



sistema de gestión de boletos aéreos – Airtickets

Integrantes:

Javier Stanley Valladares Valladares	VV230315	Líder de Proyecto – Desarrollador Backend – Documentador
Marvin Rene Martinez Gomez	MG231425	Encargado de Base de Datos – Documentador
Jonathan Josue Cardoza Perez	CP230528	Desarrollador Frontend – Documentador
Andrea Lisseth Demata Moreno	DM221223	Encargado de Base de Datos – Documentador

Fecha de entrega: 30 de agosto de 2025

Docente: Mario Alvarado

asignatura: Desarrollo de Aplicaciones con Web Frameworks

Enlace de figma Mockups

<https://www.figma.com/design/QFQtnNAFYdEt8uNgz0BBq/Sin-t%C3%ADtulo?node-id=105-12&t=KxXe9b7im5lgINFC-0>

Enlace de Notion Cronograma Calendario y Hitos

https://www.notion.so/25e278d9a20f800dbf57ce5417a876bf?v=25e278d9a20f8015b933000cfa9aee90&source=copy_link

Enlace de Github

https://github.com/Valladaresst/DWF_Proyecto_de_catedra.git

Indice

3. Descripción del Proyecto	4
3.1 Objetivo General	4
3.2 Objetivos Específicos	4
4. Gestión Integral del Proyecto	4
4.1 Gestión de la Integración	4
4.2 Gestión del Alcance	5
4.3 Gestión del Cronograma	6
4.4 Gestión de Costos	11
1. Presupuesto Técnico (Infraestructura y Herramientas)	11
2. Presupuesto de Recursos Humanos (Estimación de horas-hombre)	12
4.5 Gestión de la Calidad	13
a) Revisión de Código	13
b) Estándares de Codificación	13
c) Pruebas y Validación	13
4.6 Gestión de los Recursos Humanos	14
b) Colaboración y Comunicación	14
4.7 Gestión de la Comunicación	15
a) Informes Semanales de Progreso	15
b) Canales de Resolución de Dudas	15
4.8 Gestión de Adquisición y Cierre	16
a) Documentación Final	16
b) Verificación de Entregables	16
c) Cierre del Proyecto	16
5. Descripción Técnica	17
5.1. Prototipo de FrontEnd	17
Tabla: Países	22
Tabla: Ciudades	23
Tabla: Aeropuertos	23
Tabla: Aerolíneas	24
Tabla: Vuelos	24
Tabla: Pasajeros	25
Tabla: Reservaciones	25
Tabla: Pagos	26

Tabla: Reclamos	27
6. Resultados y Conclusiones	31
a) Resultados Obtenidos	31
b) Conclusiones	31
7. Anexos y Documentación de Apoyo	32

3. Descripción del Proyecto

3.1 Objetivo General

Desarrollar un sistema de gestión de boletos aéreos que permita a los usuarios consultar vuelos disponibles, realizar reservaciones, gestionar pagos y cancelaciones, así como facilitar la administración de aerolíneas, aviones, rutas y tripulación. El propósito es ofrecer una solución eficiente y confiable que optimice los procesos de compra y administración de boletos aéreos, mejorando la experiencia de los pasajeros y la operatividad de las aerolíneas.

3.2 Objetivos Específicos

- Permitir el **registro y gestión de aerolíneas** dentro del sistema.
- Implementar la **creación de vuelos, rutas, horarios y tarifas** para organizar la oferta disponible.
- Administrar la **información de aviones y tripulación** asociada a cada vuelo.
- Facilitar el **registro de pasajeros** con sus datos personales y preferencias.
- Gestionar el **proceso de reservación, confirmación y pago de boletos aéreos** con diferentes métodos de pago.
- Incorporar funcionalidades para la **cancelación o modificación de reservas**, según políticas de la aerolínea.
- Procesar y registrar **reclamos de los usuarios**, asegurando un seguimiento adecuado.
- Generar **estadísticas básicas** del sistema (reservas realizadas, cancelaciones, ingresos, etc.) para apoyar la toma de decisiones.

4. Gestión Integral del Proyecto

4.1 Gestión de la Integración

La integración del proyecto se gestionará mediante una combinación de herramientas y prácticas que faciliten la colaboración del equipo y el control de los entregables.

- **Control de versiones:** Se utilizará **GitHub** como repositorio principal del código fuente, lo que permitirá mantener un historial de cambios, trabajo colaborativo entre los integrantes y revisiones constantes de código mediante al pull request
- **Gestión de tareas y seguimiento:** El equipo empleará **Notion** como herramienta principal para la planificación, asignación de tareas y seguimiento del progreso. Esta plataforma permitirá visualizar el avance de cada fase del proyecto mediante tableros tipo *Kanban*, facilitando el control de pendientes

y prioridades

- **Metodología ágil:** Se aplicará la metodología **Scrum**, trabajando en **sprints** semanales con reuniones de seguimiento para evaluar avances, identificar impedimentos y planificar las siguientes actividades.
- **Revisiones de código:** Cada cambio realizado en el repositorio será revisado por al menos un integrante adicional antes de ser integrado en la rama principal, garantizando la calidad y consistencia del código.

Roles de los integrantes

- **Javier Stanley Valladares Valladares – Jonathan Josue Cardoza Perez – Marvin Rene Martinez Gomez – Andrea Lisseth Demata Moreno** – Documentación / QA: Redacta la documentación y revisa calidad del código y entregables.
-
- **Javier Stanley Valladares Valladares** – Líder de Proyecto / Coordinador: Responsable de la integración general, comunicación con el docente y organización del equipo.
- **Javier Stanley Valladares Valladares** – Desarrollador Backend: Encargado de la API REST en Java Spring Boot y de la lógica de negocio.
- **Jonathan Josue Cardoza Perez** – Desarrollador Frontend / Prototipos: Diseña las interfaces de usuario y en la segunda fase implementará el consumo de la API.
- **Marvin Rene Martinez Gomez – Andrea Lisseth Demata Moreno** – Encargado de Base de Datos y Documentación / QA: Diseña la base de datos.

4.2 Gestión del Alcance

El sistema de gestión de boletos aéreos abarcará un conjunto de funcionalidades clave que garanticen la operatividad mínima del proyecto y la experiencia adecuada de los usuarios.

Funcionalidades esperadas

- **Gestión de aerolíneas:** Registro, actualización y consulta de aerolíneas disponibles.
- **Creación y administración de vuelos:** Inclusión de rutas, horarios, duración y tarifas.
- **Gestión de aviones y tripulación:** Registro de aeronaves, capacidad y asignación de personal de vuelo.
- **Registro de pasajeros:** Almacenamiento de información personal, pasaporte y preferencias.
- **Reservación de boletos:** Selección de vuelo, verificación de disponibilidad de asientos y generación de la reserva.
- **Procesamiento de pagos:** Registro de pagos por distintos métodos (tarjeta, transferencia, etc.).

- **Cancelaciones y modificaciones:** Opción de cancelar o modificar una reserva según políticas de la aerolínea.
- **Gestión de reclamos:** Registro y seguimiento de solicitudes o inconformidades de los usuarios.
- **Estadísticas básicas:** Número de reservas, cancelaciones, pasajeros registrados, etc.

Límites del proyecto

- El sistema **no incluirá pasarelas de pago reales**; se simularán los métodos de pago en esta fase.
- La aplicación **no contempla integraciones con sistemas externos de aerolíneas reales**, sino que será un sistema autónomo de gestión.
- El **frontend de la primera fase será un prototipo de interfaz** (maquetas), no una aplicación completamente funcional.
- La seguridad se implementará de manera básica, sin integración con sistemas de autenticación corporativos avanzados.

4.3 Gestión del Cronograma

El cronograma del proyecto se ha estructurado siguiendo la metodología ágil Scrum, organizando las actividades en sprints quincenales. Cada sprint tiene un inicio y un fin definidos, así como responsables y entregables concretos.

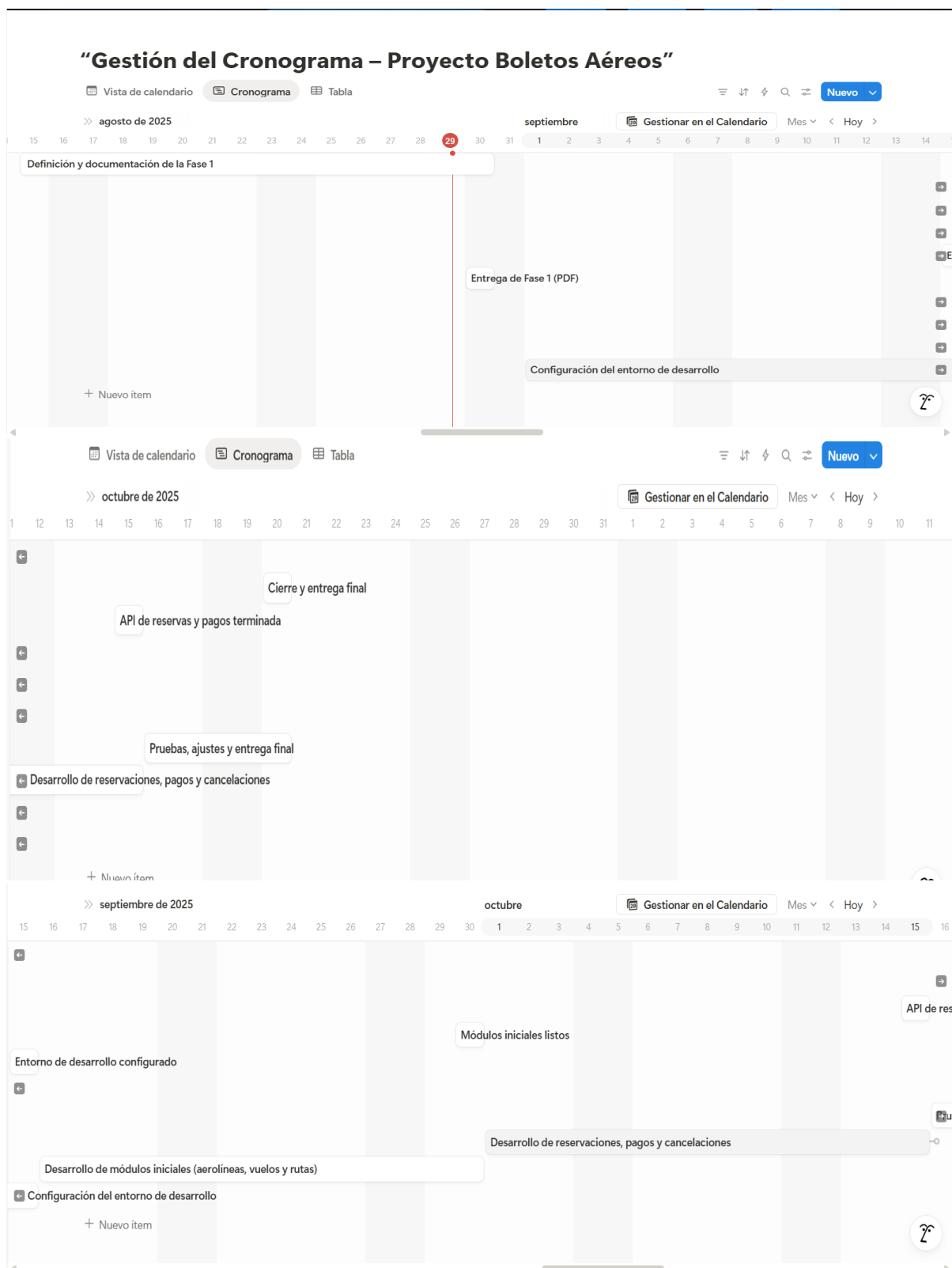
Los hitos se han identificado como fechas específicas que marcan logros clave dentro del desarrollo, tales como la entrega del documento de la Fase 1 (30/08/2025), la configuración del entorno de desarrollo (15/09/2025), la finalización de módulos iniciales (30/09/2025), la integración de las funcionalidades de reservas y pagos (15/10/2025) y el cierre del proyecto (20/10/2025).

El cronograma se gestiona en **Notion**, utilizando la vista de **Timeline** para visualizar la duración de los sprints y la vista de **Calendar** para ubicar los hitos y fechas clave. Esto permite al equipo tener un control claro sobre las actividades y plazos, asegurando el cumplimiento de los objetivos en cada fase del proyecto.

Tipo	Nombre	Fecha Inicio	Fecha Fin	Responsables	Entregable / Logro
Sprint	Sprint 1 – Definición y documentación	15/08/2025	30/08/2025	Coordinador + Todo el equipo	Documento Fase 1 (perfil del proyecto, objetivos, prototipos, BD)
Hito	Entrega de Fase 1 (PDF)	30/08/2025	30/08/2025	Todo el equipo	Documento subido al Aula Digital

Sprint	Sprint 2 – Configuración del entorno	01/09/2025	15/09/2025	Backend + BD	Proyecto base en Spring Boot, repositorio GitHub, BD inicial
Hito	Entorno de desarrollo configurado	15/09/2025	15/09/2025	Backend	Proyecto funcional en GitHub
Sprint	Sprint 3 – Módulos iniciales (aerolíneas y vuelos)	16/09/2025	30/09/2025	Backend + BD	API REST inicial para aerolíneas y vuelos
Hito	Módulos iniciales listos	30/09/2025	30/09/2025	Backend + BD	API funcionando
Sprint	Sprint 4 – Reservas, pagos y cancelaciones	01/10/2025	15/10/2025	Backend + Frontend	API REST completa de reservas, pagos y cancelaciones
Hito	API de reservas y pagos terminada	15/10/2025	15/10/2025	Backend + Frontend	Módulo integrado
Sprint	Sprint 5 – Pruebas y cierre del proyecto	16/10/2025	20/10/2025	Todo el equipo	Sistema probado y documentación final
Hito	Cierre y entrega final	20/10/2025	20/10/2025	Todo el equipo	Sistema completo + Documentación final









10	11	12	13	14	15	16
					Definición y documentación de la Fas... 2	
17	18	19	20	21	22	23
Definición y documentación de la Fase 1 2						
24	25	26	27	28	29	30
Definición y documentación de la Fase 1 2						
					Entrega de F... 1	
31	1 sept	2	3	4	5	6
	Configuración del entorno de desarrollo 2					
7	8	9	10	11	12	13
Configuración del entorno de desarrollo 2						
14	15	16	17	18	19	20
Configuración del entorno de desarr... 2		Desarrollo de módulos iniciales (aerolíneas, vuelos y rutas) 2				
	Entorno de d... 1					
21	22	23	24	25	26	27
Desarrollo de módulos iniciales (aerolíneas, vuelos y rutas) 2						
28	29	30	1 oct	+	2	3
Desarrollo de módulos iniciales (aerolíneas, vuelos y rutas) 2		Desarrollo de reservaciones, pagos y cancelaciones 2				
		Módulos inic... 1				
5	6	7	8	9	10	11
Desarrollo de reservaciones, pagos y cancelaciones 2						
12	13	14	15	16	17	18
Desarrollo de reservaciones, pagos y cancelaciones 2			Pruebas, ajustes y entrega final 1			
		API de reserv... 1				
19	20	21	22	23	24	25
Pruebas, ajustes y entrega final 1						
	Cierre y entr... 1					



“Gestión del Cronograma – Proyecto Boletos Aéreos”

Vista de calendario Cronograma **Tabla**

⌵ ⬆ ⚡ 🔍 ⚙ Nuevo ▾

Nombre	Etiquetas	Fecha	Responsable	
Definición y documentación de la Fase 1  2	stanley valladares	15 de agosto de 2025 → 30 de agosto de 2025	stanley valladares Jonathan Cardoza MRMG Lisseth Demata	
Entrega de Fase 1 (PDF)  1	stanley valladares	30 de agosto de 2025 → 30 de agosto de 2025	stanley valladares	
Configuración del entorno de desarrollo  2		1 de septiembre de 2025 → 15 de septiembre de 2025	stanley valladares MRMG	
Entorno de desarrollo configurado  1	stanley valladares	15 de septiembre de 2025 → 15 de septiembre de 2025	stanley valladares	
Desarrollo de módulos iniciales (aerolíneas, vuelos y rutas)  2		16 de septiembre de 2025 → 30 de septiembre de 2025	stanley valladares MRMG	
Módulos iniciales listos  1	stanley valladares Marvin Martinez	30 de septiembre de 2025 → 30 de septiembre de 2025	stanley valladares MRMG	
Desarrollo de reservaciones, pagos y cancelaciones  2		1 de octubre de 2025 → 15 de octubre de 2025	Jonathan Cardoza stanley valladares	
Cierre y entrega final  1	stanley valladares Marvin Martinez Jonathan Cardoza	20 de octubre de 2025 → 20 de octubre de 2025	stanley valladares Jonathan Cardoza MRMG Lisseth Demata	

+ Nueva página

Notion: Espacio de trabajo colaborativo donde se gestionó el cronograma del proyecto, los sprints, hitos y tareas del equipo.

Link de Notion

https://www.notion.so/25e278d9a20f800dbf57ce5417a876bf?v=25e278d9a20f8015b933000cfa9aee90&source=copy_link

4.4 Gestión de Costos

1. Presupuesto Técnico (Infraestructura y Herramientas)

Recurso	Descripción	Costo (académico / gratuito)	Costo (productivo / real)
Java + Spring Boot	Lenguaje y framework backend (open source).	\$0	\$0
Frontend Web (HTML, CSS, JS + React)	Desarrollo de la interfaz web de usuario. Todas las tecnologías son open source.	\$0	\$0
MySQL	Base de datos relacional. Gratuita en local; en nube administrada (ej. AWS RDS).	\$0	\$15/mes
Servidor en la nube (AWS EC2 / Azure / GCP)	Hosting del backend y frontend.	\$0 (local)	\$25–30/mes
GitHub	Control de versiones y repositorio. Plan Free en entorno académico.	\$0	\$4/usuario/mes (plan Team)
Notion	Gestión de proyecto, cronograma y tareas. Plan gratuito para estudiantes.	\$0	\$8/usuario/mes (plan Plus)
Figma	Diseño y prototipado de interfaces. Plan Free para estudiantes.	\$0	\$12/mes (plan Pro)
IntelliJ IDEA	IDE de desarrollo (Community Edition gratuita).	\$0	\$0

2. Presupuesto de Recursos Humanos (Estimación de horas-hombre)

Rol	Cantidad de personas	Horas estimadas	Costo por hora (estimado)	Subtotal
Coordinador / Scrum Master	1	40h	\$10	\$400
Desarrollador Backend (Java + Spring Boot)	2	80h c/u (160h total)	\$10	\$1,600
Desarrollador Frontend Web (HTML, CSS, JS, React)	1	60h	\$10	\$600
Diseñador UI/UX (Figma)	1	30h	\$8	\$240
Total estimado	4	290h	—	\$2,840

Escenario académico (con planes gratuitos): \$0.

Escenario productivo (infraestructura mínima + suscripciones + horas de trabajo): alrededor de \$2,900 USD por todo el proyecto (considerando un desarrollo de 2 meses).

4.5 Gestión de la Calidad

Para asegurar la calidad del software en el proyecto, se implementarán procesos de control de calidad del código, el cumplimiento de estándares de programación y la utilización de herramientas colaborativas que permitan mantener un desarrollo ordenado y eficiente.

a) Revisión de Código

- Todo el código será administrado en GitHub, utilizando ramas por funcionalidad (feature branches).
- Las nuevas funcionalidades deberán integrarse a la rama principal (main) mediante pull requests (PRs).
- Cada PR será revisado por al menos un integrante del equipo antes de ser aprobado, siguiendo un esquema de Code Review para detectar errores y sugerir mejoras.
- Se verificará que el código cumpla con los estándares acordados y no introduzca errores en el sistema.

b) Estándares de Codificación

- Se seguirán buenas prácticas de programación en Java:
 - Uso de nombres descriptivos para variables, métodos y clases.
 - Aplicación de los principios SOLID para mantener un diseño limpio.
 - Organización del código en paquetes de acuerdo con la arquitectura definida (controladores, servicios, repositorios, etc.).
 - Comentarios claros y concisos para documentar métodos y clases.
- Para el frontend (HTML, CSS, JS/React), se seguirán convenciones estándar de identificación, uso de componentes reutilizables y estructura modular.
- Se implementará un linter (ejemplo: ESLint para JavaScript/React) y un formatter (ejemplo: Prettier, Checkstyle para Java) para asegurar uniformidad en el código.

c) Pruebas y Validación

- Se realizarán pruebas unitarias en el backend con JUnit para garantizar la funcionalidad de los módulos principales.
- Se ejecutarán pruebas manuales de integración en el frontend para validar la experiencia de usuario y el correcto consumo de la API.

4.6 Gestión de los Recursos Humanos

La gestión de los recursos humanos es fundamental para garantizar la organización y coordinación efectiva del equipo de trabajo. Cada integrante tiene asignadas responsabilidades específicas que permiten cubrir las áreas clave del proyecto: Backend, Frontend, Base de Datos y Documentación/QA. Además, se establecieron canales de comunicación que fomentan la colaboración continua.

a) Asignación de Roles

Integrante	Rol	Responsabilidades
Javier Stanley Valladares Valladares	Líder de Proyecto / Coordinador	Responsable de la integración general, organización del equipo, gestión del cronograma en Notion y comunicación directa con el docente.
Javier Stanley Valladares Valladares	Desarrollador Backend	Encargado de la implementación de la API REST en Java Spring Boot, lógica de negocio y conexión con la base de datos.
Jonathan Josue Cardoza Perez	Desarrollador Frontend / Prototipos	Diseña interfaces de usuario en Figma y en la segunda fase implementará el consumo de la API REST mediante HTML, CSS, JavaScript y React.
Marvin Rene Martinez Gomez	Encargado de Base de Datos	Diseño, normalización y gestión de la base de datos en MySQL, asegurando integridad y consistencia de los datos.
Andrea Lisseth Demata Moreno	Documentación / QA	Redacción de la documentación del proyecto, incluyendo los endpoints de la API, y revisión de la calidad del código y entregables.

b) Colaboración y Comunicación

Para asegurar la colaboración fluida y el trabajo en equipo, se establecerán los siguientes canales:

- Discord: Canal principal para reuniones de voz, discusión diaria y resolución de dudas técnicas.
- Whatsapp: Comunicación rápida y notificaciones sobre avances y tareas urgentes.
- Aula Digital: Entregas oficiales de avances, retroalimentación con el docente y revisión de resultados.

- GitHub: Gestión de código, revisión de *pull requests* y control de versiones.
- Notion: Organización del cronograma, tareas por sprint y documentación centralizada.

La combinación de estos roles y herramientas asegura una colaboración efectiva, fomentando la responsabilidad individual y el trabajo coordinado hacia los objetivos comunes del proyecto.

4.7 Gestión de la Comunicación

La comunicación dentro del equipo y con el docente se gestionará de forma estructurada, asegurando un flujo constante de información que permita dar seguimiento al proyecto, resolver dudas a tiempo y garantizar la coordinación de los integrantes.

a) Informes Semanales de Progreso

- El Coordinador/Líder de Proyecto (Javier Stanley Valladares Valladares) será el encargado de elaborar un informe semanal de progreso, el cual incluirá:
 - Avances logrados durante la semana.
 - Dificultades o bloqueos encontrados.
 - Tareas pendientes para el siguiente sprint.
- Dicho informe será presentado al docente durante las reuniones semanales y quedará registrado en Notion, como evidencia del progreso.
- Además, se realizará una breve reunión interna en Discord cada semana para alinear al equipo y actualizar el cronograma.

b) Canales de Resolución de Dudas

- Correo electrónico institucional: Medio oficial de comunicación con el docente para consultas formales y envío de reportes si es requerido.
- Discord: Canal principal para reuniones rápidas, consultas técnicas y coordinación diaria entre los integrantes.
- Whatsapp: Medio alternativo para comunicación inmediata y notificaciones rápidas de avances o recordatorios de entregas.
- Aula Digital: Espacio oficial para la entrega de documentos y proyectos, así como para recibir retroalimentación del docente.

4.8 Gestión de Adquisición y Cierre

El cierre del proyecto se realizará garantizando que todos los entregables técnicos y documentales estén completos, verificados y disponibles para su consulta. Se utilizarán repositorios y herramientas colaborativas que aseguren la trazabilidad y disponibilidad de la información.

a) Documentación Final

Al finalizar el proyecto, se entregará:

- **Documento Final en PDF:** que incluirá la descripción del proyecto, objetivos, gestión integral, cronograma, costos, roles, reportes y evidencias.
- **README en GitHub:** con información clara y estructurada sobre:
 - Descripción general del proyecto.
 - Tecnologías utilizadas (Java Spring Boot, React, MySQL, Notion, Figma).
 - Instrucciones de instalación y despliegue.
 - Integrantes del equipo y sus roles.
 - Enlaces a Notion para la gestión de tareas y cronograma.

b) Verificación de Entregables

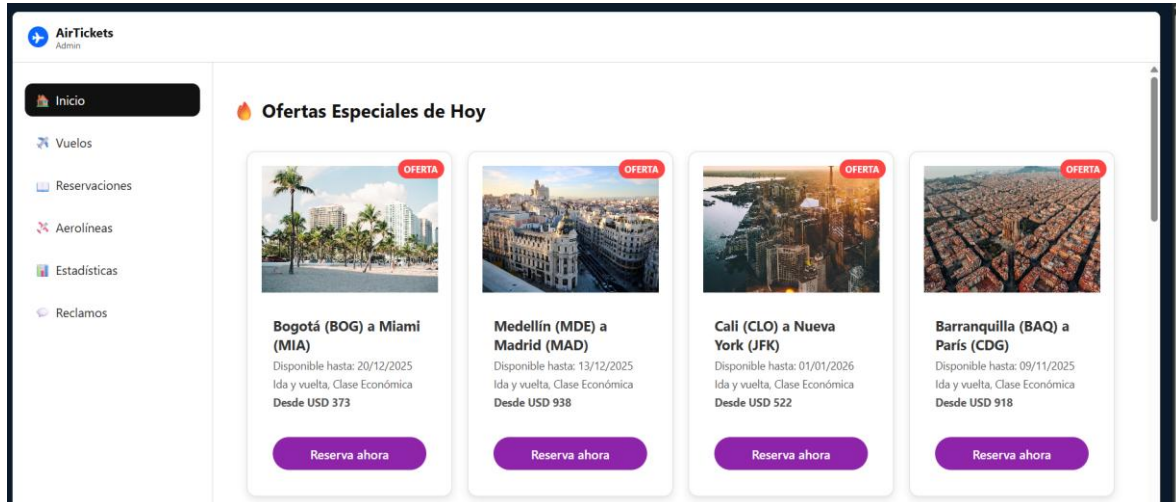
- Antes de la entrega final, se realizará una **revisión grupal** para asegurar que el código, la base de datos y la documentación cumplan con los requisitos planteados.
- Se verificará que el repositorio en **GitHub** esté actualizado y funcional, con todas las ramas integradas en la rama principal (*main*).
- El archivo **PDF final** será almacenado en el repositorio como parte de los entregables oficiales.

c) Cierre del Proyecto

- Una vez entregados todos los elementos, el equipo realizará una reunión final de cierre en **Discord**, donde se revisarán logros, aprendizajes y posibles mejoras futuras.
- El docente tendrá acceso al **GitHub (con el README y el PDF)**, al **Notion (cronograma y evidencias)** y al **Figma (prototipos)** para evaluar la totalidad del proyecto.

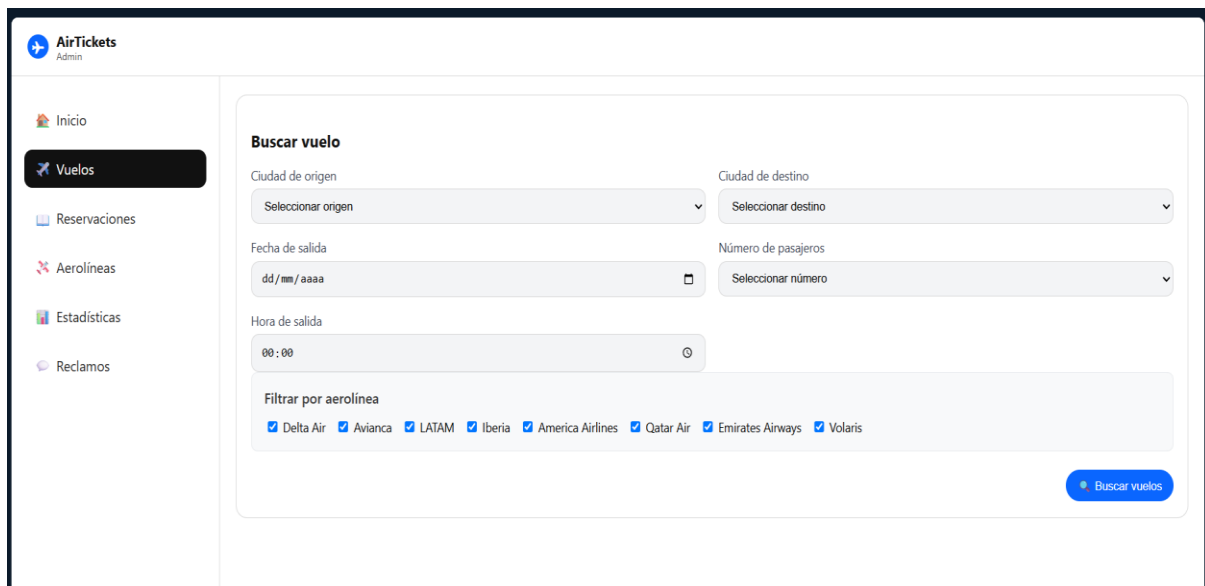
5. Descripción Técnica

5.1. Prototipo de FrontEnd



la vista de inicio

En la página de inicio se mostrarán imágenes de los destinos disponibles a los que se puede viajar con nuestra aerolínea. De esta forma, los usuarios podrán explorar de manera visual las diferentes opciones antes de iniciar el proceso de compra de sus boletos



En esta sección los usuarios podrán realizar la compra de sus vuelos ingresando la ciudad de origen, la ciudad de destino, la fecha de salida, el número de pasajeros y la hora de salida y se filtrarán por la aerolíneas.

AirTickets Admin

Inicio

Vuelos

Reservaciones

Aerolíneas

Estadísticas

Reclamos

Hora de salida

05:00

Filtrar por aerolínea

☒ Delta Air ☒ Avianca ☒ LATAM ☒ Iberia ☒ America Airlines ☒ Qatar Air ☒ Emirates Airways ☒ Volaris

Buscar vuelos

Vuelos Disponibles

Volaris VO-140

\$320.836

22 asientos disponibles

Vuelo directo

22:40 Ciudad de México (MEX) 4h 34m 02:14 Cancún (CUN)

Airbus A320

Seleccionar

Una vez ingresada la información, el sistema mostrará los vuelos disponibles hacia el destino seleccionado, permitiendo al usuario elegir la opción que mejor se adapte a sus necesidades

AirTickets Admin

Inicio

Vuelos

Reservaciones

Aerolíneas

Estadísticas

Reclamos

Avianca Código: AV-101

\$1.850.000

14:30 Bogotá (BOG) 9h 15m 06:45 Madrid (MAD)

Avión: Boeing 787-8 Dreamliner Pasajeros: 1 Vuelo directo

45 asientos disponibles

Pasajero 1

Nombre Completo * Fecha de Nacimiento *

Nombre completo como aparece en el pasaporte dd/mm/aaaa

Número de Pasaporte * Preferencia de Asiento *

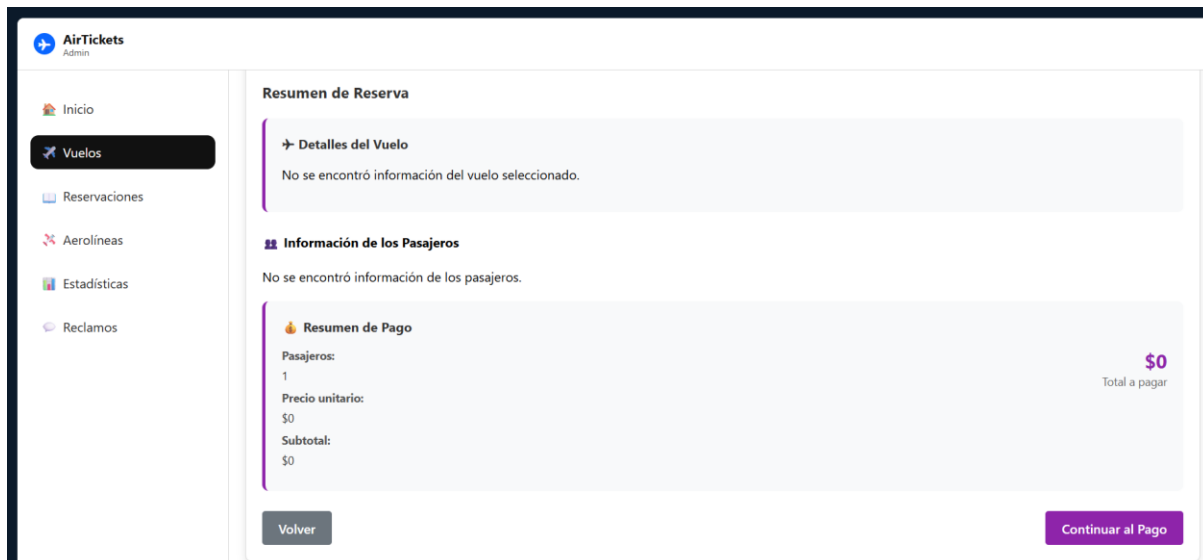
Número de pasaporte Seleccione una opción

Correo Electrónico * Teléfono de Contacto *

correo@ejemplo.com +1234567890

Volver Continuar a Resumen

Después de seleccionar el vuelo, el sistema solicitará algunos datos del pasajero: nombre completo, fecha de nacimiento, número de pasaporte y preferencia de asiento. Una vez completada esta información, podrá continuar con el proceso de compra



AirTickets Admin

Inicio

Vuelos

Reservaciones

Aerolíneas

Estadísticas

Reclamos

Resumen de Reserva

➔ **Detalles del Vuelo**

No se encontró información del vuelo seleccionado.

👤 **Información de los Pasajeros**

No se encontró información de los pasajeros.

🔥 **Resumen de Pago**

Pasajeros: 1

Precio unitario: \$0

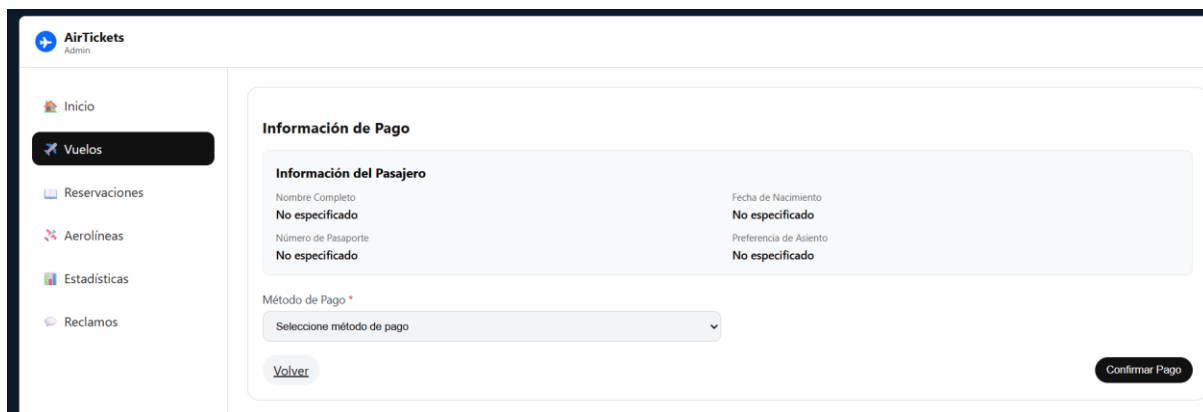
Subtotal: \$0

\$0
Total a pagar

Volver

Continuar al Pago

Después de ingresar la información solicitada, el sistema mostrará un resumen de la reservación, donde el usuario podrá visualizar sus datos personales y verificar que todo esté correcto antes de continuar



AirTickets Admin

Inicio

Vuelos

Reservaciones

Aerolíneas

Estadísticas

Reclamos

Información de Pago

Información del Pasajero

Nombre Completo: No especificado

Fecha de Nacimiento: No especificado

Número de Pasaporte: No especificado

Preferencia de Asiento: No especificado

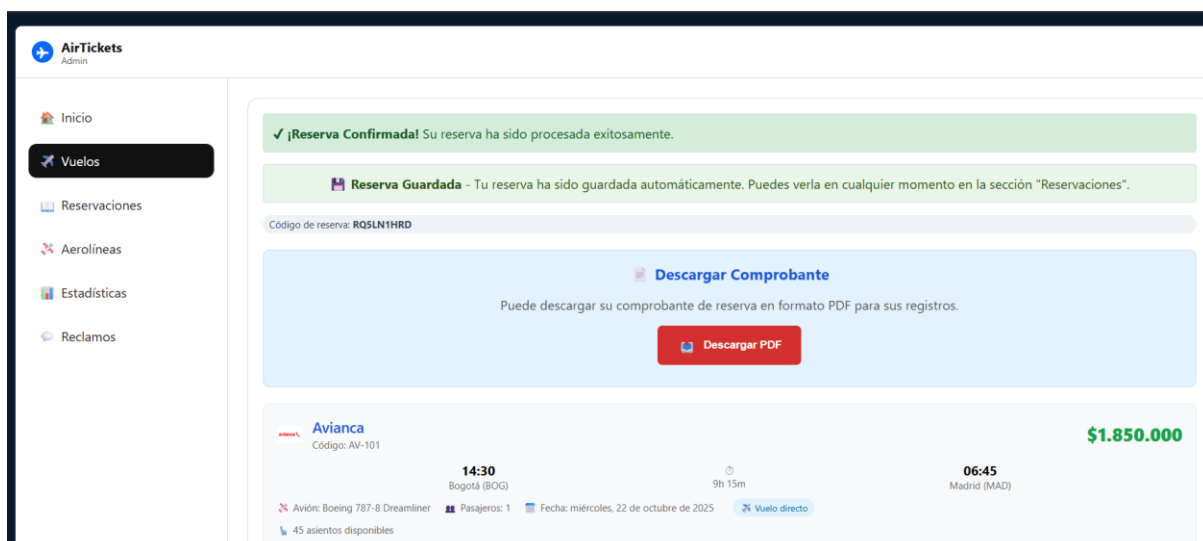
Método de Pago *

Seleccione método de pago

Volver

Confirmar Pago

Finalmente, el sistema mostrará las opciones de pago, donde el usuario podrá seleccionar el método de su preferencia: tarjeta de débito, tarjeta de crédito o transferencia bancaria



AirTickets Admin

Inicio

Vuelos

Reservaciones

Aerolíneas

Estadísticas

Reclamos

✓ **¡Reserva Confirmada!** Su reserva ha sido procesada exitosamente.

📄 **Reserva Guardada** - Tu reserva ha sido guardada automáticamente. Puedes verla en cualquier momento en la sección "Reservaciones".

Código de reserva: RQSLN1HRD

📄 **Descargar Comprobante**

Puede descargar su comprobante de reserva en formato PDF para sus registros.

Descargar PDF

Avianca
Código: AV-101

14:30
Bogotá (BOG)

9h 15m

06:45
Madrid (MAD)

\$1.850.000

Avión: Boeing 787-8 Dreamliner

Pasajeros: 1

Fecha: miércoles, 22 de octubre de 2025

Vuelo directo

45 asientos disponibles

Finalmente, el sistema generará el boleto con la confirmación del pago, dejando listo el vuelo para el pasajero

Mis Reservas

Todos los estados ▼ Todas las aerolíneas ▼

Total reservas: 3 Confirmadas: 2 Check-in: 0 Canceladas: 1

Reserva #RL5QATM9J NUEVA Confirmada

Bogotá BOG 1h 25m **Miami** MIA

02:05 Salida **03:30** Llegada
Iberia • IB-935 • Embraer E190

marvin martinez
Pasaporte: 90785634
Asiento: Ventana
Pago: Tarjeta de Crédito/Débito

\$264.364

Ver detalles Check-in Reclamo Cancelar Descargar Ticket

En esta sección de la página, el usuario podrá visualizar todas sus reservaciones realizadas. Además, tendrá la opción de cancelar alguna de ellas o presentar un reclamo en caso de inconvenientes, como retrasos de vuelo sin notificación o problemas con el equipaje.

AirTickets
Admin

Nuevo Reclamo

Reserva* Vuelo*

Pasajero*

Tipo de Reclamo*

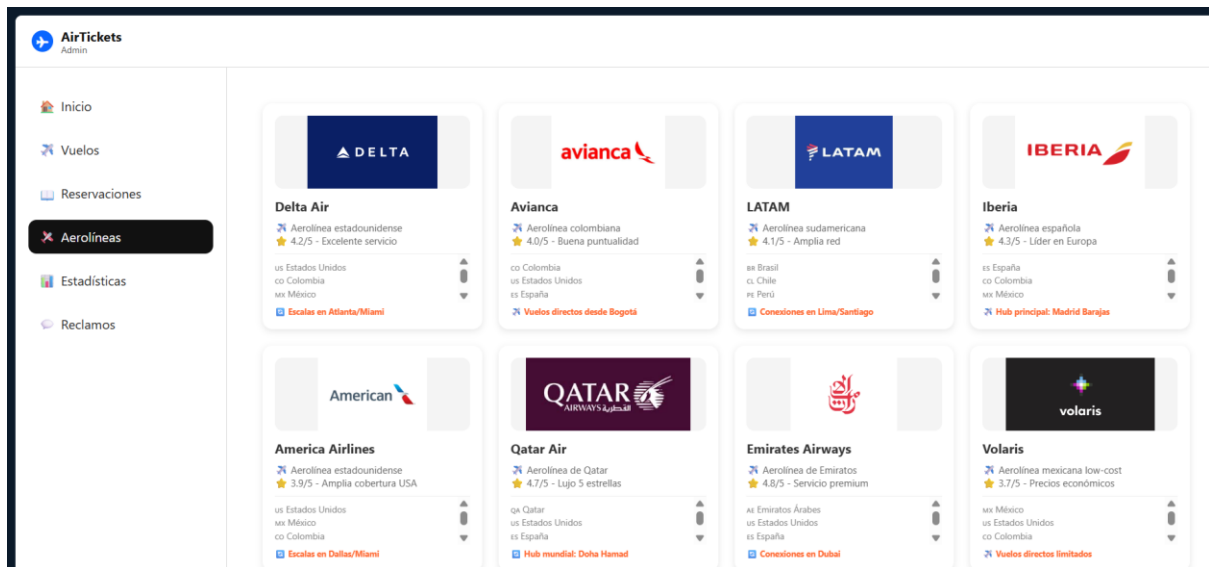
Descripción detallada del reclamo*

Cancelar Enviar Reclamo

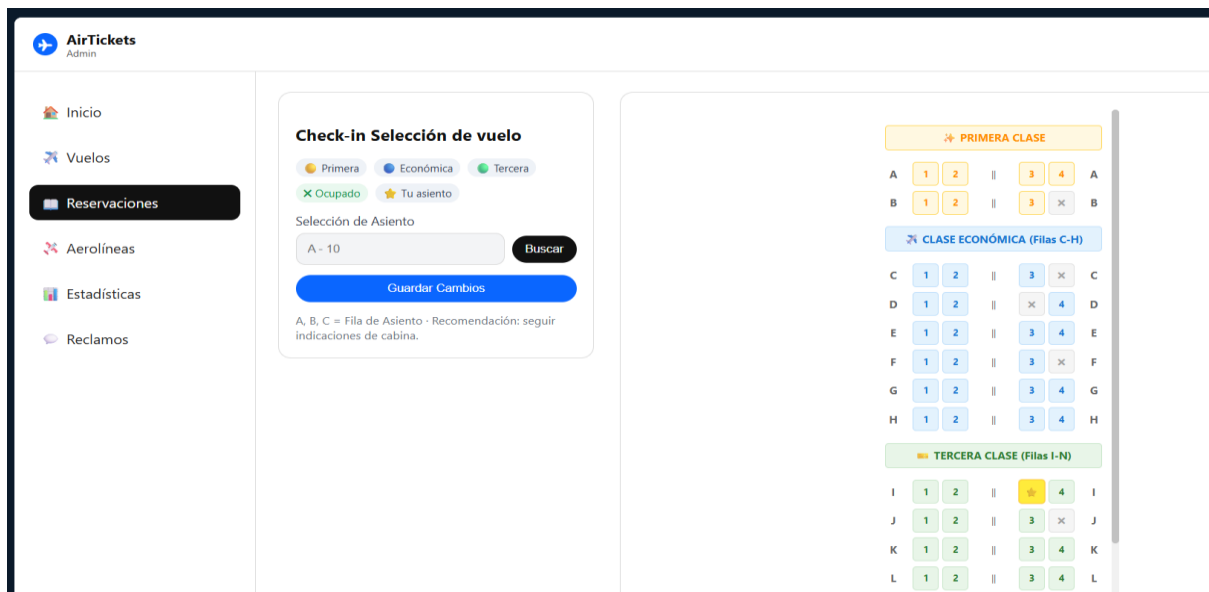
Mis Reclamos

No tiene reclamos registrados.

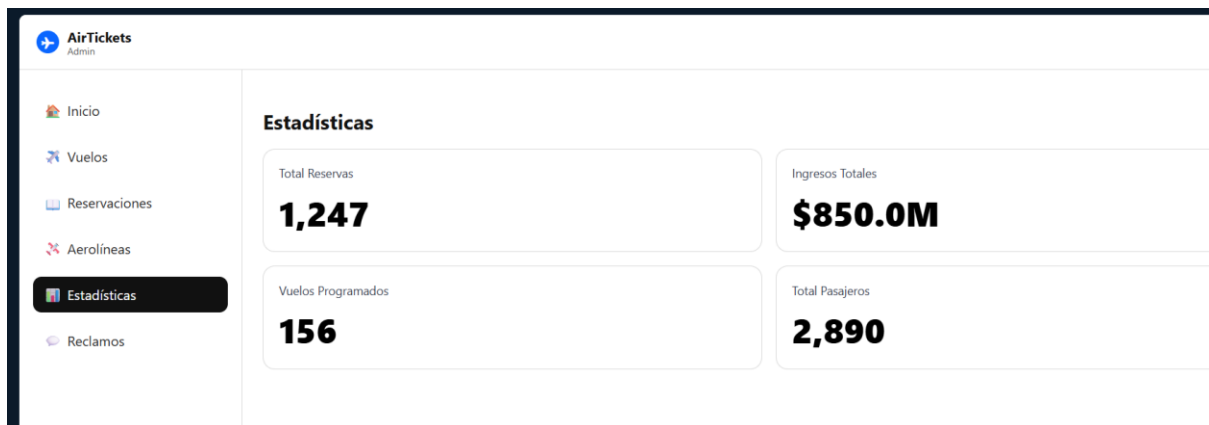
En esta sección, el usuario podrá visualizar el historial de los reclamos realizados, lo que le permitirá dar seguimiento al estado de cada uno



En esta sección, el usuario podrá visualizar todas las aerolíneas disponibles en la aplicación y seleccionar aquella de su preferencia para realizar su reserva



En esta sección, el usuario podrá realizar el check-in y visualizar el asiento asignado para su vuelo



En esta sección, se podrán visualizar todas las estadísticas de la página, incluyendo el total de reservaciones realizadas, los ingresos generados, los vuelos programados y el número total de pasajeros

Figma: Plataforma utilizada para el diseño de prototipos de las interfaces gráficas del sistema, permitiendo la validación temprana del diseño.

<https://www.figma.com/design/QFQtnNAFYdyEt8uNgz0BBq/Sin-t%C3%ADtulo?node-id=105-12&t=KxXe9b7im5IglNFC-0>

5.2 Gestión de la Base de Datos

Para el diseño de la base de datos del sistema de venta y administración de boletos aéreos decidí implementar un modelo relacional en **MySQL Server**. Esta elección se debe a que MySQL Server ofrece robustez, seguridad, soporte para transacciones, alta disponibilidad y un amplio uso en entornos de producción empresarial.

La base de datos fue diseñada bajo principios de **normalización**, buscando reducir la redundancia de datos y asegurar la integridad referencial. Así mismo, se implementaron **índices**, **restricciones** y **triggers** que permiten optimizar consultas, garantizar consistencia y automatizar procesos como la generación de códigos de reservación o números de reclamos.

a) Estructura de la Base de Datos

El modelo contempla las principales entidades y procesos del sistema:

- Países, Ciudades, Aeropuertos: permiten identificar de manera única la geografía y los puntos de operación.
- Aerolíneas, Aviones, Tipos de Avión, Tripulación: gestionan la flota y el personal.
- Rutas, Vuelos, Tarifas: organizan la planificación de los vuelos, sus características y precios.
- Pasajeros: almacenan la información de los clientes.
- Reservaciones, Pagos, Cancelaciones: administran todo el flujo de compra, confirmación, pago y eventual cancelación.
- Reclamos y Tipos de Reclamo: dan soporte a la gestión de incidencias.
- MetodosPago, ClasesServicio: permiten flexibilidad en la compra.
- TripulacionVuelos: administra la asignación de personal a vuelos.

Cada tabla cuenta con claves primarias, claves foráneas, índices y restricciones que aseguran integridad y optimización de consultas.

c) Diccionario de Datos

En este apartado detallamos los campos más importantes de las tablas:

Tabla: Países

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
PaisID	INT	-	Identificador único del país	PK, Not Null
NombrePais	NVARCHAR	100	Nombre oficial del país	Not Null

CodigoPais	NCHAR	3	Código ISO-3166-1 alfa-3	Único
Activo	BIT	-	Indica si el país está disponible	Default=1

Tabla: Ciudades

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
CiudadID	INT	-	Identificador único de la ciudad	PK, Not Null
NombreCiudad	NVARCHAR	100	Nombre oficial de la ciudad	Not Null
CodigoCiudad	NCHAR	3	Código IATA de la ciudad	Único
PaisID	INT	-	País al que pertenece	FK → Países

Tabla: Aeropuertos

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
AeropuertoID	INT	-	Identificador único del aeropuerto	PK, Not Null
NombreAeropuerto	NVARCHAR	200	Nombre oficial del aeropuerto	Not Null
CodigoIATA	NCHAR	3	Código estándar internacional IATA	Único
CodigoCAO	NCHAR	4	Código estándar OACI	Único
CiudadID	INT	-	Ciudad donde está ubicado	FK → Ciudades

Tabla: Aerolíneas

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
AerolineaID	INT	-	Identificador único de aerolínea	PK, Not Null
NombreAerolinea	NVARCHAR	150	Nombre de la aerolínea	Not Null
CodigoIATA	NCHAR	2	Código IATA de la aerolínea	Único
CodigoICAO	NCHAR	3	Código ICAO de la aerolínea	Único
PaisOrigenID	INT	-	País de origen de la aerolínea	FK → Países

Tabla: Vuelos

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
VueloID	INT	-	Identificador único del vuelo	PK, Not Null
NumeroVuelo	NVARCHAR	10	Código de vuelo	Único, Not Null
AerolineaID	INT	-	Aerolínea que opera el vuelo	FK → Aerolíneas
RutaID	INT	-	Ruta asignada	FK → Rutas
AvionID	INT	-	Avión asignado	FK → Aviones
FechaSalida	DATETIME	-	Fecha y hora de salida	Not Null
FechaLlegada	DATETIME	-	Fecha y hora de llegada	Not Null

EstadoVuelo	NVARCHAR	20	Programado, Retrasado, Cancelado...	Default='Programado'
-------------	----------	----	-------------------------------------	----------------------

Tabla: Pasajeros

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
PasajeroID	INT	-	Identificador único del pasajero	PK, Not Null
Nombres	NVARCHAR R	100	Nombres del pasajero	Not Null
Apellidos	NVARCHAR R	100	Apellidos del pasajero	Not Null
NumeroDocumento	NVARCHAR R	20	Documento único (pasaporte, etc.)	Único, Not Null
Nacionalidad	NVARCHAR R	50	País de origen del pasajero	Not Null
Email	NVARCHAR R	100	Correo electrónico	Not Null
Telefono	NVARCHAR R	20	Número de teléfono	Opcional

Tabla: Reservaciones

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
-------	--------------	--------	-------------	---------------

ReservacionID	INT	-	Identificador único de la reservación	PK, Not Null
CodigoReservacion	NVARCHAR	10	Código único de reservación (auto)	Único
PasajeroID	INT	-	Pasajero que realizó la reservación	FK → Pasajeros
VueloID	INT	-	Vuelo reservado	FK → Vuelos
ClaseID	INT	-	Clase de servicio	FK → ClasesServicio
NumeroAsiento	NVARCHAR	5	Asiento asignado	Not Null
EstadoReservacion	NVARCHAR	20	Pendiente, Confirmada, Pagada, Cancelada...	Default='Pendiente'

Tabla: Pagos

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
PagoID	INT	-	Identificador único del pago	PK, Not Null
ReservacionID	INT	-	Reservación asociada	FK → Reservaciones
MetodoPagoID	INT	-	Método de pago utilizado	FK → MetodosPago
MontoPagado	DECIMAL	10,2	Cantidad pagada	Not Null
EstadoPago	NVARCHAR	20	Procesando, Aprobado, Rechazado, etc.	Default='Procesando'

Tabla: Reclamos

Campo	Tipo de Dato	Tamaño	Descripción	Restricciones
Reclamoid	INT	-	Identificador único del reclamo	PK, Not Null
NumeroReclamo	NVARCHAR	15	Número de reclamo (auto generado)	Único
PasajeroID	INT	-	Pasajero que realiza el reclamo	FK → Pasajeros
ReservacionID	INT	-	Reservación asociada (opcional)	FK → Reservaciones
TipoReclamoid	INT	-	Tipo de reclamo	FK → TiposReclamo
Descripcion	NVARCHAR	1000	Detalle del reclamo	Not Null
EstadoReclamo	NVARCHAR	20	Abierto, En Proceso, Resuelto, etc.	Default='Abierto'

d) Características adicionales**Triggers:**

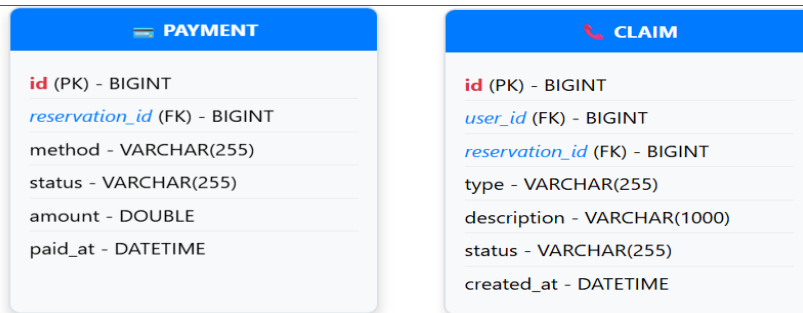
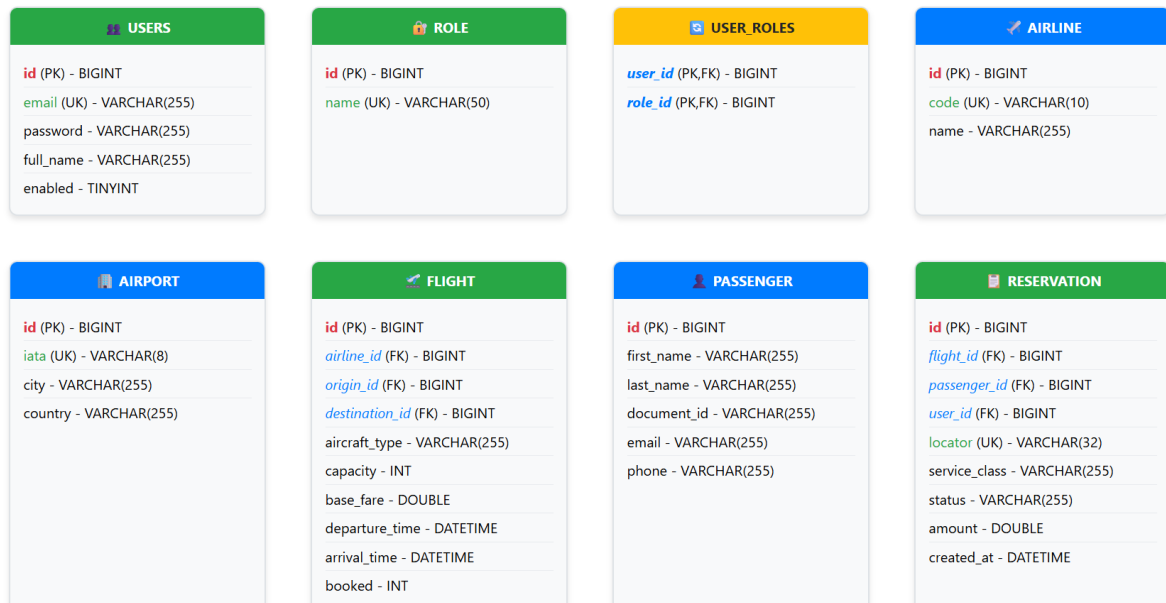
- Actualizan los asientos vendidos de cada tarifa cuando se inserta, actualiza o elimina una reservación.
- Generan automáticamente el código de reservación.
- Generan números de reclamos consecutivos.

Índices:

Se crearon en campos clave como fechas de vuelos, estado de reservaciones, estado de pagos y documentos de pasajeros, optimizando así las consultas más frecuentes.

Con esta estructura, la base de datos garantiza un manejo eficiente de vuelos, pasajeros y transacciones, manteniendo la consistencia y confiabilidad de la información.

Diagrama Entidad Relación



Relaciones y Cardinalidades

USERS ↔ ROLE (Relación Muchos a Muchos)

USERS **1** ————— **N** USER_ROLES **N** ————— **1** ROLE

USERS ↔ RESERVATION (Relación Uno a Muchos)

USERS **1** ————— **N** RESERVATION

RESERVATION ↔ PAYMENT (Relación Uno a Uno)

RESERVATION **1** ————— **1** PAYMENT

El diagrama presentado representa la estructura relacional de la base de datos diseñada en MySQL Server para gestionar los procesos de vuelos, reservaciones y servicios asociados a la aerolínea.

1. Estructura Geográfica y Operativa

[Países] → [Ciudades] → [Aeropuertos]

Define la jerarquía de localización: cada aeropuerto pertenece a una ciudad y cada ciudad a un país.

Esto permite gestionar rutas de vuelo de forma ordenada y mantener la integridad geográfica.

2. Gestión de Aerolíneas y Flota

[Aerolíneas] → [Aviones]

Cada aerolínea opera una flota de aviones.

[Aviones] → [Vuelos] → [Rutas]

Los aviones son asignados a vuelos que cubren rutas específicas (origen y destino).

3. Tripulación de Vuelos

[Vuelos] → [TripulacionVuelos] → [Tripulación]

Se lleva el control del personal asignado a cada vuelo (pilotos, copilotos, azafatas, etc.), lo que asegura trazabilidad y gestión de recursos humanos.

4. Reservaciones y Pasajeros

[Reservaciones] ↔ [Pasajeros]

Cada pasajero puede tener varias reservaciones, y cada reservación está vinculada a un vuelo en particular.

Se registran detalles como asiento, clase de servicio y estado de la reservación.

5. Pagos y Métodos de Pago

[Reservaciones] → [Pagos] → [MetodosPago]

Los pasajeros realizan pagos asociados a sus reservaciones, con flexibilidad para elegir entre diferentes métodos de pago (tarjeta, transferencia, efectivo, etc.).

6. Cancelaciones

[Reservaciones] → [Cancelaciones]

Permite registrar y controlar las cancelaciones de boletos, manteniendo histórico de los cambios y el estado final de la reservación.

7. Reclamos

[Pasajeros] / [Reservaciones] → [Reclamos] → [TiposReclamo]

Los pasajeros pueden generar reclamos vinculados a una reservación o de manera general, clasificándolos por tipo (equipaje perdido, retrasos, mal servicio, etc.).

8. Clases de Servicio

[Reservaciones] → [ClasesServicio]

Identifica si el boleto corresponde a clase económica, ejecutiva o primera clase, lo que influye en precios, beneficios y asignación de asientos.

Aspectos Técnicos

Normalización: se evitan redundancias y se definieron claves primarias y foráneas para garantizar integridad referencial.

Índices: en campos críticos como fechas de vuelo, documentos de pasajeros y estados de pagos/reservaciones, optimizando el rendimiento de consultas.

Triggers: automatizan procesos como la generación de códigos de reservación, números de reclamo y actualización de disponibilidad de asientos.

En conclusión, este modelo refleja de forma integral cómo se gestionan vuelos, pasajeros, reservaciones, pagos y reclamos, asegurando un flujo eficiente y controlado en el sistema de venta y administración de boletos aéreos.

6. Resultados y Conclusiones

a) Resultados Obtenidos

- Durante la primera fase del proyecto, se lograron definir de manera clara el **alcance, objetivos y cronograma** de trabajo, así como la **gestión de roles y comunicación del equipo**.
- Se elaboraron los **prototipos iniciales en Figma**, los cuales sirven como referencia para el diseño de la interfaz de usuario.
- Se configuró el espacio de trabajo en **Notion y GitHub**, lo que permitió establecer una base sólida para la gestión del proyecto y el control de versiones.
- En términos de desempeño, se establecieron los cimientos para un desarrollo ordenado bajo la metodología ágil **Scrum**, garantizando la división del trabajo en sprints y revisiones periódicas.

En la **Fase 2**, se espera obtener resultados más técnicos, como la implementación completa del **backend en Spring Boot**, el **frontend web con React** y la integración con la base de datos **MySQL**, junto con pruebas de funcionamiento.

b) Conclusiones

- El desarrollo de este proyecto ha permitido al equipo **comprender la importancia de la planificación y la asignación de roles**, asegurando que cada integrante tenga responsabilidades claras.
- El uso de herramientas colaborativas como **Notion, GitHub y Discord** ha demostrado ser esencial para mantener una comunicación efectiva y un seguimiento constante del avance.
- Entre los principales desafíos enfrentados, se destaca la **coordinación de tiempos entre los integrantes** y la **adaptación a la metodología ágil**, lo que representó un aprendizaje significativo.
- Como mejora futura, se plantea la posibilidad de **ampliar las funcionalidades** del sistema más allá de lo requerido en el proyecto académico, como integrar reportes estadísticos o un módulo de notificaciones en tiempo real.

7. Anexos y Documentación de Apoyo

En esta sección se incluyen materiales complementarios que respaldan el desarrollo del proyecto y facilitan su comprensión técnica:

- Configuraciones de Herramientas
 - Detalles de la configuración en Notion para la gestión del cronograma y tareas.
 - Lineamientos de trabajo en GitHub para control de versiones y flujos de Pull Requests.
 - Configuración inicial de Figma utilizada para los prototipos y mockups de las interfaces.
- Ejemplos de Código (cuando aplique en Fase 2)
Se agregarán ejemplos de scripts SQL, fragmentos de código en Spring Boot (backend) y React (frontend) que servirán como referencia para la implementación final.
- **Documentación Técnica Adicional**
 - Diagramas entidad-relación de la base de datos.
 - Diccionario de datos con definición de tablas, campos y relaciones.
 - Capturas de pantallas del cronograma en Notion y prototipos en Figma.
- **Configuraciones Avanzadas:**
 - Lineamientos iniciales de instalación y configuración del entorno de desarrollo (Java, Spring Boot, React, MySQL, IntelliJ IDEA).
 - Conexión entre el backend en Spring Boot y la base de datos MySQL.