

< **SENA** /Soft >

pragma

Reto Categoría
Desarrollo Libre

SENASoft

Versión 2025

Reto: Sistema para Compra de Tiquetes Aéreos

Descripción del reto

El equipo deberá desarrollar una aplicación/web/software que permita la compra de tiquetes de avión, gestionando vuelos, asientos, pasajeros y pagos simulados.

El sistema debe garantizar que ningún asiento se venda dos veces y que la capacidad de cada vuelo no se exceda.

La capacidad del vuelo debe estar determinada por el **modelo de avión asignado**.

El sistema debe incluir búsqueda de vuelos, control de asientos, simulación de pagos, confirmación de reservas y generación de tiquetes electrónicos.

Este reto busca evaluar la capacidad de los participantes para diseñar, construir y desplegar una solución completa bajo buenas prácticas de arquitectura, control de versiones y experiencia de usuario.

Objetivo general

Diseñar e implementar una aplicación/sistema que permita buscar vuelos, registrar pasajeros, seleccionar asientos y realizar un pago simulado, asegurando la integridad de los datos, la usabilidad y el control de concurrencia.

Objetivos específicos

- Implementar un buscador de vuelos con opción de solo ida o ida y regreso.
- Restringir fechas válidas desde el día actual hasta dos meses hacia adelante.
- Autocompletar las ciudades de origen y destino.
- Registrar hasta 5 pasajeros por compra.
- Controlar la concurrencia y la venta de asientos según la capacidad del modelo de avión.

- Simular el proceso de pago de manera controlada y segura.
- Generar tiquetes electrónicos con código de reserva único.
- Permitir la confirmación y visualización de la reserva.
- Mostrar mensajes claros de éxito y error.
- Mantener un control de versiones con GitFlow y commits de ambos integrantes del equipo.
- Desplegar la solución en Internet (opcional, pero con puntaje adicional).

Requisitos funcionales

1. Búsqueda de vuelos

- Selección de tipo de viaje: solo ida o ida y regreso.
- Fechas válidas desde el día actual hasta 2 meses adelante.
- Ciudades de origen y destino con autocompletado.

2. Selección de vuelo

- Mostrar vuelos disponibles con origen, destino, fecha, hora, capacidad y precio.
- Controlar la capacidad según el modelo de avión asignado.
- Bloquear reserva si no hay asientos disponibles.

3. Datos de pasajeros

- Máximo 5 pasajeros por compra.
- Datos requeridos: primer apellido, segundo apellido, nombres, fecha de nacimiento, género, tipo y número de documento, condición de infante (menor de 3 años), celular y correo electrónico.

4. Datos del pagador

- Nombre completo, tipo y número de documento, correo y teléfono.

5. **Aceptación de términos y condiciones**

- Debe ser obligatorio antes de procesar el pago.

6. **Simulación de pago**

- Opciones: Tarjeta de crédito, débito o PSE.
- Simulación visual del pago exitoso o rechazado.

7. **Generación de tickets**

- Visualización y descarga del ticket en PDF o JSON.
- Mostrar código único de reserva, pasajeros, vuelo y asientos.

8. **Confirmación de reserva**

- Mostrar resumen completo con código, pasajeros, vuelo y valor total.

Requisitos técnicos

- Evitar venta duplicada de asientos.
- No permitir superar la capacidad del vuelo.
- Validar datos de entrada y salida.
- Arquitectura y buenas prácticas.
- Control de versiones con GitFlow.
- Despliegue opcional (otorga puntos adicionales).

Mínimo a desarrollar

1. Sección de búsqueda de vuelos.
2. Sección de selección de asientos.
3. Sección de registro de pasajeros.
4. Sección de simulación de pago.
5. Sección de generación y visualización de tiquetes.
6. Módulo administrativo para ver vuelos y reservas (opcional).
7. Registro o autenticación (opcional).

Competencias relacionadas y jornadas

Día 1 – Análisis y diseño

Actividades:

- Análisis de requerimientos.
- Diseño de base de datos.
- Creación de todos los mockups, prototipo borrador, UX/UI.
- Diagrama de arquitectura y estructura del proyecto (tecnologías a usar).
- Creación del repositorio (GitHub o GitLab) usando GitFlow.

Día 2 – Desarrollo inicial

Actividades:

- Codificación diseño UX/UI.
- Desarrollo del backend o lógica principal.
- Registro de commits de ambos integrantes (obligatorio).

Día 3 – Implementación y pruebas

Actividades:

- Avances y revisión del código.
- Desarrollo de funcionalidades faltantes.
- Despliegue de la solución (opcional).
- Presentación final ante el jurado.

Rúbrica de evaluación (escala 0 a 20) x 5 items = 100 puntos posibles

Criterio	0	1-5	6-10	11-20
Funcionalidad completa	No cumple ningún requisito	Cumple parcialmente	Cumple mayoría	Cumple totalmente
Control de concurrencia	No implementa control	Parcialmente funcional	Control aceptable	Control completo y robusto
Diseño UX/UI	Deficiente o inexistente	Básico	Aceptable	Profesional e intuitivo
Buenas prácticas y GitFlow	Sin control de versiones	Commits de un solo miembro	GitFlow aplicado parcialmente	GitFlow completo con commits de ambos
Despliegue y documentación	No hay evidencia	Evidencia incompleta	Documentación parcial	Despliegue funcional y documentación clara

Entrega final

El equipo deberá realizar Demostración funcional ante el jurado.

- Si hay empate, se dará un punto adicional por entrega temprana.