

Universidad Don Bosco



Proyecto de Desarrollo de Aplicaciones Web – Fase 1

Integrantes:

Fernando Josué Anzora Aquino AA222744 (BACK-END)

Luis Enrique Cartagena Arteaga CA241136(FRONT-END)

Marlon Omar Guzmán Mejía GM190258 (TESTER)

Daniel Enrique Flores Lino FL220294(BACK-END)

Docente: Mario Alvarado

Asignatura: Desarrollo de Aplicaciones con Web Frameworks

San Salvador, 2025

Índice

1. Descripción del Proyecto

Objetivo General

El presente proyecto tiene como propósito diseñar y documentar un sistema digital orientado a la gestión de vuelos y reservas. La iniciativa responde a la necesidad de contar con un entorno web organizado que permita a los usuarios registrar, consultar y administrar información clave de manera eficiente. El trabajo en esta primera fase busca establecer las bases conceptuales, metodológicas y de coordinación necesarias para avanzar en el desarrollo posterior del sistema, y desarrollar la planificación inicial de un sistema web que facilite la administración de vuelos, reservas y usuarios.

También representa un primer acercamiento a cómo organizar y planificar un trabajo de desarrollo en equipo para aprender a trabajar de forma estructurada a comunicarnos con claridad y a acostumbrarnos a uso de herramientas que en el futuro serán parte del trabajo profesional.

Objetivos Específicos

- Comprender la problemática planteada y traducirla en un plan de trabajo claro.
- Definir el alcance y limitaciones del proyecto.
- Organizar la metodología de trabajo en equipo.
- Establecer lineamientos de comunicación y control de avances.
- Preparar la estructura inicial de la base de datos.

2. Gestión Integral del Proyecto

Gestión de la Integración

Para integrar el trabajo de todos los miembros de nuestro equipo se decidió utilizar GitHub como repositorio común. La idea no es solo almacenar archivos, sino practicar con control de versiones. El trabajo será guiado bajo la metodología ágil Scrum, dividiendo las actividades en sprints semanales, con metas de corto plazo y revisiones periódicas. El uso de plataformas como Google Calendar compartido y WhatsApp servirá para la planificación de tareas y el seguimiento de actividades.

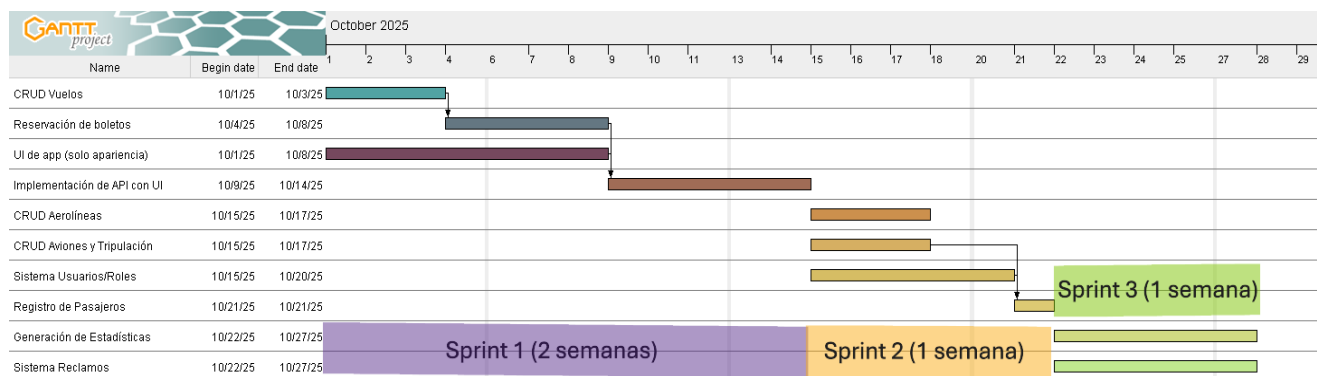
Gestión del Alcance

He aquí las funciones y características que se tienen planeadas para la aplicación final:

- **Gestionar aerolíneas.** La aplicación debe permitir añadir aerolíneas, verlas, editarlas y hasta eliminarlas de ser necesario (CRUD completo).
- **Gestionar vuelos.** Permite añadir vuelos con la información pertinente (aerolínea, ruta, duración del vuelo, tipo de avión y tarifa) y el resto del CRUD.
- **Gestión de aviones.** Permite realizar CRUD completo sobre los aviones existentes, incluyendo la cantidad de asientos que tiene.
- **Gestión de tripulación.** CRUD para los empleados que abordan los aviones, y permite asignarlos a vuelos específicos.
- **Gestión de reservaciones.** Los usuarios deben ser capaces de reservar y pagar por boletos para vuelos que se ajusten a sus preferencias. En el caso que lo deseen, se les debe permitir modificar o cancelar su reservación (en base a las políticas de la aerolínea).
- **Registro de pasajeros.** Permite ver a todos los pasajeros registrados en un vuelo, y sus asientos en el avión.
- **Procesamiento de reclamos.** Los usuarios tienen la opción de escribir y subir cualquier reclamo que tengan sobre su vuelo a través de un formulario, donde puede ser leído por empleados para solucionar los problemas del cliente y mejorar su experiencia. Si la inquietud del usuario fue resuelta, los empleados pueden marcar el reclamo como solucionado.
- **Generación de estadísticas.** Muestra información útil diversa en base al número de reservaciones, cancelaciones, etc.
- **Sistema de usuarios.** Un login que permite a usuarios crear cuentas e iniciar sesión para utilizar el servicio, y permite a empleados iniciar sesión con roles específicos para acceder y modificar a la información guardada.

Gestión del Cronograma

Calendario del proyecto



Hitos y plazos clave

- 3 de octubre: CRUD de la tabla de vuelos escrito y testeado.
- 8 de octubre: sistema de reservaciones y UI finalizadas.
- 14 de octubre: la aplicación es implementada exitosamente a la interfaz de usuario.
- 20 de octubre: sistema de usuarios y roles terminado.

Gestión de Costos

Costos iniciales	Monto
Computadoras (4)	\$ 3,000.00
Costos anuales	Monto
IntelliJ IDEA Ultimate (4)	\$ 676.00
Servidores e instancia SQL en Google Cloud	\$ 2,715.00
Total en 1 año	\$ 6,391.00

Gestión de la Calidad

En cuanto a la calidad del código del proyecto de aerolíneas, será hecho con commits en GitHub y será revisado al menos por un compañero cada fin de semana con los avances que hicimos en el proyecto. Exportaremos las dependencias necesarias para el proyecto, para no tener dependencias que no ocupemos y tener redundancia.

Haremos pruebas unitarias que prueben la lógica de negocio y utilidades del proyecto.

Validar las API y su funcionamiento bien planteado con sus respectivas pruebas con Postman u otras aplicaciones de peticiones como cURL Bruno o RestAssured.

Simularemos entornos reales donde los usuarios ocuparán la aplicación.

Se harán pruebas de dependencias Maven y plugins para no tener problemas al inicializar la aplicación.

Conclusión: Haremos pruebas del código cada fin de semana junto con lo que hemos avanzado en la aplicación con commits en la plataforma GitHub; haremos pruebas unitarias en el entorno SpringBoot; abarcaremos muchos temas del código como las dependencias, las reglas de negocio, problemas de lógica, y vulnerabilidades que pueda tener la aplicación; usaremos la documentación de SpringBoot y mantendremos el código limpio y comprensible para de esa forma tener un mejor desarrollo de la aplicación.

Gestión de los Recursos Humanos

En este proyecto para agilizar el trabajo nos comunicaremos por la aplicación de mensajería WhatsApp, con la cual se nos facilita ver los avances y dejar cosas en acuerdo. Además, el código lo estaremos subiendo a GitHub con sus commits y sus mensajes de descripción sobre que lo que hicimos. Haremos llamadas por la plataforma de Meet de Google para concordar los siguientes avances que haremos en el proyecto, además de que 2 miembros del equipo podrán tener llamadas de Meet cada fin de semana para poder hacer los tests del código. Al tener dos personas haciendo las pruebas, se acelera el tiempo de testeo, y se permite tener una mayor visión de qué mejorar, o identificar los errores en los tests. Se tendrá un reporte semanal sobre los avances del proyecto, en lo cual nos organizaremos y veremos el desempeño de todos. También, nos daremos todos retroalimentación entre los del equipo sobre mejoras del código, su funcionalidad y la lógica de este.

Asignación de roles:

El scrum master y encargado del frontend es Luis Enrique Cartagena Arteaga, el cual hará la planificación de reuniones técnicas y facilitará una mejor comunicación entre todos los miembros del equipo, además de que él se encargó de la creación de las maquetas y sus estaciones.

Backend: Fernando Josué Anzora Aquino y Daniel Enrique Flores Lino se encargarían de la lógica del proyecto y su desarrollo.

Tester: Marlon Omar Guzmán Mejía se encargaría de las pruebas que haremos en la aplicación. Estas serán necesarias cada fin de semana.

Gestión de la Comunicación

La comunicación es un aspecto fundamental para el éxito del proyecto. El equipo establecerá reuniones semanales virtuales, por medio de Google Meet o Zoom, destinadas a la presentación de avances y resolución de dudas. De manera complementaria, se utilizarán canales de mensajería instantánea como WhatsApp o Discord para la coordinación diaria. Asimismo, se mantendrá un registro de acuerdos y tareas pendientes en documentos compartidos a través de Google Drive. Esta estrategia busca garantizar la transparencia, el compromiso y la eficiencia en la ejecución de las actividades programadas.

Gestión de Adquisición y Cierre

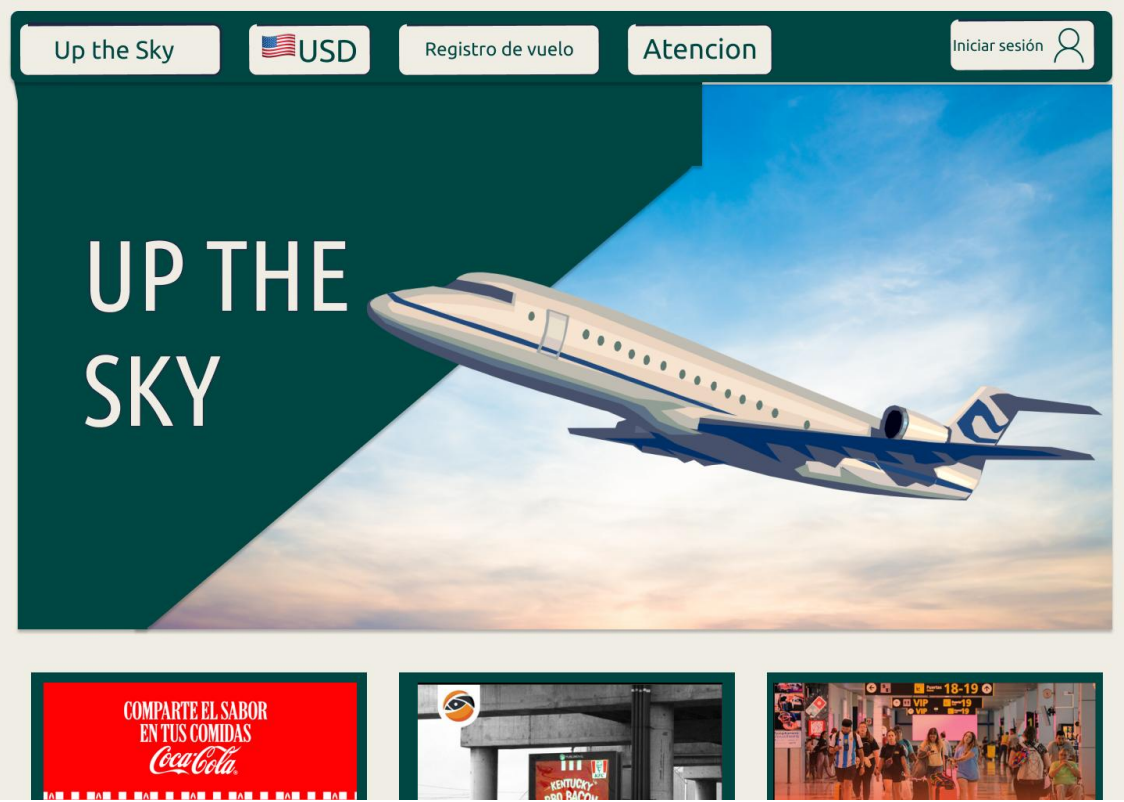
Para poder finalizar el proyecto, el programa debe estar subido a GitHub, con la documentación explicando su funcionamiento y los pasos para utilizarlo, explicados en el README y en un PDF enlazado en el repositorio.

<https://github.com/DFQuack/AppVuelos-Frameworks/>

3. Descripción de técnica

Prototipo de Frontend

Página de inicio de la plataforma de vuelos



Parte de registro de vuelo

Up the Sky

 USD

Registro de vuelo

Atencion

Iniciar sesión 

Registro de Vuelo

Nombre y apellido

Ciudad de origen

Destino

Fecha de hoy

Hora de salida

Registrar Vuelo

Cancelar

Regresar

Como se vería con datos el registro de vuelo tomaremos estos datos como ejemplo para los siguientes mockups

Up the Sky

 USD

Registro de vuelo

Atencion

Iniciar sesión 

Registro de Vuelo

Nombre y apellido

Ciudad de origen

Destino

Fecha de hoy

Hora de salida

Registrar Vuelo

Cancelar

Regresar

Como se vería la pestaña de registro de usuario para que hipotéticamente se pudiera hacer un asociado de la plataforma

The mockup shows a web page with a dark green header and a dark teal main content area. The header contains the 'Up the Sky' logo, a 'USD' currency selector, and navigation links for 'Registro de vuelo' and 'Atencion'. A user login link 'Iniciar sesión' with a person icon is in the top right. The main content area is titled 'Registro' and contains four input fields for 'Usuario', 'Correo electronico', 'Contraseña', and 'Telefono'. Below these fields is a descriptive sentence: 'Crea una cuenta para tener ofertas, y descuentos de tus vuelos y poder coleccionar puntos, para descuentos en el futuro.' At the bottom are two buttons: 'Registrarse' and 'Regresar'.

Un inicio de sesion con ya las credenciales dadas en el registro

The mockup shows a web page with a dark green header and a dark teal main content area. The header is identical to the registration page, with the 'Up the Sky' logo, 'USD' selector, 'Registro de vuelo' and 'Atencion' links, and the 'Iniciar sesión' link. The main content area is titled 'Inicio sesion' and contains two input fields for 'Usuario' and 'Contraseña'. Below these fields are two buttons: 'Iniciar sesion' and 'Regresar'. At the bottom is a descriptive sentence: 'Inicia sesion en tu cuenta para tener ofertas, y descuentos de tus vuelos y poder coleccionar puntos, para descuentos en el futuro.'

Lista de vuelos de vuelos disponibles los cuales ya están programados y con diferentes precios dependiendo el avión y el horario de salida

Up the Sky	USD	Registro de vuelo	Atencion	Iniciar sesión
Vuelos disponibles				
Aerolineas Multiples AVIANCA	Soyapango El Salvador - Nueva York El Salvador	12:00AM - 04:00PM 4 horas	Airbus a320	\$360
DELTA	Soyapango El Salvador - Nueva York El Salvador	2:00PM - 06:00PM 4 horas	Airbus a321	\$450
United	Soyapango El Salvador - Nueva York El Salvador	7:00PM - 11:00PM 4 horas	boeing 747	\$270
DELTA	Soyapango El Salvador - Nueva York El Salvador	8:00AM - 12:00AM 4 horas	Boeing 777	\$450
United	Soyapango El Salvador - Nueva York El Salvador	12:00AM - 04:00PM 4 horas	boeing 747	\$250
Aerolineas Multiples AVIANCA	Soyapango El Salvador - Nueva York El Salvador	6:00AM - 10:00AM 4 horas	Airbus a319	\$360
United	Soyapango El Salvador - Nueva York El Salvador	1:00PM - 5:00PM 4 horas	Airbus a320	\$350
Aerolineas Multiples AVIANCA	Soyapango El Salvador - Nueva York El Salvador	9:00PM - 01:00AM 4 horas	boeing 737	\$200
DELTA	Soyapango El Salvador - Nueva York El Salvador	5:00AM - 09:00AM 4 horas	Embraer 175	\$450

Ingreso de los datos del usuario junto con el vuelo ya elegido de preferencia

Up the Sky	USD	Registro de vuelo	Atencion	Iniciar sesión
Aerolineas Multiples AVIANCA	Soyapango El Salvador - Nueva York El Salvador	9:00PM - 01:00AM 4 horas	boeing 737	\$200
Ingreso de datos de usuario				
Nombre Completo				
<input type="text"/>				
Fecha Nacimiento				
<input type="text"/>				
Numero de pasaporte				
<input type="text"/>				
Confirmar		Regresar		

Verificación de los datos junto con el vuelo y el anuncio sobre que la reserva ya fue hecha junto con los asientos disponibles también que se puede modificar la reserva o cancelarla

Up the Sky

USD

Registro de vuelo

Atencion

Iniciar sesión

Verificacion de datos

Vuelo seleccionado

Aerolines Multiples
AVIANCA

Soyapango El Salvador - Nueva York El Salvador

9:00PM - 01:00AM
4 horas

boeing 737

Asientos disponibles

73B

52A

74A

23B

Datos Personales

Luis Enrique Cartagena Arteaga

04/12/04

ACA34561

Cuenta total:

\$200 Vuelo

\$30 Impuestos y tasas aeroportuarias

\$10 TUA

\$25 Cargos de la aerolínea

TOTAL

\$265 Total

Reserva generada !

Continuar con el pago

Modificar reserva

Cancelar reserva

Modificación de reserva donde se puede reelegir el vuelo con el presupuesto que se ajuste más al usuario donde ya se continua con la nueva reserva y poder cancelarla

Up the Sky

USD

Registro de vuelo

Atencion

Iniciar sesión

Modificacion de reserva

Aerolines Multiples
AVIANCA

Soyapango El Salvador - Nueva York El Salvador

9:00PM - 01:00AM
4 horas

boeing 737

Datos Personales

Luis Enrique Cartagena Arteaga

04/12/04

ACA34561

Eleccion de otro vuelo

continuar con la reserva actual

Cancelar reserva

\$200 Vuelo

\$30 Impuestos y tasas aeroportuarias

\$10 TUA

\$25 Cargos de la aerolínea


TOTAL

\$265 Total

pág. 12


Pago de reserva con tarjeta de crédito y el monto total a pagar

Up the Sky

 USD

Registro de vuelo

Atencion

Iniciar sesión 

Confirmacion de pago

Tarjeta de credito

Nombre del titular

Fecha de nacimiento MM/AA

codigo de seguridad

Confirmar pago

Cancelar

Cuenta total:

\$200 Vuelo

\$30 Impuestos y tasas aeroportuarias

\$10 TUA

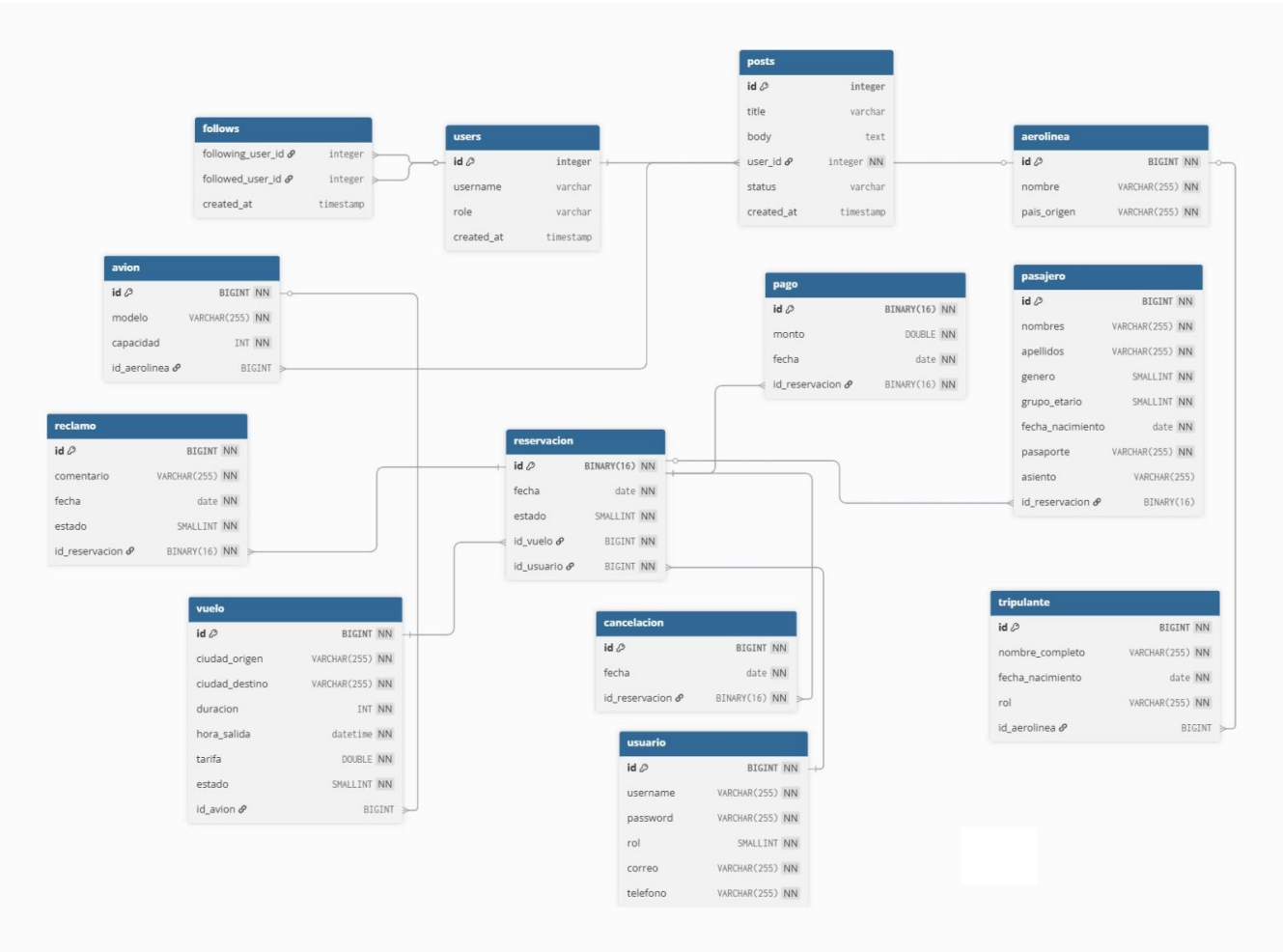
\$25 Cargos de la aerolínea

TOTAL

\$265 Total

Gestión de la Base de Datos

Estructura de la base de datos



Diccionario de datos: cubre campos clave de las principales tablas, y está estructurado para servir como guía técnica:

Usuarios						
Campo	Tipo de Datos	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id	SERIAL	Identificador único del usuario	—	Numérico incremental	Clave primaria, no nulo	—
nombre_completo	VARCHAR	Nombre completo del usuario	100	Texto libre	No nulo	—
correo	VARCHAR	Correo electrónico del usuario	100	Formato email válido	Único, no nulo	—
contraseña	VARCHAR	Contraseña cifrada	255	Texto cifrado	No nulo	—
rol_id	INT	Rol asignado al usuario	—	Referencia a roles.id	Clave foránea, no nulo	roles

Vuelos						
Campo	Tipo de Datos	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id	SERIAL	Identificador único del vuelo	—	Numérico incremental	Clave primaria, no nulo	—
aerolinea_id	INT	Aerolínea que opera el vuelo	—	Referencia a aerolineas.id	Clave foránea, no nulo	aerolineas
avion_id	INT	Avión asignado al vuelo	—	Referencia a aviones.id	Clave foránea, no nulo	aviones
ruta_id	INT	Ruta del vuelo	—	Referencia a rutas.id	Clave foránea, no nulo	rutas
fecha_salida	TIMESTAMP	Fecha y hora de salida	—	Fecha válida	No nulo	—
tarifa	DECIMAL(10,2)	Precio base del boleto	—	≥ 0	No nulo	—
estado	VARCHAR	Estado del vuelo	50	Programado, Cancelado	Valor por defecto: Programado	—

Pasajeros						
Campo	Tipo de Datos	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id	SERIAL	Identificador único del pasajero	—	Numérico incremental	Clave primaria, no nulo	—
Usuario	INT	Usuario reservado	—	Referencia a Usuarios.id	Clave foránea	usuarios
fecha_nacimiento	DATE	Fecha de nacimiento	—	Fecha válida	No nulo	—
numero_pasaporte	VARCHAR	Número de pasaporte	50	Texto alfanumérico	Único, no nulo	—

Reservaciones						
Campo	Tipo de Datos	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id	SERIAL	Identificador único de la reservación	—	Numérico incremental	Clave primaria, no nulo	—
vuelo_id	INT	Vuelo reservado	—	Referencia a vuelos.id	Clave foránea, no nulo	vuelos
pasajero_id	INT	Pasajero que reserva	—	Referencia a pasajeros.id	Clave foránea, no nulo	pasajeros
preferencia_asiento	VARCHAR	Preferencia de asiento	10	Ventana, Pasillo, NULL	Opcional	—
fecha_reserva	TIMESTAMP	Fecha de la reserva	—	Fecha válida	Valor por defecto: CURRENT_TIMESTAMP	—
estado	VARCHAR	Estado de la reserva	50	Pendiente, Confirmada, Cancelada	Valor por defecto: Pendiente	—

Pagos						
Campo	Tipo de Datos	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
id	SERIAL	Identificador único del pago	—	Numérico incremental	Clave primaria, no nulo	—
reservacion_id	INT	Reservación asociada	—	Referencia a reservaciones.id	Clave foránea, no nulo	reservaciones

tipo_pago	VARCHAR	Método de pago	50	Tarjeta, Transferencia, Efectivo	No nulo	—
monto	DECIMAL(10, 2)	Monto pagado	—	≥ 0	No nulo	—
fecha_pago	TIMESTAMP	Fecha del pago	—	Fecha válida	Valor por defecto: CURRENT_TIMESTAMP	—

4. Resultados y Conclusiones

Resultados Obtenidos

El sistema ofrece una capacidad total para la administración de billetes de avión. Las acciones de crear, leer, actualizar y borrar (CRUD) marchan sin problemas, con respuestas rápidas gracias a la base de Spring Boot y al manejo efectivo de Spring Data JPA. El sistema aguanta muchas peticiones a la vez, exhibiendo un buen rendimiento con mucha carga. La forma en que está hecho el código ayuda a entenderlo y a meterle cosas nuevas.

Conclusiones

Usar SpringBoot permitió una configuración rápida y eficiente, acortando el tiempo de desarrollo. Lo más difícil fue comprobar que los datos complejos estuvieran bien y solucionar los errores, lo cual se hizo poniendo comprobaciones en el servicio y los controladores, y usando controladores de errores globales. Aprendimos que es clave seguir una forma de hacer las cosas desde el principio para que el código se pueda ampliar y mantener bien.

5. Anexos y Documentación de Apoyo

Ejemplo de Código (Controller)

```
import org.springframework.web.bind.annotation.*;
import org.springframework.http.ResponseEntity;

@RestController
@RequestMapping("/api/v1/boletos")
public class BoletoController {

    private final BoletoService boletoService;

    public BoletoController(BoletoService boletoService) {
        this.boletoService = boletoService;
    }

    @PostMapping
    public ResponseEntity<Boleto> crearBoleto(@RequestBody Boleto boleto) {
        Boleto nuevoBoleto = boletoService.crearBoleto(boleto);
        return ResponseEntity.ok(nuevoBoleto);
    }

    @GetMapping("/{id}")
    public ResponseEntity<Boleto> obtenerBoletoPorId(@PathVariable Long id) {
        return boletoService.obtenerBoletoPorId(id)
            .map(ResponseEntity::ok)
            .orElse(ResponseEntity.notFound().build());
    }
}
```

Ejemplo de Código (Service)

```
import org.springframework.stereotype.Service;
import java.util.Optional;

@Service
public class BoletoServiceImpl implements BoletoService {

    private final BoletoRepository boletoRepository;

    public BoletoServiceImpl(BoletoRepository boletoRepository) {
        this.boletoRepository = boletoRepository;
    }

    @Override
    public Boleto crearBoleto(Boleto boleto) {
        // Lógica de validación
        return boletoRepository.save(boleto);
    }

    @Override
    public Optional<Boleto> obtenerBoletoPorId(Long id) {
        return boletoRepository.findById(id);
    }
}
```