

## ✓ **Requisitos Funcionales (qué debe hacer la app)**

### ? **Gestión de Fechas**

- Selector de fechas editable en cualquier momento
- Posibilidad de agregar/quitar días (solo al final de la lista)
- Contador de días restantes para el viaje

### ? **Gestión por Día**

- Cards con tamaño fijo y scroll interno (responsive)
- Cada día puede tener:
  - Comidas: desayuno, almuerzo, merienda, cena
  - Actividades (opcionales)
- Las actividades/comidas deben ordenarse automáticamente por horario

### ? **Actividades**

- Cada actividad es clickeable y abre un modal con:
  - Mapa embebido (iframe con URL insertable al crearla)
  - Mapa dinámico (drag, zoom)
  - Botón para abrir en Google Maps
- Detección de URL inválidas (avisar si el iframe falla)
- Puntaje y cantidad de reseñas del lugar (si las tiene en Google)

### ✓ **Todo List**

- Las actividades pueden marcarse como completadas (checkbox)
- Las completadas se deshabilitan visualmente, pero el checkbox sigue activo

### ? **Visual**

- Modo light/dark con botón pequeño
- Portada personalizada + nombre del viaje

### ? **Organización y Planificación**

- Calendario para selección de fechas
- Contador de progreso por actividades
- Checklist de equipaje editable
- Trazado de rutas entre actividades del día

## **Notas y Archivos**

- Notas personales por actividad o por día
- Subir imágenes/PDFs (tickets, reservas, etc.)

## **? Usuarios e Integrantes**

- Página de login y registro
- Edición de perfil y foto
- Agregar otros usuarios al viaje
- Chat o comentarios entre integrantes del viaje

## **? Compartir**

- Generar link de planificación para personas sin cuenta

## **? Requisitos Técnicos / No Funcionales**

### **? Arquitectura y Código**

- Frontend: React con TypeScript
- Backend: Node.js + Express
- Base de datos: MongoDB Atlas
- Arquitectura del código: Screaming Architecture
- Separar los estilos en archivos CSS externos
- Tests unitarios con Jest y Testing Library
- Documentación muy completa
- Manual de usuario claro y detallado

### **? Seguridad**

- Login seguro (incluye prácticas robustas de ciberseguridad)
- Implementación de OAuth (Google/Facebook)
- Recuperación de contraseña por email
- Protección contra ataques comunes:
  - XSS (Cross Site Scripting)
  - CSRF (Cross Site Request Forgery)
  - Expiración de JWT
  - Rate limiting
- Hashing y salting de contraseñas con bcrypt

## ? Roles

- Cuenta de administrador (tuya)
- Posibilidad de agregar cuentas de usuario

## Panel de Administración

- Ver todos los viajes creados
- Eliminar o editar usuarios
- Ver estadísticas de uso

## ? Onboarding / Tutoriales

- Tutorial interactivo la primera vez que se usa
- Guía de onboarding con pasos explicados y ejemplos

## ? UX Extra

- Soporte para pantalla táctil (gestos, swipe entre días)
- Diseño moderno, minimalista y profesional

## Detalles y Requisitos Omitidos/No Detallados

- **Presupuesto diario:** Si bien mencionás "Presupuesto diario: agregar gastos por día y categoría" en tus necesidades, esta funcionalidad no se ve reflejada explícitamente en los **Requisitos Funcionales** detallados. Es un aspecto clave para la organización de viajes.
- **Diseño:** Aunque se menciona "Diseño moderno, minimalista y profesional", y que debe parecer hecho por un "diseñador profesional", las especificaciones no incluyen ningún detalle sobre la **interfaz de usuario (UI)** o la **experiencia de usuario (UX)** más allá de la alternancia de modo claro/oscuro y el soporte táctil. Un buen diseño profesional va más allá de un "look" y se centra en la usabilidad y la intuitividad.
- **Manejo de errores generales (adicional a URL inválidas):** Si bien se especifica la detección de URL inválidas para los iframes, no hay una mención general sobre el **manejo de errores** en la aplicación (por ejemplo, errores de red, errores del servidor, entradas de usuario inválidas en otros campos, etc.). Una aplicación profesional debe gestionar estos casos para ofrecer una experiencia robusta.
- **Notificaciones:** En un contexto de colaboración y chat, no se menciona si habrá **notificaciones** (por ejemplo, cuando alguien agrega un comentario, modifica una actividad,

etc.).

- **Versionado/Control de cambios:** Para una planificación colaborativa, no se especifica si habrá un **historial de cambios** o **versionado** de las planificaciones, lo cual podría ser útil para revertir errores o ver quién hizo qué modificación.
- **Internacionalización/Localización:** No se menciona si la aplicación debe soportar **múltiples idiomas** o **formatos de fecha/hora** según la ubicación del usuario.
- **Accesibilidad:** Más allá del diseño y el soporte táctil, no se especifican requisitos de **accesibilidad** (por ejemplo, compatibilidad con lectores de pantalla, navegación por teclado, contraste de colores).
- **Optimización del rendimiento:** Aunque se menciona la base de datos y la arquitectura, no hay requisitos explícitos sobre la **optimización del rendimiento** de la aplicación (por ejemplo, tiempos de carga, latencia de las solicitudes).
- **Escalabilidad:** Se menciona MongoDB Atlas, que sugiere escalabilidad, pero no hay requisitos explícitos sobre la **capacidad de la aplicación para manejar un gran número de usuarios o viajes**.
- **Monitoreo y logging:** Para una aplicación en producción, no se mencionan requisitos sobre **monitoreo** (observar el rendimiento y el uso) y **logging** (registro de eventos y errores para depuración y auditoría).
- **Copia de seguridad y recuperación:** No se especifica si habrá una estrategia de **copia de seguridad y recuperación de datos** para la base de datos.
- **Despliegue (Deployment):** No se menciona cómo se desplegará la aplicación o si hay requisitos específicos de entorno de despliegue.