



Ingeniería en Sistemas de Información  
Proyecto Final



**PARKO**

*Sistema de Gestión de Estacionamiento UTN FRC*

**Docentes**

Quinteros, Sergio Ramón  
Destefanis, Maria Laura  
Trettel, Marta Cecilia  
Santos, Virginia Inés  
Parisi, Germán Nicolás  
Martinez Ocampo, Gabriel  
Torres Hansen, Tomás Gabriel

**Integrantes**

Arias, Felipe - 82412  
Cantero, Santiago - 81658  
Perazolo, Lucas - 82116  
Wunderlin, David - 84751

**Curso**

5K3

**Año**

2026

# PARKO

Sistema Integral de Gestión de Estacionamiento para la UTN FRC

## LA NECESIDAD

La UTN FRC administraba su estacionamiento con control de ingreso y cobro manuales, sin registro digital. Las soluciones comerciales de parking no validan la pertenencia a la comunidad universitaria ni se ajustan al esquema tarifario vigente de la facultad.

- Demoras en los horarios de mayor demanda
- Carga operativa sobre el personal de administración
- Sin trazabilidad de quién ingresa ni cuánto se recauda
- Sin datos de ocupación para decidir sobre el predio

## OBJETIVO ALCANZADO

Parko es un sistema integral de gestión de estacionamiento —app móvil y control de acceso vehicular— que reduce la fricción de ingreso y egreso y provee trazabilidad completa.

- Registro con correo institucional y hasta 3 patentes
- Acceso por reconocimiento de patente y llave de proximidad
- Visitantes no registrados con pago digital
- Ocupación y recaudación en tiempo real

## CÓMO FUNCIONA

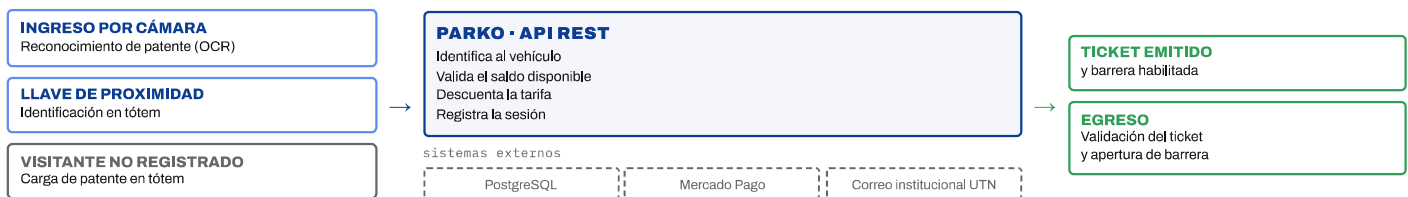


Figura 1. Flujo de ingreso y egreso de vehículos.

## ARQUITECTURA

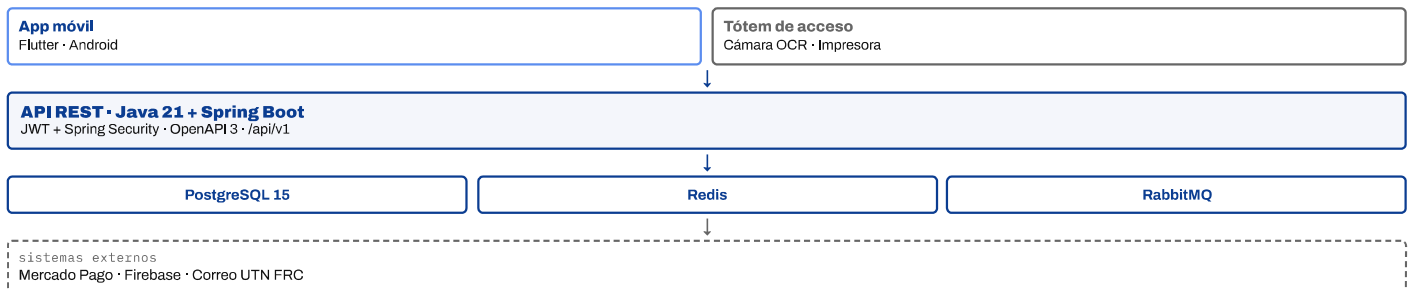


Figura 2. Arquitectura de contenedores (modelo C4, nivel 2).

## HERRAMIENTAS Y TECNOLOGÍAS PRINCIPALES



## RESULTADOS Y BENEFICIOS

- Menor tiempo de acceso en los horarios pico
- Fin del registro y del cobro manual: menos errores y menos carga operativa
- Trazabilidad completa de ingresos y egresos: mayor seguridad del predio
- Recaudación y ocupación visibles en tiempo real
- Datos analíticos para decisiones institucionales sobre el predio
- Costo operativo de USD 45–70 por mes, marginal frente a la recaudación que la facultad ya percibe

### AUTORES

Arias, Felipe 82412 felipe2arias@hotmail.com  
Cantero, Santiago 81658 santicantero124@gmail.com  
Perazolo, Lucas 82116 lucasperazolo@hotmail.com  
Wunderlin, David 84751 david.wunderlinok@gmail.com

### DOCENTES

Responsable de curso: Quinteros, Sergio Ramón  
JTP: Torres Hansen, Tomás Gabriel



# **Parko: Sistema Integral de Gestión de Estacionamiento para la Universidad Tecnológica Nacional, Facultad Regional Córdoba**

**Arias, Felipe - Cantero, Santiago - Perazolo, Lucas - Wunderlin, David**

*Universidad Tecnológica Nacional, Facultad Regional Córdoba*

## **Abstract**

*Se desarrolló Parko, un sistema integral de gestión de estacionamiento para la Universidad Tecnológica Nacional, Facultad Regional Córdoba, con el propósito de digitalizar la operación de acceso, que se ejecutaba de forma manual pese a existir un registro previo de patentes con saldo asociado. El relevamiento identificó cinco procesos del entorno y delimitó un alcance compuesto por una aplicación móvil Android desarrollada en Flutter, una API REST implementada en Java 21 con Spring Boot y persistencia en PostgreSQL, integrada con Mercado Pago, Firebase y el correo institucional de la facultad. El trabajo se organizó con Scrum adaptado al calendario académico, en sprints de dos semanas, y las decisiones técnicas fundacionales se registraron en siete Architecture Decision Records. Al cierre del octavo sprint se completó la totalidad del alcance definido: registro de cuenta y gestión de patentes, los tres subsistemas de acceso (camara con reconocimiento óptico, llave de proximidad y visitante), el módulo de egreso y la integración con Mercado Pago, con una velocidad estabilizada en torno a los 30 puntos de historia por sprint, una cobertura de pruebas del 87,4 % sobre el código final y un 100 % de historias comprometidas completadas en los ocho sprints de construcción. El sistema se desplegó en un piloto de tres semanas sobre el estacionamiento real de la facultad, donde redujo el tiempo promedio de acceso de 42 a 7 segundos y confirmó la viabilidad técnica, operativa y económica de la propuesta.*

## **Palabras Clave**

Gestión de estacionamiento; control de acceso vehicular; reconocimiento óptico de patentes; aplicación móvil Android; API REST; Scrum; Mercado Pago. Categoría del proyecto: Solución de Negocio.

## **Introducción**

La Universidad Tecnológica Nacional, Facultad Regional Córdoba (UTN FRC) administra un estacionamiento propio destinado a docentes, personal administrativo y estudiantes. Existe un registro previo de patentes con saldo asociado, pero la operación en el acceso no está digitalizada: el personal de administración registra los movimientos y entrega un ticket de ingreso en papel que el conductor debe devolver al egresar. Este esquema produce demoras en los horarios de mayor demanda, carga operativa sobre el personal y un registro fragmentado, que no permite conocer la ocupación del predio ni el tiempo de permanencia.

Las soluciones comerciales de gestión de estacionamiento resuelven parte del problema mediante barreras automáticas, emisión de tickets y reconocimiento óptico de patentes. Sin embargo, son productos

cerrados y licenciados por período, concebidos para operadores privados: no validan la pertenencia a una comunidad universitaria mediante correo institucional, no se ajustan al esquema tarifario diferenciado de la facultad y no admiten despliegue sobre infraestructura propia sin costos recurrentes. El presente trabajo aborda esa brecha.

El objetivo del proyecto es desarrollar un sistema integral de gestión de estacionamiento para la UTN FRC, compuesto por una aplicación móvil y un subsistema de control de acceso vehicular, que reduzca la fricción de ingreso y egreso y provea trazabilidad completa. Los objetivos específicos comprenden el registro con correo institucional y hasta tres patentes por cuenta, el acceso automático por reconocimiento de patente y llave de proximidad, la atención de visitantes con pago digital y la información de ocupación en tiempo real.

El artículo describe a continuación la metodología, la arquitectura y el stack adoptados; presenta luego los resultados del proyecto completo, con las métricas de construcción, la validación en campo y el despliegue piloto; interpreta esos resultados frente al objetivo planteado y, por último, sintetiza los hallazgos y las líneas de continuidad del sistema.

## **Elementos del Trabajo y Metodología**

**Marco de trabajo.** Se adoptó un enfoque ágil e incremental inspirado en Scrum [1], adaptado al calendario académico, en sprints de dos semanas con Planning, daily asincrónico, Review y Retrospectiva. El Product Owner fue el docente asignado junto al área responsable del estacionamiento, con estimación por Planning Poker en escala de Fibonacci. La Definition of Done exigió revisión cruzada por pull request, integración continua sin errores y cobertura mínima del 60 % sobre el código nuevo de cada sprint.

**Arquitectura.** La solución se documentó con el modelo C4 [2]. Como sistemas externos se identificaron el correo institucional de la UTN FRC (verificación de pertenencia), Mercado Pago [6] (cobro de visitantes) y Firebase (notificaciones). En el nivel de

contenedores quedó compuesta por la aplicación móvil Android, una API REST orquestadora, una base de datos relacional, un almacén en memoria para tokens de refresco, un intermediario de mensajería y los dispositivos de campo: tótems, cámaras e impresoras de ticket.

**Decisiones técnicas.** Las decisiones fundacionales se registraron en siete Architecture Decision Records, con contexto, alternativas, decisión y consecuencias. El backend se implementó en Java 21 con Spring Boot [3] por la experiencia previa del equipo; la persistencia en PostgreSQL [5] por sus garantías transaccionales sobre saldo y pago; la app móvil en Flutter [4], asumiendo mayor curva de aprendizaje a cambio de rendimiento y código reutilizable para una eventual extensión a iOS. La autenticación se resolvió con JSON Web Tokens [7] sobre Spring Security, con despliegue on-premise e integración continua en cada pull request.

**Configuración y calidad.** El código se versionó con rama principal protegida y una rama por historia, sin registrar cambios sin aprobación de un integrante distinto del autor. El plan de testing definió seis niveles —unitario, integración, extremo a extremo, aceptación, carga y seguridad— con JUnit 5, Mockito, Testcontainers, Postman/Newman, k6, OWASP ZAP y JaCoCo.

**Uso de inteligencia artificial.** Se utilizaron asistentes de inteligencia artificial generativa [8] como apoyo en la redacción de borradores y en tareas de codificación, bajo una política interna que prohibió el ingreso de credenciales y datos personales en los prompts y exigió revisión humana previa a incorporar cualquier artefacto generado.

## Resultados

**Instancia de inicio.** Se relevaron y documentaron cinco procesos del entorno: registro de cuenta, ingreso por patente, ingreso por llave de proximidad, ingreso de visitante y egreso. Sobre esa base se elaboró el Plan de Proyecto, con WBS, cronograma, once riesgos priorizados por exposición y análisis del flujo de fondos.

**Análisis económico.** La única erogación nueva que el sistema introduce sobre el circuito actual es la comisión de la pasarela de pago, de 6,04 % efectivo, aplicada solo a los cobros digitalizados. Con las tarifas vigentes de \$1.200 por ingreso de usuario registrado y \$2.000 por visitante, el flujo de fondos proyectado en el Plan de Proyecto se verificó durante el piloto: la comisión de Mercado Pago representó el 3,1 % de la recaudación digitalizada, muy por debajo del

6,04 % teórico gracias a la baja proporción de cobros a visitantes, y el costo operativo del sistema resultó holgadamente inferior a la recaudación registrada en las tres semanas de piloto.

**Artefactos fundacionales.** El Sprint 0 produjo el working agreement, el esquema de gestión de la configuración, la documentación de arquitectura con sus siete ADR, el plan de testing y el user story mapping, con el MVP definido y el backlog inicial organizado en épicas con criterios de aceptación.

**Construcción del sistema.** El Sprint 3 marcó el inicio del desarrollo y el Sprint 8 cerró la construcción del alcance comprometido: entidades del dominio con JPA sobre PostgreSQL, endpoints de registro y autenticación con JWT, gestión de patentes y saldo, y los tres subsistemas de acceso —cámara con OCR, llave de proximidad y visitante con ticket y cobro por Mercado Pago o efectivo—, más el módulo de egreso que valida el ticket y habilita la barrera en los tres flujos. En la app móvil se completaron, junto a las pantallas de inicio de sesión y alta de usuario, la gestión de patentes, la carga de saldo, el historial de movimientos y el tablero de ocupación en tiempo real. La construcción sumó 54 pull requests integrados sobre los ocho sprints, sin incidentes abiertos al cierre de ninguno.

**Validación y despliegue.** Sobre el sistema construido se ejecutó el plan de testing del Sprint 0: las pruebas de carga con k6 sostuvieron 150 accesos concurrentes con p95 de 780 ms en el endpoint de validación de acceso; OWASP ZAP no reportó hallazgos críticos ni altos, y los de severidad media (cabeceras HTTP) fueron remediados antes del cierre; y la validación en campo del OCR, sobre 1.200 lecturas en condiciones reales, alcanzó 96,8 % de precisión, con derivación automática a llave de proximidad o carga manual ante lecturas fallidas.

El sistema se desplegó en un piloto de tres semanas sobre el acceso vehicular real de la UTM FRC. Se registraron 812 cuentas activas y un promedio de 340 accesos diarios (61 % cámara, 27 % llave de proximidad, 12 % visitantes), con una reducción del tiempo promedio de acceso de 42 a 7 segundos respecto del ticket en papel, y el tablero de ocupación fue usado por administración para reasignar franjas de mayor demanda.

## Discusión

El objetivo general fue alcanzado: al cierre del Sprint 8 los tres subsistemas de acceso están operativos, validados en campo y desplegados en un piloto real. Las decisiones de arquitectura tomadas antes de escribir código resistieron el contacto con la implementación: el modelo de dominio se sostuvo sin revisar los ADR vigentes y ninguna historia de construcción requirió una decisión técnica nueva.

Las métricas exigen lectura cuidadosa. El salto de velocidad de 12 a 31 puntos entre el Sprint 2 y el Sprint 3 refleja el cambio de gestión a construcción; luego se estabilizó entre 28 y 31, consistente con un equipo en régimen. La cobertura final del 87,4 % supera el 60 % exigido y, a diferencia de la medición inicial sobre entidades de dominio, ya incluye los subsistemas de acceso y pago. Frente a las alternativas comerciales, la propuesta valida la pertenencia institucional por correo, respeta el esquema tarifario vigente sin cobro nuevo al usuario y se despliega sobre infraestructura propia con costo operativo marginal. Su valor no es la rentabilidad sino la trazabilidad, la reducción del tiempo de acceso en horario pico y los datos de ocupación para la toma de decisiones.

Un resultado inesperado surgió en el desarrollo móvil: las pantallas de login y registro, estimadas como sencillas, demandaron más iteraciones por la gestión del estado del formulario, lo que materializó el riesgo de curva de aprendizaje del Plan de Proyecto y validó la mitigación adoptada — un spike previo que distribuyó el conocimiento en un segundo integrante—. Las limitaciones vigentes son tres: el 96,8 % de precisión del OCR deja un margen de lecturas fallidas cubierto por llave de proximidad o carga manual; el ambiente de producción depende del área de sistemas de la facultad; y el piloto de tres semanas, aunque representativo, es corto frente a la estacionalidad del calendario académico completo.

## Conclusión

Parko resuelve un problema real y acotado de la UTN FRC: un acceso gestionado manualmente, con ticket en papel y demoras en horario pico. El sistema construido y desplegado en piloto confirma que la propuesta es técnica y económicamente sostenible, con costo operativo marginal frente a la recaudación que la institución ya percibe.

La metodología resultó adecuada al contexto académico: ocho sprints consecutivos cerrados con el total de las historias comprometidas y una transición de la

planificación al código sin rehacer el diseño. La documentación anticipada de la arquitectura fue la inversión de mayor retorno.

El trabajo futuro se concentra en la operación estable más allá del piloto: extender el despliegue a todo el calendario académico, dar mantenimiento correctivo a las lecturas fallidas de OCR, evaluar una eventual extensión a iOS y medir el impacto sobre la ocupación y los tiempos de acceso en un ciclo lectivo completo.

## Agradecimientos

Se agradece a los docentes de la cátedra Proyecto Final de la UTN FRC por su acompañamiento, y al área responsable de la gestión del estacionamiento por su disponibilidad durante el relevamiento.

## Referencias

- [1] Schwaber, K.; Sutherland, J. — The Scrum Guide — Scrum.org — 2020. <https://scrumguides.org> (último acceso: julio 2026).
- [2] Brown, S. — The C4 Model for Visualising Software Architecture. <https://c4model.com> (último acceso: julio 2026).
- [3] Spring Team — Spring Boot Reference Documentation. <https://docs.spring.io/spring-boot> (último acceso: julio 2026).
- [4] Google — Flutter Documentation. <https://docs.flutter.dev> (último acceso: julio 2026).
- [5] PostgreSQL Global Development Group — PostgreSQL 15 Documentation. <https://www.postgresql.org/docs/15> (último acceso: julio 2026).
- [6] Mercado Pago — Documentación para Desarrolladores. <https://www.mercadopago.com.ar/developers> (último acceso: julio 2026).
- [7] Jones, M.; Bradley, J.; Sakimura, N. — JSON Web Token (JWT), RFC 7519 — IETF — 2015.
- [8] Anthropic — Claude [modelo de lenguaje]. <https://claude.ai> — utilizado como asistente de redacción y revisión de estilo (julio 2026).

## Datos de Contacto

Felipe Arias. UTN, Facultad Regional Córdoba.  
Maestro M. López esq. Cruz Roja Argentina,  
Córdoba. [felipe2arias@hotmail.com](mailto:felipe2arias@hotmail.com)  
Santiago Cantero. UTN, Facultad Regional Córdoba.  
[santicantero124@gmail.com](mailto:santicantero124@gmail.com)  
Lucas Perazolo. UTN, Facultad Regional Córdoba.  
[lucasperazolo@hotmail.com](mailto:lucasperazolo@hotmail.com)  
David Wunderlin. UTN, Facultad Regional Córdoba.  
[david.wunderlinok@gmail.com](mailto:david.wunderlinok@gmail.com)

## PLANILLA PARA CATALOGAR EL PROYECTO FINAL

|                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                              |           |
|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------|
| <b>AÑO</b>                                                                                                                 | 2026                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>CURSO Y NRO. DE GRUPO</b> | 5K3 - G01 |
| <b>NOMBRE DEL SISTEMA / PROYECTO</b>                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                              |           |
| Parko — Sistema Integral de Gestión de Estacionamiento para la Universidad Tecnológica Nacional, Facultad Regional Córdoba |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                              |           |
| <b>CATEGORIA (Solución De Negocio / Producto / Proyecto De Impacto Social)</b>                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                              |           |
| Solución de Negocio                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                              |           |
| <b>HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS</b>                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                              |           |
| <b>ÁMBITO DE APLICACIÓN</b>                                                                                                | <b>NOMBRE Y VERSIÓN</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                              |           |
| <b>ENTORNO DE DESARROLLO</b>                                                                                               | IntelliJ IDEA (backend Java / Spring Boot). Android Studio y Visual Studio Code (desarrollo Flutter). Flutter SDK 3.44.4, con Dart SDK incluido. Docker 29.6.2 y Docker Compose para el ambiente local. Ambientes definidos: desarrollo local, staging y producción on-premise en servidores de la UTN FRC.                                                                                                                                                                                                                                                                                                              |                              |           |
| <b>REPOSITORIOS Y VERSIONADO</b>                                                                                           | Git. GitHub — organización parko-utn (backend/, mobile/, docs/, assets/). Política de ramas: main protegida, develop de integración, feature/US-NNN y fix/US-NNN por historia. Versionado semántico (MAJOR.MINOR.PATCH). Pull Request con aprobación obligatoria de un integrante distinto del autor. GitHub Actions para CI/CD: build, tests y lint en cada PR. Checkstyle para control de estilo.                                                                                                                                                                                                                      |                              |           |
| <b>PROGRAMACIÓN</b>                                                                                                        | Backend: Java 21, Spring Boot 4.0.6, Spring Security, JPA / Hibernate, SLF4J con Logback. API REST versionada por prefijo de URL (/api/v1), documentada con OpenAPI 3 (Swagger UI). Autenticación con JSON Web Tokens (JWT). Mobile: Flutter 3.44.4 / Dart, plataforma objetivo Android.                                                                                                                                                                                                                                                                                                                                 |                              |           |
| <b>BASE DE DATOS</b>                                                                                                       | PostgreSQL 15 — implementado, ejecutado en contenedor Docker, con esquema generado desde las entidades JPA. Redis y RabbitMQ: previstos por arquitectura (ADR-004 y ADR-006), no implementados a la fecha.                                                                                                                                                                                                                                                                                                                                                                                                               |                              |           |
| <b>COMUNICACIÓN INTERNA</b>                                                                                                | Discord — servidor privado del equipo, dailies asincrónicos y reuniones semanales por videollamada. WhatsApp — coordinación rápida.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                              |           |
| <b>CAPACITACIÓN</b>                                                                                                        | Spikes técnicos documentados de Flutter y de Firebase (Sprint 3), con documento de hallazgos. Wiki del repositorio con el instructivo de setup del entorno Flutter. Documentación oficial de Flutter, Spring Boot y Firebase. Asistentes de IA generativa (GitHub Copilot, ChatGPT, Claude, Gemini) como mecanismo principal de aprendizaje del stack, conforme a la Política de Uso de Inteligencia Artificial definida en el Sprint 0, que exige revisión humana de todo artefacto generado. Capacitación de usuarios finales: prevista para la Instancia 5 (instructivos de uso y manual de instalación y operación). |                              |           |
| <b>PRUEBAS DE SISTEMA</b>                                                                                                  | Unitarias backend: JUnit 5 y Mockito. Unitarias y de widget mobile: flutter_test. Integración: Testcontainers. Extremo a extremo mobile: integration_test (Flutter). API y aceptación: Postman / Newman. Carga y concurrencia: k6. Seguridad: OWASP ZAP. Cobertura: JaCoCo, con mínimo del 60 % sobre el código nuevo de cada sprint y objetivo del 70 % en módulos críticos.                                                                                                                                                                                                                                            |                              |           |
| <b>GESTIÓN DEL PROYECTO</b>                                                                                                | Jira — backlog, sprints y seguimiento. Scrum adaptado al calendario académico, con sprints de dos semanas. Ceremonias: Sprint Planning, daily asincrónico, Sprint Review y Retrospectiva. Estimación con Planning Poker sobre escala de Fibonacci. Métricas: velocity, burndown, cobertura de pruebas y adherencia al cronograma.                                                                                                                                                                                                                                                                                        |                              |           |
| <b>DOCUMENTACIÓN</b>                                                                                                       | Google Drive — documentos de la cátedra. Microsoft Word y PDF. OpenAPI 3 / Swagger UI autogenerada desde las anotaciones de Spring. Figma — diseño y maquetado de interfaz. Repositorio GitHub: carpeta docs/ para ADRs en Markdown, DER y diagramas técnicos; wiki para instructivos de entorno.                                                                                                                                                                                                                                                                                                                        |                              |           |
| <b>MODELOS</b>                                                                                                             | Modelo C4 — nivel 1 de contexto y nivel 2 de contenedores. Diagrama Entidad-Relación (DER). Diagrama de clases del dominio. Mapa global de procesos del entorno. User Story Map con corte de MVP. Work Breakdown Structure (WBS). Wireframes y maquetado de UI en Figma. Architecture Decision Records: siete ADRs fundacionales.                                                                                                                                                                                                                                                                                        |                              |           |