

Universidad Don Bosco



Primera entrega del proyecto de cátedra

Maestro: Ing. Mario Alvarado

Alumnos:

Daniel Alexander Girón Cornejo GC221469

Cristian Gerardo Ventura Rendón VR221500

Francisco Armando Morales Flores MF230357

Diego Fernando Ruiz Valle RV232739

Materia: Desarrollo de Aplicaciones con Web Frameworks

Fecha de entrega:30/08/2025

Índice

Descripción del Proyecto.....	3
Objetivos.....	3
Objetivo General:	3
Objetivos Específicos:	3
4.1 Gestión de la Integración	4
4.2.Gestión del Alcance	5
4.3 Gestión del Cronograma.....	6
4.4Gestión de Costos.....	8
4.5 Gestión de Calidad.....	9
4.6 Gestión de los Recursos Humanos	9
4.7. Proceso de Comunicación	10
4.8.Gestión de Adquisición y Cierre	11
5. Descripción Técnica 5.1.Prototipo de FrontEnd	11
5.2.Gestión de la Base de Datos	15
6. Resultados y Conclusiones.....	21
7.Anexo.....	22

Descripción del Proyecto

Objetivos

Objetivo General:

Desarrollar un sistema el cual permita gestionar de forma eficiente los procesos de venta y administración de boletos aéreos, lo cual incluye el registro y control de aerolíneas, vuelos, rutas, horarios, tarifas, aviones, tripulación y pasajeros; De la misma manera la administración de reservaciones, pagos, cancelaciones, reclamos y la generación de estadísticas. Con el objetivo de optimizar los tiempos de operación, conseguir una mejor experiencia para el usuario y garantizar confiabilidad de la información obtenida.

Objetivos Específicos:

Creación de módulos específicos para cada acción requerida dentro del sistema tales como reservaciones, pagos, cancelaciones, reclamos y generación de estadísticas, las cuales mejoran la experiencia de uso del usuario y consiga una optimización de la operación.

Diseño e implementación de un sistema el cual consiga la administración completa de los procesos críticos de las aerolíneas, en los cuales se incluye la gestión de vuelos, rutas, horarios, tarifas, aviones, tripulación y pasajeros.

4.1 Gestión de la Integración

La gestión de la integración del proyecto asegura que todos los componentes (código, documentación, pruebas y entregables) trabajen de manera coordinada, optimizando recursos y minimizando errores.

1. Gestión de Recursos y Herramientas

La gestión de la integración del proyecto asegura que todos los componentes (código, documentación, pruebas y entregables) trabajen de manera coordinada, optimizando recursos y minimizando errores.

Gestión de Recursos y Herramientas

Control de Versiones (GitHub): El repositorio en GitHub centraliza todo el código fuente del proyecto, incluyendo BackEnd (Spring Boot), FrontEnd (Angular/React) y documentación.

- o Se empleará Git Flow para organizar el trabajo en ramas (main, develop, feature, hotfix).

- o Pull Requests y revisiones de código obligatorias para mantener la calidad y evitar conflictos.

- o Issues de GitHub para registrar errores y nuevas funcionalidades.

- Gestión de Proyecto (Trello/Notion): Se utilizará un tablero Kanban con columnas de Pendiente, En Proceso, En Revisión, Completado.

- o Trello: asignación de tareas y fechas límite.

- Revisiones de Código:

- o Se realizarán revisiones semanales entre pares antes de aprobar Pull Requests.

- o El Líder de Proyecto y un integrante designado validarán que las prácticas de programación sigan los estándares establecidos.

- o Uso de GitHub para integración continua (compilación automática y pruebas unitarias).

2. Metodología Ágil

Se empleará Scrum como marco de trabajo ágil debido a que permite iteraciones cortas y retroalimentación continua.

- Reuniones:

- o Daily Meeting (15 min, vía Discord).

- o Sprint Planning (inicio de cada sprint).

- o Sprint Review (presentación de avances).

- o Retrospective (mejoras del proceso).

3. Roles de los Integrantes

- **Product Owner:** Define las funcionalidades clave del sistema de boletos aéreos y prioriza el backlog.
- **Scrum Master / Líder de Proyecto:** Facilita la metodología, elimina impedimentos y asegura el cumplimiento de la metodología Scrum.
- **Development Team:**
 - *BackEnd Developer:* Implementa la lógica de negocio, API REST y seguridad.
 - *FrontEnd Developer:* Desarrolla la interfaz de usuario y su integración con la API.
 - *Documentation Manager:* Responsable de los endpoints, manual de usuario y guías técnicas.

QA Tester: Valida la calidad del producto mediante pruebas funcionales, unitarias e integraciones

4.2. Gestión del Alcance

La API REST debe llevar a cabo el proceso completo de la reserva de pasajes aéreos, controlando correctamente la interacción con un posible frontend. Se cuentan con las funcionalidades contempladas: Ingreso de datos del vuelo registro de datos de búsqueda del usuario (ciudad de origen, de llegada, fecha y hora de la salida).

Visualización de opciones disponibles y listado de vuelos que cumplan con las opciones/metadatos que ha introducido el usuario. Cada vuelo se acompañará de datos como aerolínea, ruta, duración, tipo de avión y tarifa. Selección del vuelo y la carga de datos del pasajero también el registro de los datos del pasajero: nombre completo del pasajero, fecha de nacimiento, número de pasaporte y opción de la preferencia del asiento (opcional).

Verificar y resumir la reserva: Verificar que los asientos estén ocupados. Crear la reserva y generar el resumen de la reserva con los datos del vuelo, el pasajero, y el precio total.

Confirmación y pago: Opción para confirmar la reserva y registrar un pago. Relacionar la forma de pago (tarjeta, transferencia, etc). Registrar la forma de pago y su importe acumulado. Cancelaciones o modificaciones: Realizar cancelaciones o modificaciones de la reserva según políticas de la aerolínea.

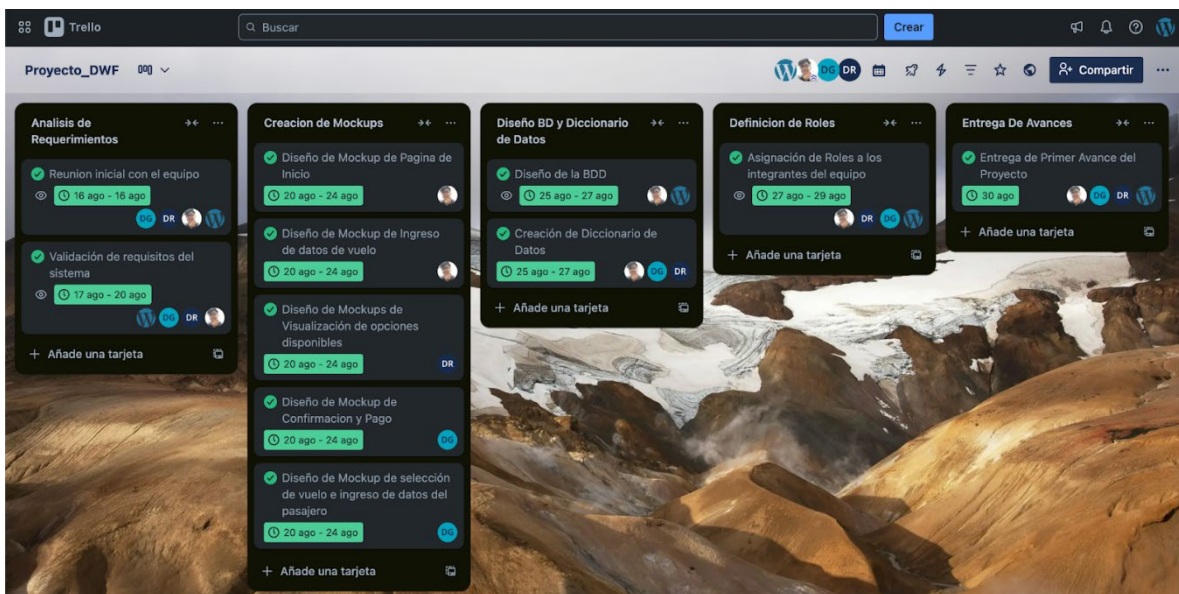
Límites del proyecto

En cuanto al alcance, se limita a la parte de backend sin incluir la parte obligatoria de frontend; el mismo podrá ser desarrollado en cualquier tecnología que desee el estudiante para el consumo de la API. No se prevé de ninguna manera la integración con pasarelas de pago reales; la gestión de pagos se implementará a través de un módulo simulado.

El proyecto no desarrollará un sistema de gestión interna de aerolíneas ni un sistema para la conexión en tiempo real con proveedores de vuelos externos; los datos serán simulados dentro del sistema

De este modo, el proyecto queda bien delimitado en sus funciones básicas y en sus limitaciones, garantizando un desarrollo cerrado y alcanzable en el tiempo determinado

4.3 Gestión del Cronograma



Para un mejor manejo de tiempo y recursos se optó a bien el utilizar un cronograma creado en Trello el cual se dividió en diferentes secciones, se asignaron las personas pertinentes las cuales realizarían cada una de las tareas, siendo así que se logra llevar un tiempo de trabajo óptimo.

Las secciones fueron las siguientes:

- **Análisis de Requerimientos:**

En esta sección se tuvieron las primeras reuniones con el equipo desarrollador en la cual se tomaron decisiones respecto a diseño y funcionamiento.

- **Creación de Mockups:**

Esta sección fue de suma importancia al igual que la siguiente, en esta se tomó el tiempo para realizar los diferentes diseños que serán mostrados y utilizados en el sistema, tales como la Página de Inicio, Ingreso de vuelos, Confirmación y Pago, etc. Esto con la intención de tener un vistazo o una idea cercana a lo que será en un futuro una vez terminado.

- **Diseño de BDD y Diccionario de Datos:**

Como se mencionó anteriormente esta sección fue de mucha importancia puesto que sabemos que al ser un sistema grande debe contar con una Base de Datos robusta la cual pueda manejar una gran cantidad de información de manera ordenada y clara, es por ello que también se realizó un diccionario de datos.

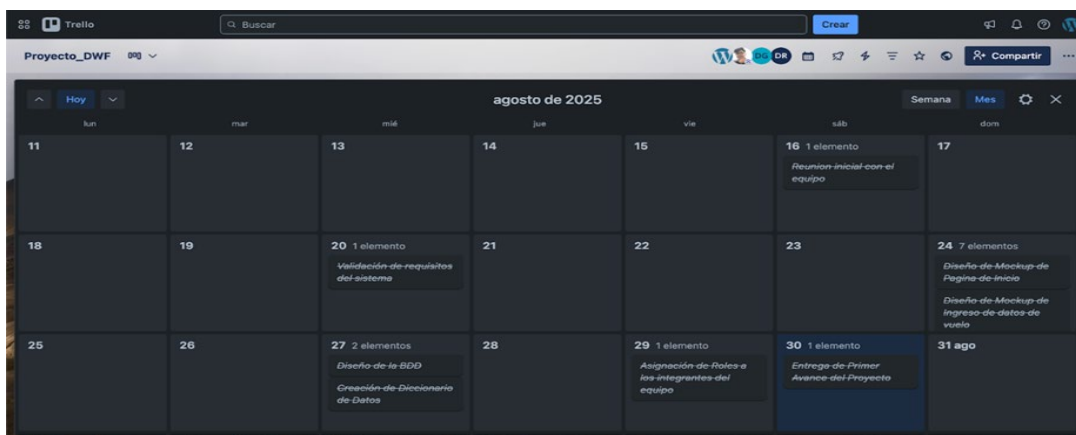
- **Definición de Roles:**

A estas alturas del proyecto se comenzó con la asignación de Roles los cuales fueron asignados de la siguiente manera:

- **Cristian Ventura -> Líder de Proyecto y BackEnd.**
- **Francisco Morales -> Documentación de Endpoints.**
- **Daniel Cornejo -> FrontEnd.**
- **Diego Ruiz -> Soporte de Integración y Pruebas.**
- **Entrega de Avances:**

Como última sección se tiene la entrega de avances, la cual solo fue una meta y límite de fecha de entrega del proyecto.

Calendarización del proyecto



Como apoyo y complemento del punto anterior se creó la calendarización del proyecto para tener una mejor vista de las fechas importantes en la cuales se

hicieron avances para la creación del sistema. En el mismo se reflejan las fechas en la cuales iniciaban cada una de las actividades que se mostraron en el cronograma anterior, siendo las siguientes:

- **16 de Agosto del 2025:** Reunión inicial con el equipo.
- **20 de Agosto del 2025:** Validación de requisitos del sistema.
- **24 de Agosto del 2025:** Comienzo de la creación de los Mockups.
- **27 de Agosto del 2025:** Diseño de la Base de Datos y Creación del Diccionario de Datos.
- **29 de Agosto del 2025:** Asignación de Roles a cada uno de los integrantes del equipo.
- **30 de Agosto del 2025:** Entrega del primer avance del proyecto.

4.4 Gestión de Costos

Para el desarrollo del sistema se utilizarán herramientas gratuitas o de bajo costo, incluyendo Java, Spring Boot y entornos de desarrollo como Eclipse IDE o IntelliJ IDEA Community, que permitirán programar, probar y mantener el código de manera eficiente. Además, se empleará GitHub para el control de versiones y la colaboración del equipo, mientras que la infraestructura y el despliegue se realizarán en servicios de nube como Heroku o Google Cloud, asegurando que la aplicación funcione correctamente en entornos de prueba y producción.

Presupuesto estimado por componente:

Licencias y recursos profesionales: \$50 USD

Control de versiones y colaboración en GitHub: \$30 USD

Servicios en la nube (Heroku o Google Cloud): \$40 USD

Herramientas adicionales, plugins o certificados SSL: \$20 USD

El costo total estimado para la primera fase del proyecto es de \$140 USD. Este presupuesto incluye todos los recursos necesarios para el desarrollo, despliegue inicial, pruebas y ajustes menores de la aplicación. Cubre tanto las herramientas de

programación y control de versiones, como la infraestructura en la nube y los complementos adicionales, asegurando que el equipo pueda trabajar de manera eficiente y mantener la calidad del proyecto durante toda esta primera etapa.

4.5 Gestión de Calidad

La calidad del código se garantizará mediante revisiones periódicas de los cambios realizados por cada desarrollador. Cada modificación será enviada a un repositorio en GitHub, donde otro miembro del equipo la revisará antes de integrarse al proyecto principal. Esto permitirá detectar errores temprano, asegurar consistencia en la programación y mantener un código limpio, organizado y fácil de mantener. Además, se fomentará la documentación de cada módulo, facilitando futuras modificaciones y ampliaciones del sistema, así como la transferencia de conocimiento dentro del equipo.

Asimismo, se implementarán herramientas de análisis de código y estándares de programación en Java y Spring Boot. Se usarán linters y validaciones automáticas que revisen estilo, posibles errores y redundancias, asegurando que el código cumpla con criterios de calidad técnica. También se realizan pruebas periódicas de funcionalidad para verificar que las operaciones principales de la aplicación, como reservas, pagos y gestión de vuelos, funcionen correctamente. Estas medidas garantizarán una aplicación estable, confiable y preparada para futuras actualizaciones o escalabilidad.

4.6 Gestión de los Recursos Humanos

La correcta gestión de los recursos humanos es clave para el éxito del proyecto, ya que permite definir responsabilidades claras, fomentar el trabajo en equipo y garantizar una comunicación fluida entre los integrantes.

a) Asignación de Roles

Dentro del equipo de desarrollo del **sistema de venta y administración de boletos aéreos**, se han asignado los siguientes roles:

- Líder de Proyecto-Cristian Ventura:
Encargado de la planificación general, coordinación de tareas y toma de decisiones estratégicas. Supervisa los avances, resuelve bloqueos y garantiza el cumplimiento de los objetivos en tiempo y forma.
- Documentación de Endpoints (BackEnd – Java Spring Boot)-Francisco Morales:

Responsable de la documentación técnica de la API REST, describiendo de manera clara cada endpoint (parámetros, métodos, respuestas y códigos de error). Esta documentación permitirá que el equipo de FrontEnd consuma los servicios sin inconvenientes.

- Backend (Java Spring Boot)-Cristian Ventura:

Encargado de desarrollar la lógica de negocio del sistema, incluyendo la gestión de usuarios, vuelos, reservas y pagos. Se asegura de que la API REST cumpla con las reglas del negocio, la seguridad y el rendimiento esperado.

- FrontEnd (Angular/React)-Daniel Girón:

Responsable del diseño y desarrollo de la interfaz de usuario, garantizando una experiencia intuitiva para los clientes que compren boletos en línea. También asegura la correcta integración con la API documentada.

- Soporte de Integración y Pruebas-Daniel Girón:

Encargado de validar la interoperabilidad entre FrontEnd y Backend, además de realizar pruebas funcionales y de usabilidad, reportando errores y proponiendo mejoras.

4.7. Proceso de Comunicación

La comunicación es un eje indispensable para llevar a cabo la adecuada coordinación y el éxito del desarrollo de la API REST. Para ello, se establecen medios oficiales y una periodicidad de actualizaciones que permitan mantener un flujo continuo de la información tanto con el docente como entre el profesorado del equipo.

Informes Semanales de Progreso: El coordinador del proyecto es la persona que presenta los avances en reuniones semanales, en las cuales queda documentado el estado en el que se encuentran las tareas y posibles dificultades.

Canales de Comunicación:

WhatsApp: usado para la comunicación rápida y para la coordinación instantánea entre los miembros del equipo.

GitHub: principal plataforma para la gestión del código fuente, control de versiones y el registro de los avances en el repositorio del proyecto.

Discord: utilizado para las reuniones semanales del equipo y las sesiones de trabajo colaborativo en tiempo real.

De esta manera se garantiza un flujo de información adecuado y organizado que facilita la resolución de dudas y la toma de decisiones que se hace durante todo el ciclo de desarrollo.

4.8.Gestión de Adquisición y Cierre

https://github.com/Fran503chichico/Proyecto_DWF.git

5. Descripción Técnica - 5.1.Prototipo de FrontEnd

Pagina principal





Cancelación y Modificación:

Ingreso datos de vuelo
Vuelo disponibles
Pago de vuelo
Cancelacion de vuelo

Cancelación de vuelo

Si esta seguro de eliminar su vuelo, ingrese su numero de reserva:

Numero de reserva:

Aerolinea:

Origen:

Destino:

Cliente:

[Cancelar Viaje](#)

ADVERTENCIA

Recuerde que al cancelar su vuelo esta de acuerdo a los términos y condiciones de cancelación según la aerolínea con la que viajaría; Al cancelar usted acepta dichos términos y condiciones.

Modificación de Datos

Acontinuation puede modificar algunos datos de su reserva, estos estaran sujetos a cambios en la tarifa, ingrese su numero de reserva:

Numero de reserva:

Numero de asientos:

Tipo de clase:

[Modificar Viaje](#)

ADVERTENCIA

Recuerde que al modificar su vuelo esta de acuerdo a los términos y condiciones según la aerolínea con la que viajaría; Al modificar estos datos usted acepta dichos términos y condiciones.

Siguenos en nuestras redes sociales para tener mas informacion

Verificación y Resumen de reserva



[Ingrese datos de vuelo](#) | [Vuelo disponibles](#) | [Pago de vuelo](#) | [Cancelacion de vuelo](#) | ...

Verifica los datos de tu vuelo

A continuación puedes verificar la información de tu vuelo, unicamente ingresa tu numero de reserva:

Numero de reserva:

Aerolínea:	<input type="text"/>	Tipo de Avión:	<input type="text"/>
Origen:	<input type="text"/>	Asignación de asientos:	<input type="text"/>
Destino:	<input type="text"/>	Numeros de asientos:	<input type="text"/>
Hora de Salida:	<input type="text"/>	Tipo de clase:	<input type="text"/>
Hora Aprox. Llegada:	<input type="text"/>	Tarifa:	<input type="text"/>

Síguenos en nuestras redes sociales para tener mas informacion  

Visualización de opciones disponibles

[Ingrese datos del pasajero](#) | [Verificacion de vuelo](#) | [modificacion de vuelo](#) | ...

Aerolienas disponibles

Aerolínea	Ruta	Duración	Avión	Tarifa	Acción
Avianca	SAL → LAX	4h 55m	Airbus A320	\$420 USD	[RESERVAR]
American Airlines	SAL → DFW	3h 20m	Boeing 737-800	\$380 USD	[RESERVAR]
Volaris	SAL → MEX	2h 00m	Airbus A320neo	\$250 USD	[RESERVAR]
Copa Airlines	SAL → PTY	2h 15m	Boeing 737-800	\$270 USD	[RESERVAR]
Spirit Airlines	SAL → FLL	2h 35m	Airbus A320	\$230 USD	[RESERVAR]

Síguenos en nuestras redes sociales para tener mas informacion  

Pago de vuelo

Frame 37

Ingrese datos de vuelo | Vuelo disponibles | Pago de vuelo | Cancelacion de vuelo ...

VISA  Pago de vuelo

Titular de cuenta

Ingresar número de tarjeta

Ingresar código de vuelo

Valida hasta Código de seguridad

Fecha de salida del vuelo Fecha de regreso del vuelo

[Confirmar pago](#)

Síguenos en nuestras redes sociales para tener más información  

Selección del vuelo e ingreso de datos del pasajero

Ingrese datos de vuelo | Vuelo disponibles | Pago de vuelo | Cancelacion de vuelo ...


Nombre



Fecha de nacimiento

Número de pasaporte

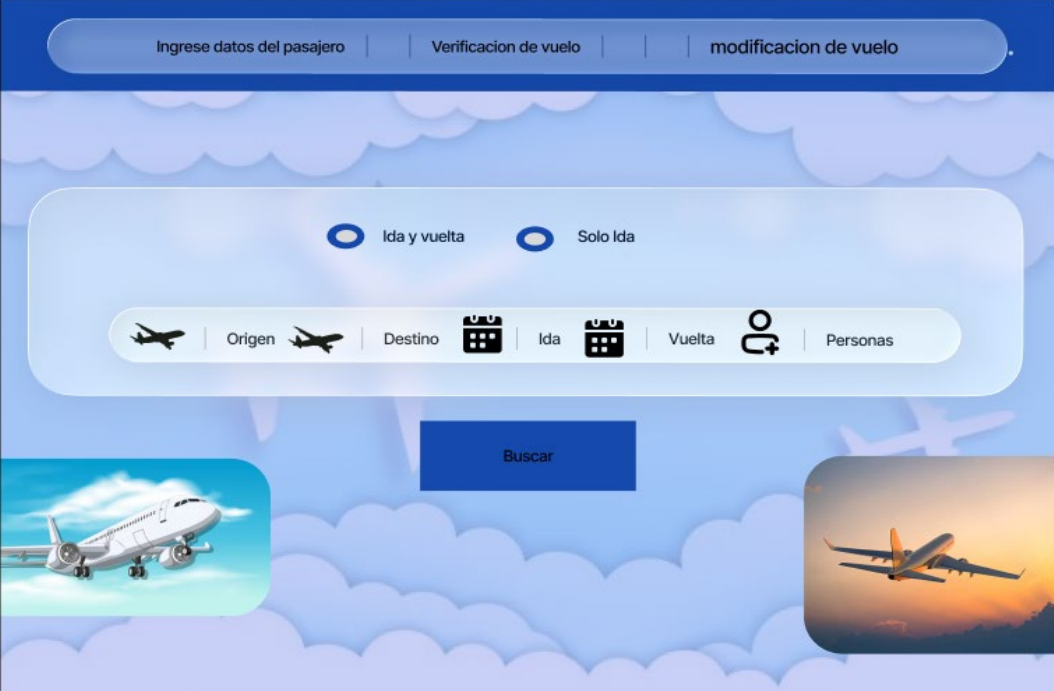
Preferencia de asiento

[Confirmar](#)



Síguenos en nuestras redes sociales para tener más información  

Ingreso de datos del vuelo



The screenshot shows a web interface for flight booking. At the top, there is a navigation bar with three tabs: 'Ingrese datos del pasajero', 'Verificacion de vuelo', and 'modificacion de vuelo'. The first tab is active. Below the navigation bar, there is a form with a light blue background and a cloud pattern. The form has two radio buttons for 'Ida y vuelta' (selected) and 'Solo Ida'. Below these, there are input fields for 'Origen' (with an airplane icon), 'Destino' (with an airplane icon), 'Ida' (with a calendar icon), 'Vuelta' (with a calendar icon), and 'Personas' (with a group of people icon). A blue 'Buscar' button is centered below the input fields. The background of the form features two images of airplanes: one on the left and one on the right.

5.2.Gestión de la Base de Datos

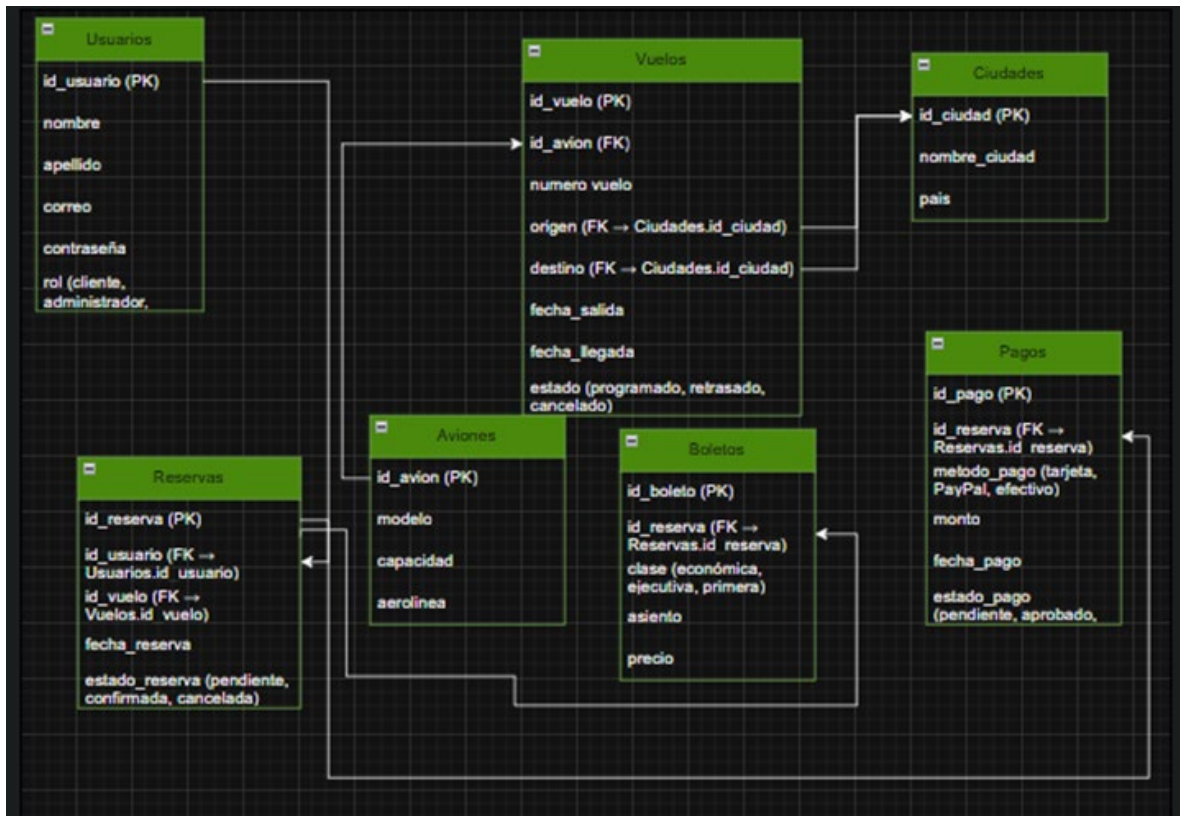
Gestión de la Base de Datos

La base de datos es el núcleo del sistema de venta y administración de boletos aéreos, ya que almacena la información de vuelos, clientes, reservas y transacciones de pago. Se diseñó siguiendo un modelo relacional que permite garantizar integridad, consistencia y escalabilidad.

a) Estructura de la Base de Datos

Modelo Relacional

Las principales tablas y sus relaciones son:



b) Relaciones entre las Tablas

- Un usuario puede tener múltiples reservas (1:N).
- Una reserva puede generar uno o varios boletos (1:N).
- Un vuelo puede estar asociado a muchas reservas, pero cada reserva corresponde a un solo vuelo (1:N).
- Cada reserva tiene un pago asociado (1:1).
- Los vuelos están relacionados con ciudades de origen y destino (N:1).
- Cada vuelo utiliza un avión asignado (N:1).

c) Diagrama Entidad-Relación (E-R)

- Usuarios (1) — (N) Reservas
- Reservas (1) — (N) Boletos
- Reservas (1) — (1) Pagos
- Vuelos (1) — (N) Reservas
- Vuelos (N) — (1) Aviones

- Vuelos (N) — (1) Ciudades (Origen)
- Vuelos (N) — (1) Ciudades (Destino)

Diccionario de Datos

Tabla: Usuarios

Campo	Tipo de Dato	Tamaño	Descripción	Valores Permitidos	Restricciones	Relaciones
id_usuario	INT	11	Identificador único del usuario	Numérico autoincremental	PK, NOT NULL	Relación con <i>Reservas</i>
nombre	VARCHAR	50	Nombre del usuario	Texto	NOT NULL	-
apellido	VARCHAR	50	Apellido del usuario	Texto	NOT NULL	-
correo	VARCHAR	100	Correo electrónico del usuario	Formato email válido	UNIQUE, NOT NULL	-
contraseña	VARCHAR	255	Contraseña cifrada	Texto cifrado	NOT NULL	-
rol	ENUM	-	Rol del usuario en el sistema	{cliente, administrador, agente}	NOT NULL	-

Tabla: **Ciudades**

Campo	Tipo de Dato	Tamaño	Descripción	Valores Permitidos	Restricciones	Relaciones
id_ciudad	INT	11	Identificador único de la ciudad	Numérico autoincremental	PK, NOT NULL	Relación con <i>Vuelos</i>
nombre_ciudad	VARCHAR	100	Nombre de la ciudad	Texto	NOT NULL	-
pais	VARCHAR	100	País de la ciudad	Texto	NOT NULL	-

Tabla: **Aviones**

Campo	Tipo de Dato	Tamaño	Descripción	Valores Permitidos	Restricciones	Relaciones
id_avion	INT	11	Identificador único del avión	Numérico autoincremental	PK, NOT NULL	Relación con <i>Vuelos</i>
modelo	VARCHAR	50	Modelo del avión	Texto	NOT NULL	-
capacidad	INT	4	Número de asientos	Entero positivo	NOT NULL	-
aerolinea	VARCHAR	100	Aerolínea propietaria	Texto	NOT NULL	-

Tabla: Reservas

Campo	Tipo de Dato	Tamaño	Descripción	Valores Permitidos	Restricciones	Relaciones
id_reserva	INT	11	Identificador único de la reserva	Numérico autoincremental	PK, NOT NULL	Relación con <i>Boletos y Pagos</i>
id_usuario	INT (FK)	11	Usuario que realizó la reserva	Id de <i>Usuarios</i>	NOT NULL	Usuarios.id_usuario
id_vuelo	INT (FK)	11	Vuelo reservado	Id de <i>Vuelos</i>	NOT NULL	Vuelos.id_vuelo
fecha_reserva	DATE TIME	-	Fecha de la reserva	Fecha válida	NOT NULL	-
estado_reserva	ENUM	-	Estado de la reserva	{pendiente, confirmada, cancelada}	NOT NULL	-

Tabla: Boletos

Campo	Tipo de Dato	Tamaño	Descripción	Valores Permitidos	Restricciones	Relaciones
id boleto	INT	11	Identificador único del boleto	Numérico autoincremental	PK, NOT NULL	-
id_reserva	INT (FK)	11	Reserva a la que pertenece	Id de <i>Reservas</i>	NOT NULL	Reservas.id_reserva
asiento	VARCHAR	5	Número de asiento	Ej: A12, B4	UNIQUE por vuelo, NOT NULL	-
clase	ENUM	-	Clase del boleto	{económica, ejecutiva, primera}	NOT NULL	-
precio	DECIMAL	8,2	Precio del boleto	Valor monetario > 0	NOT NULL	-

Tabla: Pagos

Campo	Tipo de Dato	Tamaño	Descripción	Valores Permitidos	Restricciones	Relaciones
id_pago	INT	11	Identificador único del pago	Numérico autoincremental	PK, NOT NULL	-
id_reserva	INT (FK)	11	Reserva asociada al pago	Id de <i>Reservas</i>	NOT NULL	Reservas.id_reserva
metodo_pago	ENUM	-	Forma de pago	{tarjeta, PayPal, efectivo}	NOT NULL	-
monto	DECIMAL	8,2	Monto total del pago	> 0	NOT NULL	-
fecha_pago	DATETIME	-	Fecha de pago	Fecha válida	NOT NULL	-
estado_pago	ENUM	-	Estado del pago	{pendiente, aprobado, rechazado}	NOT NULL	-

Tabla: usuario

Campo	Tipo de Dato	Tamaño	Descripción	Valores Permitidos	Restricciones	Relaciones
id_usuario	INT	11	Identificador único del usuario	Numérico autoincremental	PK, NOT NULL	Relación con <i>Reservas</i>
nombre	VARCHAR	50	Nombre del usuario	Texto	NOT NULL	-
apellido	VARCHAR	50	Apellido del usuario	Texto	NOT NULL	-
correo	VARCHAR	100	Correo electrónico del usuario	Formato email válido	UNIQUE, NOT NULL	-
contraseña	VARCHAR	255	Contraseña cifrada	Texto cifrado	NOT NULL	-
rol	ENUM	-	Rol del usuario en el sistema	{cliente, administrador, agente}	NOT NULL	-

6. Resultados y Conclusiones

Resultados:

Cobertura funcional esperada: El sistema deberá ser capaz de gestionar aerolíneas, aeropuertos, rutas, aeronaves, tripulación y pasajeros infinitos; así como soportar búsqueda de vuelos, selección de asiento, creación de reserva, pago, cancelación y gestión de reclamos infinitos.

Consistencia de datos: Con restricciones relacionales y controles transaccionales permiten que un asiento no pueda asignarse a su vez a una reserva y que las operaciones críticas sean atómicas.

Control de concurrencia: En teoría, el uso de bloqueos/locks o mecanismos optimistas evita el double-booking (overbooking) ante accesos concurrentes.

Accesibilidad y capacidad de respuesta: La indexación y el diseño modular garantizan respuestas rápidas en búsquedas y un tratamiento eficaz de reservas; la capacidad operativa real dependerá de la infraestructura y su dimensionado con el hardware adecuado.

Seguridad y acceso: Un modelo de roles proporciona una separación y especificación de los permisos (administración, personal de aerolínea, agentes, pasajeros, soporte) para garantizar que las operaciones sensibles sólo sean ejecutadas por los actores adecuados.

Conclusión

Esperamos que nuestra API REST sea efectiva y logre simular de forma integral el funcionamiento de una aerolínea; aplicaremos todo lo aprendido en el curso, trabajando en equipo y bajo la guía de nuestro docente, para garantizar que cada funcionalidad opere de la mejor manera posible y que el proyecto se consolide como una experiencia de aprendizaje sólida y práctica.

7.Anexo

Spring Boot – Tutorial REST (Spring.io):

<https://spring.io/guides/tutorials/rest>

Plataforma UDB Virtual – Aula Digital:

<https://www.udbvirtual.edu.sv/auladigital/login/index.php>

Repositorio del Proyecto en GitHub:

https://github.com/Fran503chichico/Proyecto_DWF

Tablero de Tareas en Trello:

<https://trello.com/b/cS3G3fR6/proyectodwf>