



PROYECTO FINAL DE CARRERA

Sistema de Gestión Integral de Viajes

DOCENTES

Victoria Yamila Ahumada
María Natalia Jaime

AÑO
2026

CURSO
5K4

GRUPO
4

INTEGRANTES

Barbosa Marchi, Nicolás Daniel	94654
Cavalleri, Mateo	95980
Chiapero, Juan Ignacio	96457
Gallo Quinteros, Bruno	94267
Oría, Facundo	94595
Pasté Kvapil, Ignacio José	95802

REIS

Plataforma web y móvil para la
gestión centralizada de viajes



IDEA ORIGINAL

La planificación de un viaje implica información distribuida en múltiples apps, correos y conversaciones.

Reis centraliza todo en un único lugar, facilitando la organización y creando una memoria compartida del viaje.



OBJETIVOS

- ✓ Centralizar itinerarios, alojamientos, vuelos, actividades y documentación.
- ✓ Facilitar la planificación individual y colaborativa.
- ✓ Permitir acceso a información actualizada y compartida.
- ✓ Reducir la pérdida de información y el uso de múltiples aplicaciones.



DESCRIPCIÓN DEL TRABAJO



Reis es una plataforma web que permite organizar y vivir cada viaje de forma colaborativa.



Creá viajes, invitá participantes y definí ciudades, fechas y alojamientos.



Descubrí y guardá lugares de interés en cada destino.



Compartí fotos asociadas al viaje, a cada ciudad o a cada lugar de interés.

HERRAMIENTAS Y TECNOLOGÍAS



React



TypeScript



Python



Django



PostgreSQL



Celery



Docker



Cloudinary



OpenStreetMap



GitHub



Jira



Figma



ChatGPT



Claude



Drive



Discord

RESULTADOS ESPERADOS Y BENEFICIOS

- ★ Menor tiempo dedicado a buscar información.
- ★ Reducción de documentos y reservas dispersas.
- ★ Acceso rápido a la documentación durante el viaje.
- ★ Mayor coordinación entre los integrantes.
- ★ Memoria visual compartida organizada por viaje, ciudad y lugar.

RESUMEN

Reis reúne toda la información y los recuerdos de cada viaje en un único entorno accesible para todos sus participantes.

Organizá. Viví. Recordá.



AUTORES

- Barbosa Marchi, Nicolas Daniel nicolasbarbosa250@gmail.com
- Cavalleri, Mateo gromateoca@gmail.com
- Chiapero, Juan Ignacio juanichiapero@gmail.com
- Gallo Quinteros, Bruno brunogallocat@gmail.com
- Oria, Facundo facuoria2003@gmail.com
- Pasté Kvapil, Ignacio José ignapaste@gmail.com

DOCENTES

- Victoria Ahumada
- Maria Natalia Jaime

Gracias a quienes
nos acompañaron
en este camino.



AVENTURAS
SE COMPARTEN
★ RECUERDOS ★
SE CONSERVAN

Reis

Plataforma web y móvil para la gestión centralizada de viajes

**Barbosa Marchi, Nicolas - Cavalleri, Mateo - Chiapero, Juan Ignacio -
Gallo Quinteros, Bruno - Oría, Facundo - Pasté Kvapil, Ignacio José**

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

La planificación de un viaje requiere administrar información distribuida entre mapas, plataformas de reserva, notas, fotografías y aplicaciones independientes. Esta fragmentación dificulta la consulta de datos y la construcción de una memoria compartida del recorrido. Para abordar esta problemática, se desarrolló Reis, una plataforma web y móvil orientada a centralizar la organización y el seguimiento de viajes colaborativos. La solución permitió crear viajes, registrar ciudades, gestionar alojamientos e incorporar lugares de interés dentro de cada destino. Además, se implementó una galería compartida con tres niveles de granularidad: fotografías generales del viaje, fotografías asociadas a una ciudad y fotografías vinculadas a un lugar de interés. Como resultado, se obtuvo un prototipo funcional y modular que reúne información, lugares y recuerdos de cada viaje en un único entorno accesible para todos sus participantes.

Palabras Clave

Producto. Gestión de viajes, plataforma web y móvil, viajes colaborativos, lugares de interés, galería compartida, React Native, React, Django.

Introducción

La organización de un viaje implica reunir y consultar información proveniente de diferentes fuentes. Los destinos, alojamientos, actividades, ubicaciones y fotografías suelen distribuirse entre mapas, notas, plataformas especializadas y conversaciones grupales. Esta dispersión obliga a los viajeros a alternar entre múltiples herramientas y dificulta que todos los participantes dispongan de una visión común del recorrido.

Reis surge como respuesta a esta problemática mediante una plataforma web y móvil que centraliza la información asociada a un viaje. La plataforma permite crear viajes colaborativos, incorporar participantes, definir ciudades y fechas, organizar alojamientos y registrar lugares de interés dentro de cada destino, como museos, monumentos, parques o restaurantes.

El producto también permite construir una memoria visual compartida. Las fotografías pueden asociarse al viaje en general, a una ciudad específica o a un lugar de interés, manteniendo distintos niveles de detalle sin perder su pertenencia al mismo recorrido. Una vez publicadas, las imágenes quedan disponibles para los demás participantes. El objetivo del proyecto es simplificar la planificación, reducir la fragmentación de datos y ofrecer un espacio común para organizar y conservar la experiencia del viaje.

Elementos del Trabajo y Metodología

El proyecto se desarrolló mediante una metodología ágil basada en Scrum [1], organizada en iteraciones de tres semanas. Cada sprint comenzó con la planificación y priorización de historias de usuario, continuó con el seguimiento periódico de tareas y finalizó con instancias de revisión y retrospectiva. Este enfoque permitió incorporar cambios de alcance, redefinir funcionalidades y mantener entregas incrementales durante la evolución del producto.

La gestión del trabajo se realizó con Jira [2], donde se organizaron épicas, historias de usuario, tareas técnicas y defectos. La documentación del proyecto se almacenó en Google Drive [3], mientras que Discord se utilizó como principal canal de comunicación del equipo. Para la exploración de ideas se empleó Mural y las interfaces se diseñaron con Figma [4]. El código fuente se administró mediante Git y GitHub [5], favoreciendo la colaboración y la trazabilidad de los cambios. Como apoyo al desarrollo, el equipo incorporó asistentes de inteligencia artificial para programación, como Codex [12] y Claude Code [13], que agilizaron tareas de generación de código, depuración y documentación técnica.

El desarrollo de los clientes se dividió en dos frentes: la aplicación móvil, construida con React Native [6] para mantener una única base de código entre Android e iOS, y la interfaz web, que amplió el acceso al producto desde el navegador. Ambos consumieron una API REST común, implementada con Python, Django y Django Ninja [7], responsable de validar solicitudes, aplicar las reglas de negocio y administrar la persistencia de la información.

Para ejecutar procesos asincrónicos se incorporó Celery [8], permitiendo encolar tareas y desacoplar operaciones que no requieren una respuesta inmediata. El entorno del backend se organizó con Docker Compose [9], que facilitó la configuración y el levantamiento coordinado de los servicios necesarios para el desarrollo.

Las fotografías se gestionaron mediante Cloudinary [10]. El modelo de dominio contempló tres niveles de asociación: viaje, ciudad y lugar de interés. Esta jerarquía permitió consultar imágenes generales del recorrido o acceder a colecciones más específicas, manteniendo la relación con el mismo viaje y compartiendo el contenido entre sus participantes. Para la búsqueda de ciudades se utilizaron servicios basados en OpenStreetMap [11], mientras que Leaflet [14] se empleó como biblioteca para la visualización e interacción con la información geográfica dentro de la plataforma.

Resultados

Como resultado del desarrollo se obtuvo un prototipo funcional de Reis capaz de administrar la información principal de un viaje colaborativo. El sistema permitió crear viajes, incorporar participantes, establecer fechas, registrar ciudades de destino y organizar los alojamientos correspondientes a cada etapa.

Se incorporó la gestión de lugares de interés dentro de cada ciudad. Los usuarios pudieron registrar puntos relevantes del destino, como el Museo del Prado en una visita a Madrid, y conservarlos como parte de la planificación y del recorrido. Esta funcionalidad permitió representar el viaje con mayor nivel de detalle que una organización basada únicamente en ciudades y alojamientos.

También se implementó una galería compartida con tres niveles de granularidad. Las fotografías pudieron vincularse al viaje completo, a una ciudad determinada o a un lugar de interés específico. Todas conservaron su pertenencia al viaje y quedaron disponibles para los demás participantes, permitiendo construir de forma colaborativa un registro visual organizado.

La separación entre las interfaces web y móvil, la API, las tareas asincrónicas y los servicios externos permitió mantener responsabilidades diferenciadas. La aplicación móvil y la plataforma web concentraron la interacción con el usuario; Django y Django Ninja administraron las reglas de negocio; Celery gestionó procesos encolados; Docker Compose simplificó el entorno de ejecución; y Cloudinary almacenó los recursos multimedia.

Discusión

Reis centraliza información que habitualmente permanece distribuida entre conversaciones, mapas, notas, plataformas de reserva y galerías personales. Esta concentración reduce la cantidad de herramientas que los viajeros deben consultar y mejora la disponibilidad de la información para todos los participantes.

La incorporación de lugares de interés amplía el alcance del producto, ya que permite organizar no solo las etapas generales del viaje, sino también los puntos concretos que conforman la experiencia en cada ciudad. A su vez, la jerarquía de fotografías aporta una estructura que combina una visión global del recorrido con colecciones específicas por destino y actividad.

El carácter compartido de la galería favorece la colaboración entre participantes y evita que los recuerdos queden fragmentados en dispositivos individuales. Sin embargo, esta funcionalidad requiere continuar trabajando sobre aspectos como permisos, eliminación de contenido, moderación y experiencia de consulta cuando la cantidad de imágenes aumenta.

El prototipo cumple el objetivo de concentrar viajes, ciudades, alojamientos, lugares de interés y fotografías compartidas. No obstante, todavía presenta oportunidades de mejora relacionadas con el acceso sin conexión, la integración con proveedores externos, las notificaciones contextuales y la validación del producto con una muestra más amplia de usuarios.

Conclusión

Reis constituye una solución web y móvil orientada a simplificar la planificación y el seguimiento de viajes mediante la centralización de ciudades, alojamientos, lugares de interés y fotografías. El producto ofrece un espacio colaborativo en el que los participantes pueden consultar la información del recorrido y construir una memoria visual compartida.

La jerarquía de imágenes a nivel de viaje, ciudad y lugar de interés representa uno de los principales aportes de la solución, ya que permite organizar los recuerdos con distintos niveles de granularidad sin perder la relación con el viaje principal. La arquitectura implementada proporciona una base modular para incorporar nuevas funcionalidades y escalar los componentes de manera independiente.

Como trabajo futuro se plantea incorporar vuelos y actividades, mejorar la gestión de permisos sobre el contenido compartido, habilitar el acceso sin conexión, sumar notificaciones contextuales e integrar recomendaciones asociadas a cada destino. También resulta necesario realizar pruebas de usabilidad y definir métricas cuantitativas que permitan evaluar la reducción del tiempo de organización y la satisfacción de los viajeros.

Agradecimientos

Se agradece a la cátedra de Proyecto Final, a los docentes y tutores que acompañan el desarrollo de Reis, y a las personas que participaron en las instancias de relevamiento y validación del producto.

Referencias

- [1] Scrum.org, “The Scrum Guide.” Disponible en: <https://scrumguides.org/>. Último acceso: ago. 2026.
- [2] Atlassian, “Jira Software.” Disponible en: <https://www.atlassian.com/software/jira>. Último acceso: ago. 2026.
- [3] Google, “Google Drive.” Disponible en: <https://workspace.google.com/products/drive/>. Último acceso: ago. 2026.
- [4] Figma, “Figma Design.” Disponible en: <https://www.figma.com/>. Último acceso: ago. 2026.
- [5] GitHub, “GitHub Documentation.” Disponible en: <https://docs.github.com/>. Último acceso: ago. 2026.
- [6] Meta Open Source, “React Native Documentation.” Disponible en: <https://reactnative.dev/>. Último acceso: ago. 2026.
- [7] Django Software Foundation, “Django Documentation”; Django Ninja, “Django Ninja Documentation.” Disponible en: <https://www.djangoproject.com/> y <https://django-ninja.dev/>. Último acceso: ago. 2026.
- [8] Celery Project, “Celery Documentation.” Disponible en: <https://docs.celeryq.dev/>. Último acceso: ago. 2026.
- [9] Docker, “Docker Compose Documentation.” Disponible en: <https://docs.docker.com/compose/>. Último acceso: ago. 2026.
- [10] Cloudinary, “Image and Video API Platform.” Disponible en: <https://cloudinary.com/>. Último acceso: ago. 2026.
- [11] OpenStreetMap Foundation, “OpenStreetMap.” Disponible en: <https://www.openstreetmap.org/>. Último acceso: ago. 2026.
- [12] OpenAI, “Codex.” Disponible en: <https://chatgpt.com/codex/>. Último acceso: ago. 2026.
- [13] Anthropic, “Claude Code.” Disponible en: <https://www.anthropic.com/claude-code>. Último acceso: ago. 2026.
- [14] Leaflet, “Leaflet — an open-source JavaScript library for mobile-friendly interactive maps.” Disponible en: <https://leafletjs.com/>. Último acceso: ago. 2026.

Datos de Contacto

Barbosa Marchi, Nicolas Daniel - Legajo 94654 - nicolasbarbosa250@gmail.com
Cavalleri, Mateo - Legajo 95980 - gromateoca@gmail.com
Chiapero, Juan Ignacio - Legajo 96457 - juanichiapero@gmail.com
Gallo Quinteros, Bruno - Legajo 94267 - brunogallocat@gmail.com
Oría, Facundo - Legajo 94595 - facuoria2003@gmail.com
Pasté Kvapil, Ignacio José - Legajo 95802 - ignapaste@gmail.com

PLANILLA PARA CATALOGAR EL PROYECTO FINAL

AÑO	2026	CURSO Y NRO DE GRUPO	5K4 - G4
NOMBRE DEL SISTEMA / PROYECTO			
REIS			
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 [1.132.0]	
REPOSITORIOS Y VERSIONADO		Git [2.55.0], GitHub [3.6.3]	
PROGRAMACIÓN		Python [3.11], TypeScript [6.0.2], React [19.2.6], React Native [19.1.0], Django [5.1.5], Cloudinary Python SDK [1.45.0]	
BASE DE DATOS		PostgreSQL [18.6]	
COMUNICACIÓN INTERNA		Discord, WhatsApp	
CAPACITACIÓN		Youtube, Documentación de Tecnologías	
PRUEBAS DE SISTEMA		Testing Manual, Postman [12.17], Pytest [9.1.1], Jest [30.4.2]	
GESTIÓN DEL PROYECTO		Jira [11.3.8]	
DOCUMENTACIÓN		Google Drive [2.25.280], GitHub [3.6.3]	
MODELOS		Mermaid web, Figma	