



Instituto Tecnológico de Buenos Aires

72.40 - Ingeniería del Software II

White Paper

Comisión: S2

María Agostina Squillari

Pilar Frutos

Profesor: Emilio Esteban Rasic

Fecha: 13 de noviembre de 2025

1. Introducción.....	2
2. Definición estratégica de producto.....	2
3. Alcance funcional de alto nivel.....	2
4. Atributos de calidad priorizados.....	3
4.1. Usabilidad.....	3
4.2. Rendimiento.....	4
4.3. Disponibilidad.....	4
4.4. Seguridad.....	4
4.5. Mantenibilidad.....	5
5. Arquitectura del sistema.....	5
5.1 Vista Física.....	6
5.2 Vista de Procesos.....	6
6. Eje Técnico.....	7
6.1 Descripción técnica.....	7
6.2 Implementación en PlaneIt (PoC).....	7
7. Eje Funcional.....	8
7.1 Utilidad del eje técnico.....	8
7.2 Comparativa tecnológica.....	8
8. Conclusión.....	8
9. Anexo.....	9
Diagrama 1.....	9
Diagrama 2.....	9

1. Introducción

Organizar viajes en grupo suele ser caótico por la dispersión de información en distintas plataformas. PlaneIt propone una solución centralizada que integra itinerarios, documentos, reservas y mapas en un único espacio colaborativo. Así, mejora la coordinación entre los participantes, distribuye responsabilidades y permite la organización en tiempo real gracias a su arquitectura técnica e integraciones con servicios externos como mapas y vuelos.

2. Definición estratégica de producto

PlaneIt busca convertir la planificación grupal en una experiencia organizada y compartida, centralizando toda la información del viaje en una sola plataforma. Se apoya en alianzas con Google Maps, Skyscanner y Tripadvisor, entre otros, para ofrecer datos útiles y reservas integradas.

Su modelo de negocio combina versión gratuita con publicidad, planes premium y comisiones por reservas. Está dirigido a grupos de estudiantes, amigos, familias, instituciones y empresas. La relación con los usuarios se fortalece mediante soporte digital, encuestas y presencia en redes y tiendas de apps.

3. Alcance funcional de alto nivel

PlaneIt ofrece un conjunto de funcionalidades clave que sostienen su propuesta de valor:

- Gestión integrada de itinerarios y eventos, mediante un calendario compartido con vuelos, check-ins, excursiones y actividades.
- Administración centralizada de documentos, que reúne tickets, pasajes y reservas en un único espacio accesible.
- Visualización geográfica de destinos y actividades, mediante la integración con mapas interactivos.
- Colaboración en tiempo real, permitiendo que todos los integrantes propongan, editen o confirmen actividades.
- Recordatorios y notificaciones inteligentes, que mantienen visibles las fechas y la información crítica.

De este modo, PlaneIt aporta orden, eficiencia y confianza a la planificación grupal, mejorando la experiencia de viaje desde la organización hasta la ejecución.

4. Atributos de calidad priorizados

Esta sección describe los principales atributos de calidad que guían el diseño y la evolución de PlaneIt. Cada uno de ellos, usabilidad, rendimiento, disponibilidad, seguridad y mantenibilidad, define aspectos clave para garantizar que la plataforma sea intuitiva, ágil, confiable y sostenible a medida que crece en funcionalidades y usuarios.

- Calidad General del Sistema
 - Atributos de Usuario
 - Usabilidad
 - Atributos de Funcionamiento
 - Rendimiento
 - Disponibilidad
 - Seguridad

- Atributos de Diseño y Sistema
 - Mantenibilidad
- Atributos de Negocio
 - Escalabilidad

4.1. Usabilidad

PlaneIt ofrece una experiencia clara y simple para todo tipo de usuarios, con una interfaz visual basada en calendario, autenticación sencilla y retroalimentación inmediata. Próximamente incorporará un onboarding guiado, diseño móvil-first, notificaciones contextuales y mejoras de accesibilidad y coherencia multiplataforma.

4.2. Rendimiento

El sistema prioriza la fluidez mediante sincronización periódica, consultas optimizadas y notificaciones en tiempo real. Se prevé mejorar con WebSockets para sincronización instantánea, carga diferida, cacheo local, uso de CDN y compresión de recursos para reducir la latencia.

4.3. Disponibilidad

Gracias a su arquitectura contenerizada y servicios independientes, PlaneIt mantiene estabilidad ante fallos. En el futuro incorporará modo offline, sincronización automática, infraestructura distribuida en la nube, réplicas de base de datos, monitoreo activo y backups automáticos para garantizar alta disponibilidad.

4.4. Seguridad

Actualmente protege datos con hashing de contraseñas, tokens seguros y validaciones estrictas. Las próximas mejoras incluyen cifrado AES-256, TLS 1.3, roles y permisos, autenticación multifactor, auditoría de acciones, rate limiting, gestión de sesiones segura y administración centralizada de secretos.

4.5. Mantenibilidad

El proyecto presenta una estructura modular con separación de responsabilidades, dockerización y configuración flexible. Se planea implementar arquitectura hexagonal, tests automatizados, documentación API, migración a TypeScript, CI/CD, logging estructurado, manejo centralizado de errores y versionado del API, asegurando evolución sostenible y bajo mantenimiento.

5. Arquitectura del sistema

La arquitectura de PlaneIt se estructura según el Modelo 4+1, que permite describir el sistema desde diferentes perspectivas complementarias para garantizar el cumplimiento de los atributos de calidad priorizados: usabilidad, disponibilidad y mantenibilidad.

El diseño combina tres estilos arquitectónicos fundamentales:

- Dataflow, aplicado al procesamiento y transformación secuencial de datos, así como en la integración y sincronización de itinerarios y documentos.
- Sistemas distribuidos, que habilitan escalabilidad y resiliencia en un entorno cloud mediante la ejecución de componentes en múltiples nodos.
- Event-based, que sostiene la colaboración en tiempo real al disparar notificaciones y actualizaciones asincrónicas ante acciones de los usuarios.

Estos estilos se integran de manera híbrida y complementaria: el enfoque dataflow gestiona los flujos de datos estructurados (por ejemplo, la carga de reservas desde APIs externas); la

arquitectura distribuida asegura redundancia geográfica y tolerancia a fallos; y el modelo event-based garantiza baja latencia en las interacciones grupales mediante un bus de eventos centralizado.

Para documentar la solución, se seleccionaron dos vistas principales del Modelo 4+1:

- La Vista Física, que representa la infraestructura de despliegue —hardware, red y servicios—, alineada con el estilo de sistemas distribuidos para maximizar la disponibilidad.
- La Vista de Procesos, que muestra los flujos dinámicos y concurrentes entre componentes, integrando los enfoques event-based y dataflow para ilustrar la colaboración en tiempo real y el procesamiento de información.

Cada vista se complementa con diagramas representativos que evidencian la relación entre los componentes y los estilos arquitectónicos adoptados.

5.1 Vista Física

Esta vista describe cómo se despliegan los componentes del sistema en nodos físicos o virtuales, enfatizando la distribución para soportar alta disponibilidad y rendimiento. PlaneIt utiliza una arquitectura distribuida en la nube (basada en Supabase y servicios externos como Google Cloud o AWS), con redundancia multi-región para tolerar fallos. El frontend se conecta a un API Gateway que integra llamadas a microservicios backend, como el de autenticación, itinerarios y notificaciones. La base de datos PostgreSQL (Supabase) maneja persistencia con replicación en tiempo real, mientras que integraciones externas (Google Maps, Skyscanner) se acceden vía adaptadores para mantener el desacoplamiento. Este diseño incorpora el estilo de sistemas distribuidos, permitiendo escalabilidad horizontal (agregar nodos bajo demanda) y resiliencia, alineado con atributos como disponibilidad y rendimiento.

El diagrama 1 del anexo muestra la distribución de componentes, con flechas indicando flujos de datos y eventos, destacando cómo el estilo distribuido asegura redundancia (réplicas) y el event-based maneja notificaciones asincrónicas.

5.2 Vista de Procesos

Esta vista enfoca los procesos concurrentes y flujos de ejecución, ilustrando cómo el sistema maneja interacciones en tiempo real y procesamiento de datos. En PlaneIt, los procesos clave incluyen la edición colaborativa de itinerarios (donde eventos de usuarios disparan actualizaciones) y el procesamiento de documentos/reservas (un flujo secuencial de validación e integración). Se incorpora el estilo event-based para manejar eventos asincrónicos (ej. un usuario agrega una actividad, generando un evento que notifica al grupo vía bus), combinado con dataflow para transformar datos entrantes (ej. parseo de reservas de Skyscanner a formato interno). Esto soporta atributos como rendimiento (sincronización < 500ms) y mantenibilidad (procesos modulares fáciles de extender). La vista resalta la concurrencia: múltiples usuarios interactúan simultáneamente sin bloqueos, con threads para background tasks como sincronización offline.

El diagrama 2 del anexo ilustra el flujo event-based (disparo de eventos para notificaciones) y dataflow (transformación secuencial en backend), asegurando procesos eficientes y escalables para escenarios grupales.

6. Eje Técnico

El desarrollo de PlaneIt requiere automatizar flujos y coordinar eventos entre servicios, garantizando una experiencia colaborativa en tiempo real. Para resolver esta necesidad se incorporó N8N, una herramienta de automatización basada en eventos (event-driven workflows) que permite

orquestar procesos de forma visual y modular. Su enfoque bajo-código facilita integrar APIs, manejar tareas asíncronas y ejecutar acciones automáticas sin sobrecargar el backend.

6.1 Descripción técnica

N8N es una plataforma de automatización open source que permite construir flujos de trabajo mediante nodos conectados entre sí. Cada nodo representa una acción. Los flujos pueden dispararse por eventos y ejecutar pasos de procesamiento o comunicación entre servicios.

Se ejecuta como un servicio stateless con persistencia en base de datos, generalmente Postgres, y puede desplegarse fácilmente con Docker. Los flujos se guardan como definiciones JSON versionables, lo que permite su integración continua y portabilidad. Gracias a su interfaz visual, los equipos técnicos y funcionales pueden definir procesos sin necesidad de programar complejas colas o pipelines.

6.2 Implementación en PlaneIt (PoC)

La prueba de concepto se centró en implementar un sistema de notificaciones automáticas para eventos próximos en una aplicación de gestión de calendarios de viaje colaborativa. El caso de uso específico elegido fue: "Como usuario, quiero recibir una notificación automática cuando un evento de mi calendario esté próximo a comenzar (dentro de 1 hora), para poder prepararme adecuadamente".

Esta historia de usuario se desarrolla en un entorno colaborativo, donde distintos usuarios pueden crear sus propios calendarios de viaje, solicitar acceso a los de otros y gestionar eventos de forma compartida. El principal desafío fue lograr que el sistema detecte automáticamente los eventos próximos y envíe notificaciones a los participantes correspondientes, evitando la necesidad de consultas manuales o actualizaciones constantes desde el cliente.

7. Eje Funcional

7.1 Utilidad del eje técnico

La integración de n8n en la arquitectura de PlaneIt demuestra cómo las decisiones técnicas fortalecen la propuesta de valor del producto. La automatización de notificaciones basada en eventos mejora la coordinación entre los usuarios y refuerza atributos clave como usabilidad, disponibilidad y mantenibilidad.

El sistema genera recordatorios de forma automática y distribuye la información en tiempo real, evitando tareas manuales y optimizando la experiencia de uso. Además, al desacoplar la lógica de automatización del backend, n8n aporta flexibilidad y escalabilidad, permitiendo modificar flujos sin afectar los servicios centrales.

7.2 Comparativa tecnológica

N8N vs Kafka: Kafka ofrece alta escalabilidad y persistencia de eventos, pero requiere una infraestructura más compleja (brokers, topics, consumers). n8n permite resultados rápidos, visuales y de bajo mantenimiento, adecuados para una PoC funcional.

N8N vs RabbitMQ: RabbitMQ maneja colas de mensajes de forma eficiente, aunque sin interfaz visual ni conectores predefinidos. n8n simplifica la integración con servicios externos y reduce el desarrollo manual.

N8N vs Airflow/Prefect: Estas herramientas se orientan al batch processing de datos, mientras que n8n está pensado para procesos continuos y reactivos, más coherentes con la colaboración en tiempo real que caracteriza a PlaneIt.

En conclusión, n8n equilibra simplicidad, flexibilidad y coherencia con la arquitectura de PlaneIt, permitiendo demostrar de manera práctica cómo la automatización de eventos potencia la calidad y disponibilidad del sistema.

8. Conclusión

PlaneIt busca ser la herramienta práctica para cualquiera que organice un viaje en grupo: un espacio único donde toda la información se concentra, evitando pérdidas de tiempo entre chats, correos y carpetas. Así, la organización se simplifica y el viaje se disfruta más.

9. Anexo

Diagrama 1

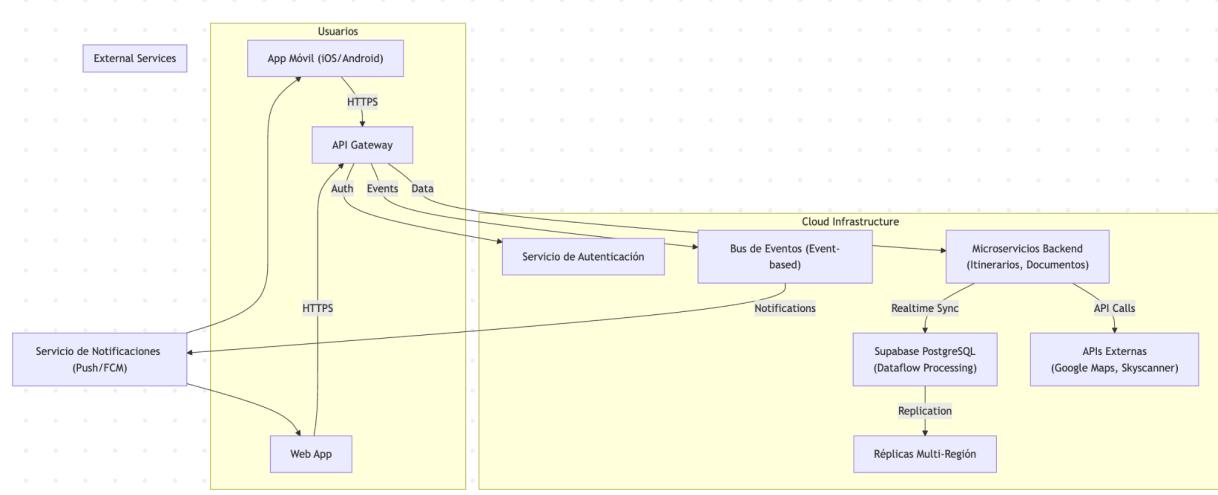


Diagrama 2

