando búsqueda vectorial para RAG. El modo offline ejecuta recordatorios localmente en la Raspberry Pi con TTS offline (Piper) y sincronización automática al reconectar. El costo operativo proyectado para 10 usuarios es de 27 USD mensuales: VPS (10 USD), Supabase (15 USD) e IA (2 USD). La infraestructura de CI/CD se configuró con GitHub Actions.

Discusión

Aurora integra capacidades tradicionalmente dispersas en un único ecosistema [4]. El procesamiento local de voz prioriza la privacidad sobre la precisión de modelos cloud [11], coherente con la Ley 25.326 [15]. Los microservicios agregan complejidad operativa pero aseguran disponibilidad de recordatorios críticos [10]. PostgreSQL con pgvector aborda persistencia relacional y búsqueda semántica para RAG [9]. La infraestructura híbrida con Docker Compose permite migración flexible [7]. Aurora no reemplaza la atención médica; complementa la labor del cuidador [1].

Conclusión

Aurora es un ecosistema tecnológico que resuelve el acompañamiento domiciliario mediante IA y monitoreo biométrico. El diseño arquitectónico es viable bajo restricciones de costo y privacidad. El sistema extiende la autonomía del paciente y reduce la carga del cuidador mediante un modelo de asistencia proactiva.

Los principales aportes del sistema incluyen: el procesamiento de voz en el

borde (edge) que preserva la privacidad del paciente, la estrategia de modo offline que garantiza la disponibilidad de recordatorios críticos ante fallas de conectividad, y la integración de RAG para personalizar las interacciones con recuerdos y preferencias del paciente. Estos componentes buscan extender la autonomía del paciente y reducir la carga del cuidador, transitando desde un modelo reactivo hacia uno proactivo de acompañamiento.

Como trabajo futuro se prevé la validación del sistema con un grupo piloto de usuarios reales, la incorporación de wearables adicionales con sensores de temperatura corporal y actividad electrodérmica, la ampliación de canales de notificación, y la exploración de modelos de escalabilidad B2B para centros de salud y geriátricos.

Agradecimientos

Se agradece a la Mgter. Ing. Virginia Santos por su orientación y apoyo durante el desarrollo. Asimismo, se agradece a la cátedra Proyecto Final de la Facultad Regional Córdoba de la Universidad Tecnológica Nacional por la oportunidad de desarrollo de este trabajo.

Referencias

- [1] Organización Mundial de la Salud. "Dementia," 2024.
- [2] Alzheimer's Association. "2024 Alzheimer's disease facts and figures," Alzheimer's & Dementia, vol. 20, no. 5, pp. 3708-3821, 2024.
- [3] World Health Organization. "Global action plan on the public health response to dementia 2017–2025," Ginebra: OMS, 2019.
- [4] L. Gavelin et al., "Technology-assisted interventions for persons with dementia: a systematic review and meta-analysis," Aging & Mental Health, vol. 25, no. 9, pp. 1635-1649, 2021.
- [5] B. Reisberg, "Global measures of disability in Alzheimer's disease," International Psychogeriatrics, vol. 19, no. 3, pp. 529-556, 2007.
- [6] S. Brown, "The C4 model for visualising software architecture," 2024.
- [7] A. Rooijmans, "Architecture Decision Records," Lean Architecture for Agile Software Development, 2020.
- [8] Django Software Foundation. "Django," 2024.

[9] Supabase Inc. "Supabase," 2024.

[10] C. Richardson, Microservices Patterns, Manning Publications, 2019.

[11] G. et al., "Whisper: Robust Speech Recognition via Large-Scale Weak Supervision," Proceedings of the 40th International Conference on Machine Learning, 2023.

[12] K. Schwaber y J. Sutherland, "The Scrum Guide," Scrum.org, 2020.

[13] GitHub Inc. "GitHub Copilot," 2024.

[14] Hostinger International. "VPS Hosting," 2024.

[15] República Argentina. "Ley 25.326: Protección de Datos Personales," Boletín Oficial, 4 de noviembre de 2000.

[16] GitHub Inc. "GitHub Copilot — AI pair programmer," 2024.

[17] Google. "Gemini — AI assistant," 2024.

[18] OpenAI. "ChatGPT — AI language model," 2024.

Datos de Contacto

Jeremías Daniel Maldonado Gómez

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Email: jmaldonadogomez21@gmail.com

PLANILLA PARA CATALOGAR EL PROYECTO FINAL

AÑO	2026	CURSO Y NRO. DE GRUPO	5K3 - G12
NOMBRE DEL SISTEMA / PROYECTO			
AURORA			
CATEGORÍA (Solución De Negocio / Producto / Proyecto De Impacto Social)			
Producto			
HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS			
ÁMBITO DE APLICACIÓN		NOMBRE Y VERSIÓN	
ENTORNO DE DESARROLLO		Visual Studio Code 1.130.0	
REPOSITORIOS Y VERSIONADO		Git Hub	
PROGRAMACIÓN		React 19.1 React Native 0.81.5 Python 3.14.3	
BASE DE DATOS		PostgreSQL	
COMUNICACIÓN INTERNA		Discord, WhatsApp	
CAPACITACIÓN		YouTube - Chat GPT - Claude	
PRUEBAS DE SISTEMA		- Pruebas manuales: Google Chrome, Postman, OpenAPI/Swagger - Test unitarios: pytest, pytest-django	
GESTIÓN DEL PROYECTO		Jira	
DOCUMENTACIÓN		Google Drive - GitHub	
MODELOS		Draw.io	