
OBJETIVO GENERAL:

Diseñar e implementar una aplicación de Gestión de Boletos Aéreos que permita repasar todos los conceptos aprendidos en cursos anteriores.

OBJETIVOS ESPECIFICOS:

- Repasar los fundamentos de Java (Tipos de datos, eventos y operadores)
- Manejar correctamente la herencia, paquetes, interfaces y las bibliotecas.
- Implementar haciendo uso correcto de Capas y Patrones de Diseño.
- Implementar haciendo uso de Manejo de Colecciones, paquetes y bibliotecas.

DESCRIPCION:

El proyecto permitirá la gestión de tiquetes aéreos, por lo que el usuario y funcionarios de la empresa podrán administrar toda la información relacionada a este proceso.



El proyecto consta de los siguientes módulos:

- **Sección pública**

En la sección pública del sitio web, los usuarios podrán explorar las diferentes páginas informativas relacionada con la empresa, como el referentes institucional, historia de la empresa y contáctenos, así como un buscador de vuelos, en esta sección del sitio web los usuarios podrán realizar búsqueda de los diferentes vuelos que ofrece la empresa:

- Solo ida o ida y regreso
- Origen
- Destino
- Fecha ida
- Fecha regreso (opcional si es solo de ida)
- Cantidad pasajeros

En la página principal deberá aparecer un Banner Slider con la información de las rutas que tienen descuento (esta información deberá consultarse desde la base de datos).

Notas

- El diseño del sitio web deberá ser responsivo¹.
- Para la consulta de de vuelos, si el resultado de la consulta es mayor a 10, deberá existir paginación del resultado.

1. Opciones de usuario

Además de la sección pública antes mencionada, el usuario tendrá acceso a la gestión de su información para poder hacer uso del servicio, por lo cual el usuario tendrá la opción para registrarse o “loguearse” en el sitio web, en una opción de menú llamada “Acceder a mi cuenta” y tendrá dos opciones:

1. "Logueo": El usuario deberá indicar su nombre de usuario y contraseña, podrá realizarlo en cualquier momento o antes de llegar a paso de la opción de pago, el sistema deberá validar la información suministrada por el usuario, de ser correcto, el usuario será almacenado en sesión (el sitio web siempre deberá mostrar el nombre del usuario logueado en ese momento), posterior a esto, el sistema permitirá:
 - Que un usuario logueado pueda buscar reservaciones realizadas con anterioridad.
1. Búsqueda y consulta de vuelos: en la búsqueda de vuelos el usuario podrá seleccionar un vuelo, en las diferentes rutas y horarios disponibles.

¹El diseño web adaptable, adaptativo o responsivo, es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla

- Al finalizar la compra, el sitio web deberá implementar un medio de pago electrónico el cual para realizar el pago correspondiente, en caso contrario el usuario podrá cancelar el pedido.

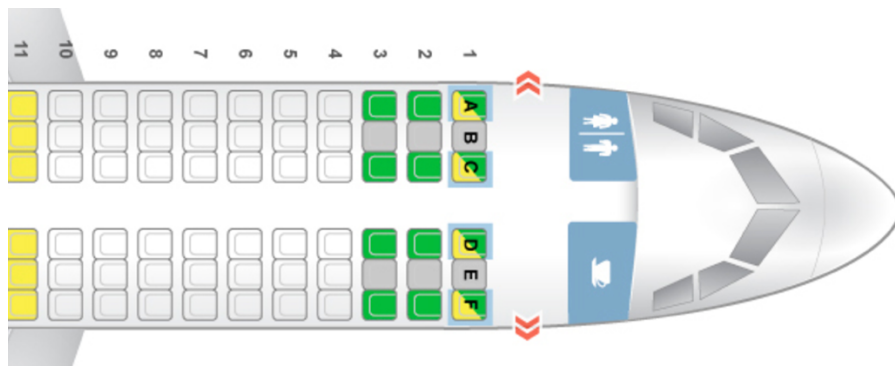
Notas:

La compra o reserva solo debe validar la cantidad de asientos disponibles en el vuelo, la selección de asientos se realizará en el check-in.

- Si el usuario no posee cuenta, este podrá registrarse en el sitio web, suministrando la siguiente información:
 - Usuario
 - Contraseña
 - Nombre
 - Apellidos
 - Correo electrónico
 - Fecha Nacimiento
 - Dirección (la dirección deberá ser mediante en señalamiento de un punto en un mapa utilizando el API de google Maps)
 - Teléfono de trabajo
 - Celular

Cuando el usuario se registra, podrá tener acceso a las opciones de usuario.

- Check-in: la única opción disponible en el check-in será la selección de asientos en el vuelo y estará disponible 24 horas antes de su partida. El usuario debe observar claramente cuales asientos estan ocupados y cuales estan reservados.



- Otras opciones con las que contrará el usuario son:
 - Ver histórico de compras con su respectivo detalle
 - Administrar su información de usuario (modificar la información de registro)

2. Opciones de administración

El sistema permitirá la gestión de la información relacionada a la compra y reserva de tiquetes aéreos, como lo son los siguiente módulos.

- **Gestión de tipos de aviones**
 - A este módulo solo van a tener acceso los administradores de la aplicación, y permitirá a estos usuarios identificar los diferentes tipos de aviones que tendrá la empresa. La información relevante de los tipo de aviones es: identificador, año, modelo, marca, cantidad de pasajeros, cantidad de filas y cantidad de asientos por fila (mínimo 6, es decir, un pasillo y máximo 9, es decir, dos pasillos).
- **Gestión de rutas**
 - Solo tendrá acceso los administradores, permitira gestionar la información relacionada con las rutas, identificador, ruta y duración (horas y minutos).
- **Gestión de horarios**
 - Para cada ruta que exista en la flota de la empresa se deberá crear un horario (día de la semana, hora y minutos), su hora de llegada (esto deberá ser calculado según la duración del vuelo) así como su respectivo precio.
- **Gestión de aviones**
 - Es importante para la empresa tener la información de la flota, por lo que deberá permitir agregar nuevo aviones a la flota, en donde deberá almacenar un identificador del avion (puede ser un número o nombre), ruta (asociado a la gestión de rutas), horario y el tipo de avión (también asociado el modulo anterior de tipos de aviones).

3. Reportes de la aplicación Web

- a. El sistema deberá tener una serie de reportes que den apoyo a la toma de decisiones relacionadas a la administración de la empresa:
 - 3.a.i. Gráfico de cantidad facturado por mes (los últimos 12 meses)
 - 3.a.ii. Gráfico de cantidad de Ingresos por año (el año actual)
 - 3.a.iii. Listado de clientes por avión.

3.a.iv. Las 5 rutas con más ventas o reservaciones.

Notas:

- El proyecto deberá ser desarrollado con la arquitectura vista en clases (MVC²), en caso de que se incumpla con este punto el proyecto no será evaluado.
- Todos los precios de los vuelos deberá ser en dolares, si el usuario desea ver los precios en colones el sistema deberá permitir mostrar los montos en esta moneda, para lo que deberá consultar el tipo de cambio del BCCR.
- El día de defensa del proyecto se deberán realizar una prueba con varias reservas y pruebas de los diferentes módulos.

INSTRUCCIONES GENERALES:

- No se aceptarán trabajos después de la fecha indicada, excepto por razones debidamente justificadas (comprobante).
- Se habilitará en el aula virtual las opciones para que los estudiantes puedan entregar el laboratorio respectivo. Si por alguna razón el profesor pasa la fecha de la entrega de alguna evaluación, lo notificará por medio de la plataforma.
- El laboratorio debe mostrar la eficiencia de un trabajo hecho por un futuro profesional. Por lo tanto se esperan implementaciones bien hechas y que demuestren dominio de los temas solicitados.
- Los trabajos deben ser diferentes, en caso de plagio del trabajo o descarga de internet sin la debida justificación, los estudiantes serán sancionados según el reglamento establecido para tales efectos por la universidad.
- En caso de que el (la) o los (las) estudiantes no puedan demostrar que realizaron el laboratorio, pierden el laboratorio y el mismo será anulado.
- El laboratorio debe estar funcional, de lo contrario no será evaluado. Por lo tanto en su defecto se procederá a poner una nota de CERO.

INSTRUCCIONES ESPECIFICAS:

²Modelo vista-controlador arquitectura a 5 capas.

- El laboratorio se deberá realizar en grupos de estudiantes, según la distribución del profesor.
- A la hora de la entrega del proyecto, el (la) o los (las) estudiante(s) deberán subir un solo trabajo, el cual debe tener los números de las cédulas respectivos. Por ejemplo: Ced-1...Ced-N.zip.
- No se aceptarán laboratorios enviados fuera de la plataforma y fuera de la fecha de entrega.
- Deben realizar un solo back-end y cuatro front-end:
 - Desktop
 - Web
 - Móvil (será visto en el transcurso del curso, utilizando en el front-end kotlin).
- Cualquier otro elemento que el profesor considere pertinente.

EVALUACION:

Rubro por evaluar	Total porcentaje	Porcentaje obtenido
BackEnd <ul style="list-style-type: none"> • Base de Datos (funciones y procedimientos/PL-SQL). (20%). • Arquitectura por Capas (Modelo a 5 capas, utilizar MVC). (15%). • Patrones de diseño (Observer, Sigleton y Iterator). (15%). 	40%	
FrontEnd <ul style="list-style-type: none"> • Navegabilidad del sistema (20%). • Manejo adecuado de la Interfaz gráfica (15%) • Uso correcto de cookies, sessions, u otros recursos para la administración de las interfaces gráficas. (15%). 	60% - 30% c/u	
PORCENTAJE TOTAL OBTENIDO	100%	