📘 Bitácora del Proyecto EasyFlight App

Este documento recopila el desarrollo completo del proyecto EasyFlight App, realizado en 4 sprints siguiendo metodologías ágiles. Se detallan las funcionalidades, diseño, paletas de colores, decisiones técnicas, problemas encontrados y sus soluciones.

# 🚀 Sprint 1: Configuración inicial y registro/login

En este sprint se realizaron las configuraciones iniciales del proyecto tanto en el frontend (React) como en el backend (Spring Boot). Se implementaron los siguientes aspectos:  
- Configuración de React con rutas iniciales (React Router).  
- Creación del backend con Spring Boot 3.5 y configuración de H2 para pruebas.  
- Implementación de entidades iniciales: User, Category, Product.  
- Desarrollo de endpoints básicos para el registro y login de usuarios.  
- Envío de correos de confirmación de registro usando Spring Mail.

🎨 Paleta de colores inicial: Azul turquesa (#187b7b) y tonos claros (#cdeeee, #e0f6f6).

# 🛠️ Sprint 2: Gestión de productos y categorías

Se amplió la funcionalidad para permitir la gestión completa de productos y categorías:  
- CRUD de categorías desde el panel administrador.  
- CRUD de productos con carga de imágenes en el backend.  
- Implementación del componente ProductList.jsx en el frontend.  
- Filtros por categorías y barra de búsqueda.  
- Integración de favoritos para que el usuario pueda marcar productos con un corazón.  
- Diseño responsive con cuadrículas de productos.

# 📦 Sprint 3: Funcionalidades avanzadas del catálogo

En este sprint se desarrollaron características más avanzadas:  
- Sistema de reseñas con puntuaciones dinámicas.  
- Políticas de los productos en secciones bien diseñadas.  
- Funcionalidad de compartir productos en redes sociales.  
- Implementación de un sistema de favoritos persistente conectado al backend.  
- Mejoras en la interfaz con botones estilizados y diseño más atractivo.

# 📅 Sprint 4: Reservas y gestión de disponibilidad

Este sprint se enfocó en la funcionalidad de reservas:  
- Implementación del calendario en el frontend con selección de rangos de fechas.  
- Validación de disponibilidad desde el backend.  
- Creación de reservas vinculadas a productos y usuarios.  
- Envío de correos electrónicos con confirmación de reserva y detalles.  
- Visualización del historial de reservas en la sección 'Mis reservas'.  
- Implementación de un botón de contacto vía WhatsApp en el frontend.

📌 Se añadieron campos opcionales en la reserva como comentarios y número de pasajeros.

# 🎨 Diseño y estilos

La aplicación mantiene una estética moderna y limpia:  
- Colores principales: #187b7b (turquesa), #cdeeee (celeste claro), #e0f6f6 (fondo claro).  
- Tipografía: Segoe UI / sans-serif.  
- Uso de tarjetas con bordes redondeados y sombras suaves.  
- Botones con degradados y efectos hover.

# 🐞 Problemas encontrados y soluciones

- Problemas con CORS: Solucionado con configuración en WebConfig.  
- Error en rutas de React: Se corrigieron usando useNavigate y rutas dinámicas.  
- Problemas de compatibilidad con SweetAlert2: Solucionado importando estilos.  
- Bucle en consultas de reservas: Ajuste en los DTOs para evitar referencias circulares.

# 🖼️ Capturas de pantalla

A continuación se presentan ejemplos de pantallas clave:

1. Página principal con listado de productos.

2. Detalle de producto con reseñas y botón de reserva.

3. Pantalla de confirmación de reservas con datos del usuario.

4. Historial de reservas accesible desde el menú de usuario.

# ✅ Conclusiones

El proyecto EasyFlight App ha logrado cumplir con todos los criterios definidos en los cuatro sprints, ofreciendo una solución completa para la reserva de vuelos y paquetes turísticos. La aplicación combina un backend robusto en Spring Boot con un frontend moderno en React, asegurando una experiencia de usuario intuitiva y atractiva.