

Universidad Don Bosco

Tema del proyecto: Sistema de gestión de boletos aéreos para un aeropuerto.

Integrantes:

Jonathan Josue Segura Ramirez SR230847 **Rol:** Tester & DevOps

Juan Diego Peña Vivas PV230210 **Rol:** Líder & Arquitecto

Jorge Armando Hernandez Gonzalez HG242654 **Rol** Desarrollador Frontend

Andres Eduardo Navidad NF220677 **Rol:** Desarrollador Backend & Documentador

Fecha de entrega: 30/08/2025

Docente: Mario Alvarado

Asignatura: Desarrollo de Aplicaciones con Web Frameworks DWF404 G01T

Índice:

3.	Descripción del Proyecto	. 3
3.	1 Objetivo General	. 3
3.	2 Objetivo Específico	. 3
4.	Gestión Integral del Proyecto	. 3
	4.1 Gestión de la Integración	. 3
	4.2 Gestión del Alcance	. 3
	4.3 Gestión del Cronograma	. 3
	4.4 Gestión de Costos	. 4
	4.5 Gestión de la Calidad	. 5
	4.6 Gestión de los Recursos Humanos	. 5
	4.7 Gestión de la Comunicación	. 6
	4.8 Gestión de Adquisición y Cierre	. 7
5.	Descripción Técnica	. 8
	5.1 Prototipo de FrontEnd:	. 8
	5.2 Gestión de la Base de Datos	29
6.	Resultados y Conclusión.	37
7	Anexo y Documento de Anovo	38

3. Descripción del Proyecto

3.1 Objetivo General

Desarrollar un sistema integral de venta y administración de boletos aéreos que permita gestionar de manera eficiente el registro de pagos, la programación y control de vuelos, las reservaciones, el procesamiento de reclamos y la generación de estadísticas básicas. El sistema deberá presentar esta información de forma sencilla y dinámica, brindando una experiencia de usuario óptima, y garantizando en todo momento la accesibilidad, seguridad y operatividad en la administración de los servicios.

3.2 Objetivo Específico

- Implementar una interfaz fácil e intuitiva para el usuario.
- Implementar un módulo para el registro y gestión de aerolíneas, aviones y tripulación.
- Diseñar un sistema de administración de reservaciones, que incluya pagos, confirmaciones y cancelaciones.

4. Gestión Integral del Proyecto

4.1 Gestión de la Integración

En términos generales, la gestión del proyecto se llevará a cabo utilizando la metodología ágil Scrum, mediante sprints semanales que faciliten un desarrollo eficiente, organizado y escalable. Para el control de versiones se empleará GitHub, mientras que para la planificación y seguimiento de actividades de corto plazo se utilizará Notion, garantizando así una coordinación efectiva entre los miembros del equipo.

4.2 Gestión del Alcance

El proyecto como mínimo, incorporará los siguientes módulos:

Administraciones de vuelos:

- Registro, edición y eliminación de aerolíneas.
- Administración de aviones (modelo, capacidad, mantenimiento).

Reservaciones y pagos:

- Realización y confirmación de reservas.
- Procesamiento de pagos seguros.
- Gestión de cancelaciones y reembolsos.

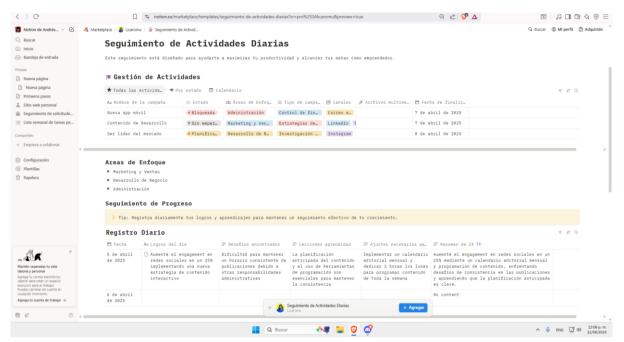
Generación de estadísticas:

- Reportes de número de reservas, cancelaciones y ocupación de vuelos.
- Visualización de métricas clave para la administración.

4.3 Gestión del Cronograma

Consideramos esencial una planificación estructurada que permita una asignación eficiente de recursos y el cumplimiento de los plazos establecidos. Para ello, se emplea **Notion** como herramienta de gestión, organizando el proyecto en fases específicas con fechas de entrega claramente definidas. Esta planificación facilitará el seguimiento del progreso, la identificación temprana de posibles retrasos y el mantenimiento de un flujo de trabajo organizado y

eficiente, asegurando que las metas del proyecto se cumplan dentro del tiempo estimado.



"Plantilla a utilizar en nuestro proyecto"



Para mayor definición de detalles se le deja adjutado este archivo:

https://docs.google.com/document/d/1tLwX7nKr nujmsonxmrLR895reob7ZcB2TYE4sfhpBQ/edit?usp=sharing

4.4 Gestión de Costos

En el análisis de costos, es fundamental considerar los diferentes escenarios que se ajusten a las necesidades y recursos disponibles para el desarrollo del proyecto. A continuación, se presentan las opciones contempladas para las herramientas principales.

Planificación.	
Notion	Costo Estimado (USD
Notion Free	Gratis
Notion Plus	\$8 por Usuario/mes
Notion Business	\$15 por Usuario/mes

Base de Datos.				
Plan	Costo Estimado (USD			
MySQL	Gratis			
MySQL Standard Edition	\$2,000 USD/año			
MySQL Cloud (AWS RDS)	Desde \$15 USD/mes			

4.5 Gestión de la Calidad

Consideramos que, para garantizar la calidad del sistema de gestión de boletos aéreos, se implementarán metodologías y estándares que aseguren la consistencia, mantenibilidad y confiabilidad del código tanto en el backend (Java/Spring) como en el frontend.

Revisión de Código en Git/GitHub

Para realizar y mantener la calidad en nuestro sistema de boletos aéreos, seguimos un proceso claro y colaborativo:

- Creamos ramas para cada tareas o nuevas funcionalidades
- Hacemos commits descriptivos
- Se hará el uso del Pull Request

También se hará la implementación del Checklist para hacer la revisión del código. Cuando revisamos código de otros, verificamos:

- El código es claro y fácil de entender
- Sigue nuestros estándares de programación
- Incluye pruebas para la nueva funcionalidad
- No afecta funcionalidades existentes
- La documentación está actualizada.

Ejemplo del proceso a la hora de la implementación:

Diego realiza la función del check-in online en su rama Cuando termina el crea un pull request en github Jonathan y Andrés revisan el código para la recomendación de mejoras Armando implementa las sugerencias.

4.6 Gestión de los Recursos Humanos

Asignación de Roles y Responsabilidades

Para el desarrollo del sistema de gestión de boletos aéreos, hemos definido los siguientes roles con sus respectivas responsabilidades

Líder & Arquitecto Backend

Responsabilidades Técnicas:

- Diseñar la estructura de la base de datos (vuelos, usuarios, reservas, asientos).
- Crear los endpoints principales con Spring Boot.
- Configurar la seguridad

Tareas concretas en la implementación del proyecto:

- Crear entidades: Vuelo, Usuario, Reserva, Asiento.
- Endpoints: GET /vuelos, POST /reservar, GET /mis-reservas.
- Lógica de asignación de asientos y precios.

Desarrollador Backend & Documentador

Responsabilidades Técnicas:

- Implementar endpoints adicionales y más complejos.
- Conectar frontend con backend.
- Documentar toda la API con Swagger o Postman

Tareas concretas en la implementación del proyecto:

- . Endpoints: POST /pagar, GET /asientos-disponibles, DELETE /cancelar-reserva.
 - Subir documentación a GitHub Wiki o Readme.
 - Ayudar con queries complejas de SQL.

Desarrollador Frontend

Responsabilidades Técnicas:

- Maquetar todas las vistas de la aplicación.
- Consumir los endpoints del backend.
- Hacer la app responsive y usable.

<u>Tareas concretas en la implementación del proyecto:</u>

- Pantallas: Login, Buscador de vuelos, Selección de asientos, Pasarela de pago, Historial.
- Usar Angular/React/Vue con Bootstrap o Tailwind.
- Manejar estados globales (ej: la compra de vuelo ,etc.).

Tester & DevOps

Responsabilidades Técnicas:

- Probar todos los flujos de la app.
- Gestionar GitHub (issues, pull requests).
- Hacer deploy en un entorno de prueba (Heroku, Netlify, Railway).

Tareas concretas en la implementación del proyecto:

- Probar: búsqueda, reserva, pago, cancelación, login.
- Reportar bugs en GitHub Issues.
- Configurar CI/CD con GitHub Actions.

4.7 Gestión de la Comunicación

Informes semanales de progreso

Para garantizar que la información se transmita de manera constante y resolver las dudas a tiempo, se determina que el coordinador enviará semanalmente(viernes) un informe con los logros y dificultades, así como con las tareas por realizar. Cada integrante comunicará sus avances a través de cualquier canal o lo actualizará en Trello, facilitando así el seguimiento visual del proyecto. Para que el grupo se mantenga en sintonía, estas actualizaciones serán revisadas en reuniones cortas.

Se implementará lo siguiente para aclarar las dudas del equipo:

WhatsApp: comunicación rápida y notificaciones urgentes.

Discord: encuentros sincrónicos, discusiones técnicas y colaboración en grupo.

Correo electrónico: comunicación formal con el profesor.

4.8 Gestión de Adquisición y Cierre

Para este punto hemos considerado varios puntos en cuenta, pero lo que más priorizaremos lógicamente será el cumplimiento del sistema a su 100%:

- Codigo Fuente: Todo el código backend (Java/Spring Boot) y frontend de la aplicación.
- **Documentación Técnica (README.md**): El archivo principal que sirve como punto de entrada para entender, configurar y ejecutar el proyecto.
- **Documentación en PDF:** Versión en formato PDF de toda la documentación del proyecto (este mismo documento).
- Base de Datos Scripts SQL (Schema y datos semilla) para recrear la base de datos del sistema.
- Colección de Postman Archivo JSON con la colección de endpoints de la API para pruebas.
 Enlance al repositorio:

https://github.com/JonathanSeguraXP/DWF404 PROYECTO CATEDRA.git

Enlance a Trello:

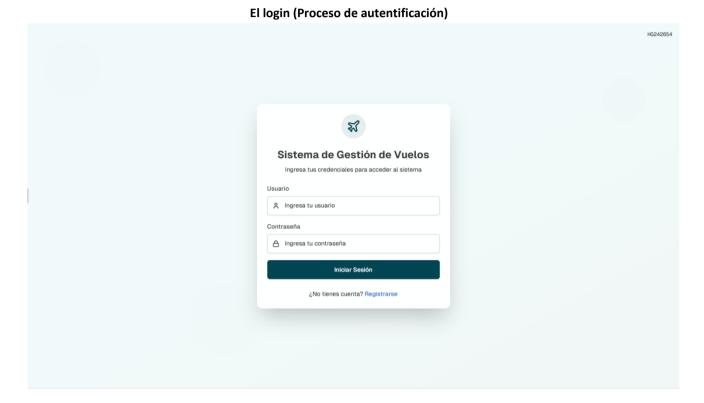
https://trello.com/invite/b/68b3c86bdf74b75a7a507a6e/ATTI073bd8629e1a66bfa672e48fb125ead96062A C3B/plantrabajoproyectocatedradwf

5. Descripción Técnica

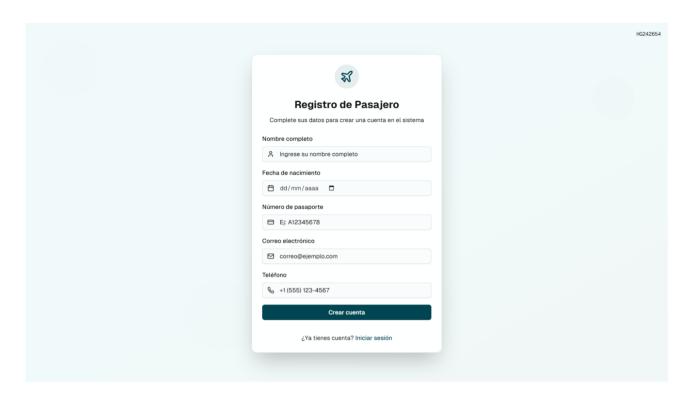
5.1 Prototipo de FrontEnd:

A continuación, se presenta la idea central de los prototipos de interfaz que dejaran mas en claro como funcionara en un futuro con las implementaciones planteadas y con su buen uso de la API REST para la parte del usuario final como lo observara:

Modelos hechos desde la perspectiva de la computadora (Desktop):



En dado caso el usuario no tenga una cuenta con nosotros pues se le indicara a que se registre para que luego pueda logearse como un usuario en el programa.



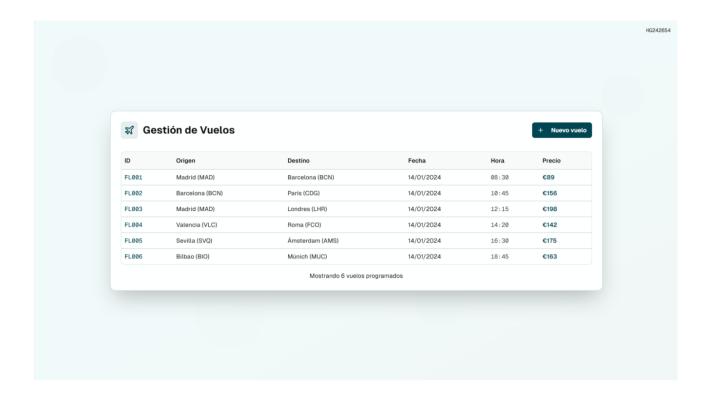
De momento en el lado desde el usuario solo estas 2 imágenes hemos visualizado la idea de el usuario.

Ahora en adelante viene desde la perspectiva de los empleados y sus roles que tienen permisos de visualizar lo que no pueden observar los pasajeros que en este caso seria el cliente por ciertos privilegios:

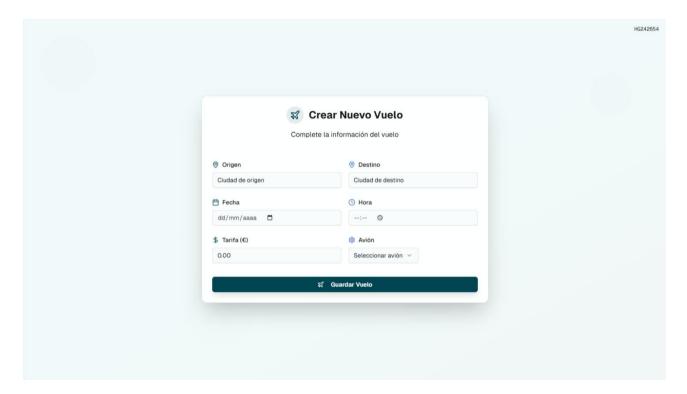
Como se puede observar con lo anterior si el administrador se logea le aparecerá la siguiente interfaz que le ayudará a tener un mejor control de flujo con respecto a su servicio y como se está desarrollando el día a día con la aerolínea.



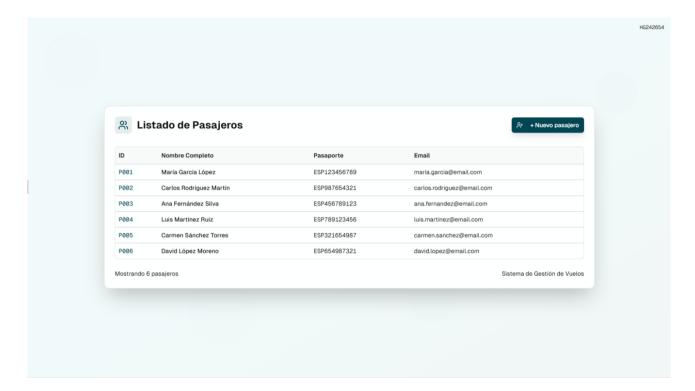
Aquí se muestra el flujo de cómo está la información de los vuelos de los pasajeros que han hecho y sus respectivos datos



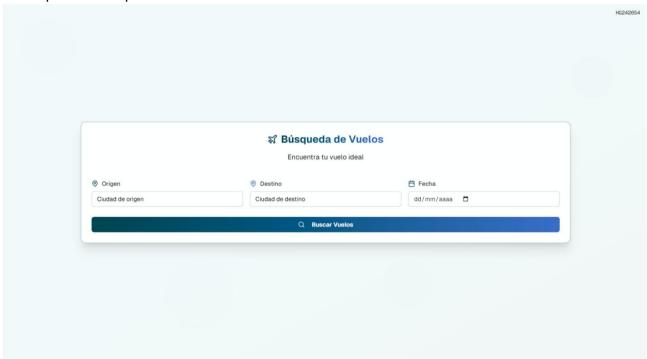
Desde aquí se hace la creación de los datos del vuelo para el usuario.

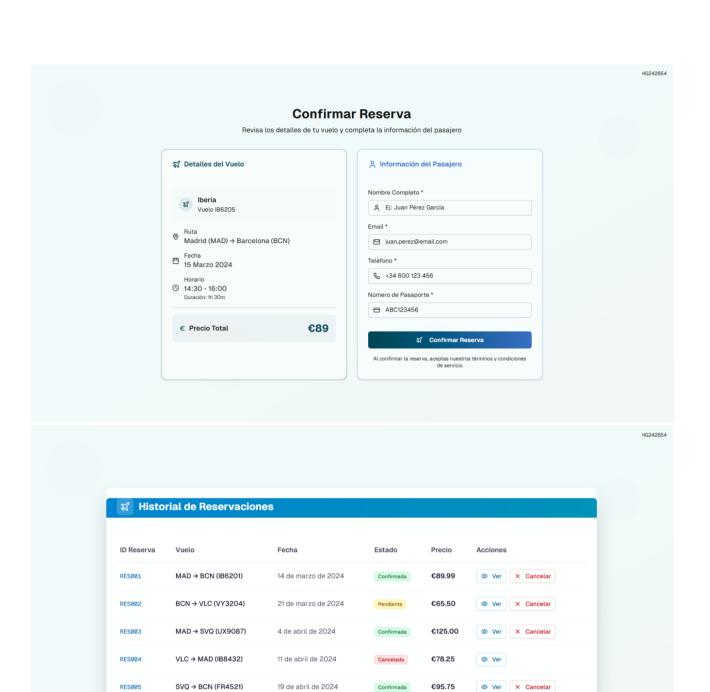


Verificación de los datos personales de los pasajeros.



Su respectiva búsqueda de vuelos









Registro de Pasajero

Complete sus datos para crear una cuenta en el sistema

Nombre completo



Fecha de nacimiento



Número de pasaporte

☐ Ej: A12345678

Correo electrónico

□ correo@ejemplo.com

Teléfono

♦ +1 (555) 123-4567

Crear cuenta

¿Ya tienes cuenta? Iniciar sesión



Dashboard Administrador

Sistema de Gestión de Vuelos



Gestión de Vuelos

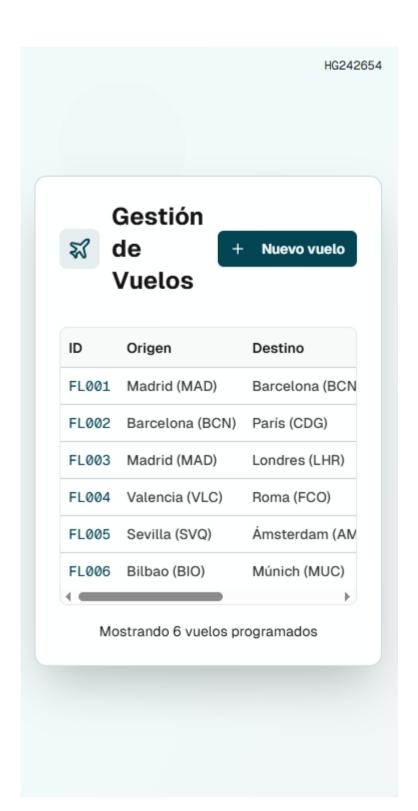


Gestión de Pasajeros



Reservaciones

Selecciona una opción para continuar





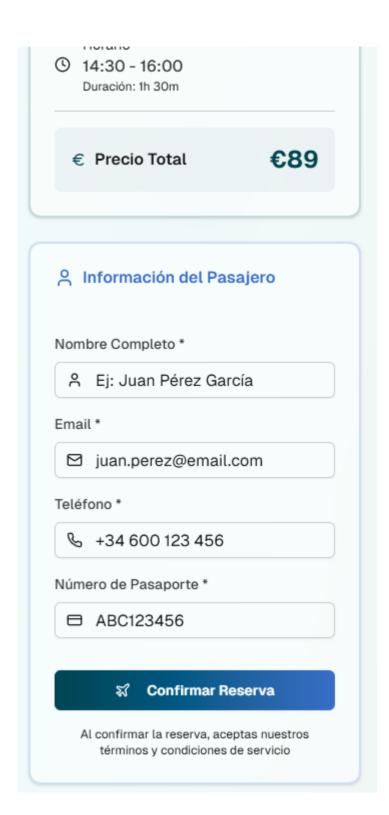


ID	Nombre Completo	Pasaporte
P001	María García López	ESP123456789
P002	Carlos Rodríguez Martín	ESP98765432:
P003	Ana Fernández Silva	ESP456789123
P004	Luis Martínez Ruiz	ESP789123456
P005	Carmen Sánchez Torres	ESP321654987
P006	David López Moreno	ESP654987321
4 4)

pasajeros Vuelos







₩ Historial de **Reservaciones** Vuelo Fecha Estado serva 14 de $MAD \rightarrow$ marzo BCN S001 Confirmada de (IB6201) 2024 21 de BCN → marzo VLC S002 Pendiente de (VY3204) 2024 4 de $MAD \rightarrow$ abril SVQ S003 Confirmad: de (UX9087) 2024 11 de VLC → abril S004 MAD Cancelada de (IB8432) 2024



1. Autenticación y Registro

Método	Endpoint	Descripción
POST	/auth/login	Login de usuario (admin o pasajero)
POST	/auth/registro	Registro de nuevo pasajero

2. Gestión de Vuelos (Admin)

Método	Endpoint	Descripción
GET	/admin/vuelos	Listar todos los vuelos
POST	/admin/vuelos	Crear un nuevo vuelo
PUT	/admin/vuelos/{id}	Actualizar un vuelo existente
DELETE	/admin/vuelos/{id}	Eliminar un vuelo
GET	/admin/vuelos/{id}	Obtener detalles de un vuelo

3. Búsqueda de Vuelos (Pasajero)

Método	Endpoint	Descripción
GET	/vuelos/buscar	Búsqueda de vuelos por origen, destino y fecha
GET	/vuelos/{id}	Detalles de un vuelo específico

4. Reserv	vaciones (Pasajero)			
Método	Endpoint		Descripción	
POST	/reservas		Crear una nueva re	serva
BET	/reservas/usuario/{id	Pasajero}	Listar reservas de u	un pasajero
UT	/reservas/{id}/cancel	ar	Cancelar una reser	va
ET	/reservas/{id}		Detalles de una res	serva
. Gestió	n de Pasajeros (Admin)			
⁄létodo	Endpoint	Descripc	ión	
ET	/admin/pasajeros	Listar too	dos los pasajeros	
ET	/admin/pasajeros/{id}	Ver detal	les de un pasajero	
ELETE	/admin/pasajeros/{id}	Eliminar	un pasajero	
3. Dashb	oard (Admin)			
Método	Endpoint	Descripción	n	
vietodo				
GET	/admin/estadisticas	Obtener est	tadísticas (reservas,	ingresos, etc.

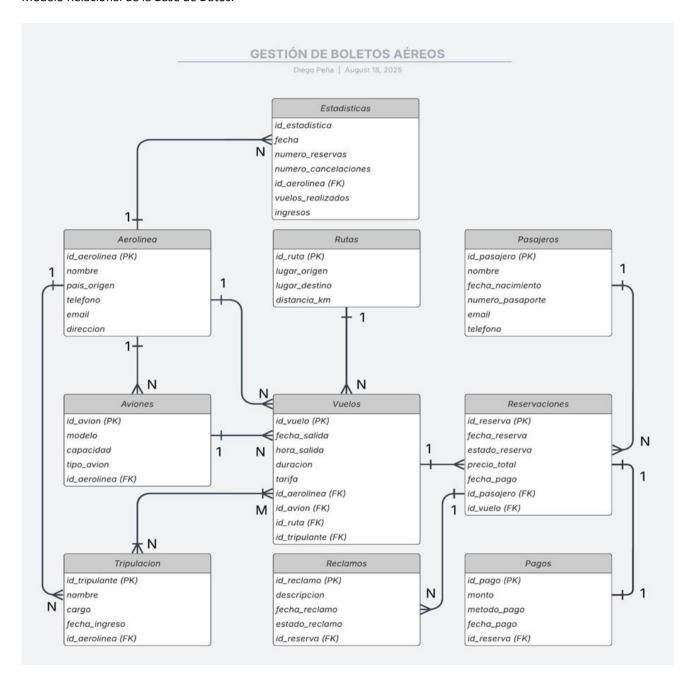
7. Gestión de Reclamos

Método	Endpoint	Descripción
POST	/reclamos	Crear un reclamo asociado a una reserva
GET	/reclamos/usuario/{idPasajero}	Listar reclamos de un pasajero
PUT	/reclamos/{id}/estado	Actualizar estado de un reclamo (admin)

5.2 Gestión de la Base de Datos

Estructura de la Base de Datos:

Modelo Relacional de la Base de Datos.



- Relaciones entre tablas.

[°] Aerolínea – Aviones: Una aerolínea tiene varios aviones

[°] Rutas – Vuelos: Una ruta puede tener muchos vuelos y cada vuelo tiene una ruta asociada.

[°] Aviones – Vuelos: Un avión puede realizar muchos vuelos.

Diccionario de Datos:

Tabla: Aerolínea

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_aerolinea	INT	Identificador único de la aerolínea		Enteros >0	PRIMARY KEY, UNIQUE, NOT NULL	Aviones, Vuelos, Tripulación, Estadísticas
Nombre	VARCHAR	Nombre de la aerolínea	100		NOT NULL, UNIQUE	
pais_origen	VARCHAR	País de origen de la aerolínea	50		NOT NULL	
Teléfono	VARCHAR	Teléfono de contacto	20		NULL	
Email	VARCHAR	Correo Electrónico de contacto	100	Email válido	UNIQUE, NOT NULL	
Dirección	VARCHAR	Dirección de la aerolínea	200		NULL	

[°] Aerolínea – Vuelos: Una aerolínea puede tener muchos vuelos.

[°] Pasajero – Reservaciones: Un pasajero puede hacer muchas reservaciones.

[°] Vuelos – Reservaciones: Un vuelo puede tener muchas reservaciones.

[°] Reservaciones – Pagos: Una reservación puede tener un solo pago, pero un pago corresponde a una sola reservación.

^{*} Vuelos – Tripulación: Un vuelo puede tener varios tripulantes y un tripulante puede volar en varios vuelos.

[°] Reservaciones – Reclamos: Una reservación puede tener varios reclamos asociados.

[°] Estadística – Aerolínea: Una aerolínea puede tener muchas estadísticas registradas.

[°] **Tripulación – Aerolínea:** Muchos tripulantes pueden estar asociados a una aerolínea.

Tabla: Aviones

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_avion	INT	Identificador único del avión		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	Vuelos
Modelo	VARCHA R	Modelo del Avión	100		NOT NULL	
capacidad	INT	Capacidad de pasajeros		Entero >0	NOT NULL	
tipo_avion	VARCHA R	Tipo o clase (Boeing, Airbus, etc.)	100		NOT NULL	
id_aerolinea	INT	Aerolínea Propietaria	100	Entero >0	FOREIGN KEY, NOT NULL	Aerolínea

Tabla: Rutas

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_ruta	INT	Identificador único de la ruta		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	Vuelos
lugar_origen	VARCHAR	Lugar de salida del vuelo	100		NOT NULL	

lugar_destino	VARCHAR	Lugar de destino del vuelo	100		NOT NULL	
distancia_km	DECIMAL (6,2)	Distancia entre el lugar de origen a lugar de destino en km		Decimales >0	NOT NULL	

Tabla: Vuelos

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_vuelo	INT	Identificador único del vuelo		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	Reservaciones
fecha_salida	DATE	Fecha de Salida del Vuelo		YYYY-MM- DD	NOT NULL	
hora_salida	TIME	Hora de Salida del Vuelo		00:00:00	NOT NULL	
Duración	TIME	Duración estimada del vuelo		00:00:00	NOT NULL	
Tarifa	DECIMAL (10,2)	Precio Base del Vuelo		Decimales >=0	NOT NULL	
id_aerolinea	INT	Aerolínea que opera el vuelo		Enteros	FOREIGN KEY, NOT NULL	Aerolínea
id_avion	INT	Avión asignado al vuelo		Enteros >0	FOREIGN KEY, NOT NULL	Aviones

id_ruta	INT	Ruta del vuelo	Enteros >0	FOREIGN KEY, NOT NULL	Rutas	
id_tripulante	e INT	Tripulante asignado en el vuelo	Enteros >0	FOREIGN KEY, NOT NULL	Tripulación	

Tabla: Pasajeros

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_pasajero	INT	Identificador único del pasajero		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	Reservaciones
nombre	VARCHAR	Nombre y Apellido completo	100		NOT NULL	
fecha_nacimie nto	DATE	Fecha de Nacimiento del Pasajero		YYYY-MM- DD	NOT NULL	
numero_pasap orte	VARCHAR	Número del Pasaporte	50	Único por persona	UNIQUE, NOT NULL	
email	VARCHAR	Correo Electrónico	100	Email válido	NOT NULL	
telefono	VARCHAR	Teléfono de Contacto	20		NULL	

Tabla: Reservaciones

id_reserva	INT	Identificador único de la reserva	Enteros	PRIMARY KEY, UNIQUE, NOT NULL	Pagos, Reclamos
fecha_reserva	DATE	Fecha en que se realizó la reserva	YYYY-MM- DD	NOT NULL	
estado_reserva	ENUM	Estado actual de la reserva	Confirmada , Pendiente, Cancelada	NOT NULL	
precio_total	DECIMAL (10,2)	Precio final pagada por el pasajero	Decimal >=0	UNIQUE, NOT NULL	
fecha_pago	DATE	Fecha en que se realizó el pago	YYYY-MM- DD	NOT NULL	
id_pasajero	INT	Pasajero que realizó la reserva	Enteros >0	FOREIGN KEY, NOT NULL	Pasajeros
id_vuelo	INT	Vuelo reservado	Enteros	FOREIGN KEY, NOT NULL	Vuelos

Tabla: Pagos

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_pago	INT	Identificador único del pago		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	
Monto	DECIMAL (10,2)	Monto pagado de la reserva		Decimales >=0	NOT NULL	

método_pago	ENUM	Método en que se realizara el pago	Tarjeta, Transferen cia	NOT NULL	
fecha_pago	DATE	Fecha del Pago	YYYY-MM- DD	NOT NULL	
id_reserva	INT	Correo Electrónico	Enteros >0	FOREIGN KEY, NOT NULL	Reservaciones

Tabla: Tripulación

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_tripulante	INT	Identificador único del tripulante		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	Vuelos
nombre	VARCHAR	Nombre y Apellido completo	100		NOT NULL	
Cargo	ENUM	Rol que desempeñar a el tripulante		Piloto, Copiloto, Azafata	NOT NULL	
fecha_ingreso	DATE	Fecha de Incorporació n		YYYY-MM- DD	NOT NULL	
id_aerolinea	INT	Aerolínea a la que pertenece		Enteros >0	FOREIGN KEY, NOT NULL	Aerolínea

Table: Reclamos

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_reclamo	INT	Identificador único del reclamo		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	Vuelos
descripción	VARCHAR	Descripción detallada del reclamo	255		NOT NULL	
fecha_reclamo	DATE	Fecha en que se presentó el reclamo		YYYY-MM- DD	NOT NULL	
estado_reclam o	ENUM	Estado del reclamo		Pendiente, Resuelto, Cancelado	NOT NULL	
id_reserva	INT	Reserva asociada		Enteros	FOREIGN KEY, NOT NULL	Reservas

Tabla: Estadísticas

Nombre del Campo	Tipo de Dato	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id_estadistica	INT	Identificador único de la estadística		Enteros	PRIMARY KEY, UNIQUE, NOT NULL	
Fecha	DATE	Fecha de la estadística		YYYY-MM- DD	NOT NULL	
numero_reserv as	INT	Total de reservas registradas		Enteros >=0	NOT NULL	
numero_cance laciones	INT	Cantidad de cancelacione s registradas		Enteros >=0	NOT NULL	

id_aerolinea	INT	Aerolínea a la que pertenece la estadística	Enteros >0	FOREIGN KEY, NOT NULL	Aerolínea
vuelos_realiza dos	INT	Cantidad de vuelos realizados	Enteros	NOT NULL	
Ingresos	Decimal (10,2)	Ingreso totales que obtuvo la aerolínea	Decimales >=0	NOT NULL	

6. Resultados y Conclusión.

a) Resultados Obtenidos

Resultados obtenidos: La API REST facilitó la integración del frontend con el backend en el proyecto. Se implementaron las funciones esenciales para gestionar vuelos, aerolíneas, pasajeros y reservas, de acuerdo con los objetivos fijados para la fase académica. El sistema funcionó de manera estable dentro del alcance predefinido, lo que posibilitó la demostración de la viabilidad de la propuesta.

Conclusiones

Durante el proceso de desarrollo se obtuvieron aprendizajes significativos tanto en el ámbito técnico como en el organizativo:

- Aplicación práctica de conocimientos en diseño de bases de datos, programación con Java y Spring Boot, integración de MySQL y desarrollo frontend.
- Importancia de la gestión del proyecto, utilizando Trello/Notion y GitHub para organizar tareas, controlar el avance y centralizar el código.

Principales desafíos:

- O Definir qué módulos incluir en la primera fase para no exceder el alcance.
- Conectar las entidades de la base de datos con la lógica de negocio.
- o Diseñar una interfaz clara que muestre las funciones principales sin sobrecargar al usuario.
- **Trabajo en equipo**: se logró distribuir tareas entre los integrantes y coordinar entregas parciales de manera efectiva.

Mejoras Futuras

El sistema puede evolucionar incorporando:

- Módulo de pagos en línea, para completar el flujo de compra de boletos.
- Gestión de reclamos para dar soporte al cliente.
- Panel de estadísticas con métricas de vuelos, ingresos y cancelaciones.
- Mejoras en la experiencia de usuario (UX/UI) con interfaces más modernas e intuitivas.

7. Anexo y Documento de Apoyo.

Documentacion de Spring Boot

https://spring.io/projects/spring-boot

Documentacion de java

https://www.java.com/es/

Procedimientos recomendados para el diseño de LA API web RESTful

https://learn.microsoft.com/es-es/azure/architecture/best-practices/api-design