



Desafío - Servicio Control de Destinos

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta y sube el `.zip`.

Descripción

Crear una aplicación que:

- Permita el ingreso de pasajeros con sus respectivos datos además de listarlos.
- Permita ingresar destinos (ciudades) y listarlas.
- Al momento de crear un destino, se debe poder asignar a un usuario con el objetivo de identificar destinos por usuarios.

Requerimientos

1. Usar Spring Initializr para generar proyecto con las respectivas dependencias.
2. Crear modelo `Pasajero` con los siguientes atributos, en carpeta `model` la cual debe contener anotaciones de JPA y tener una relación de 1:N con `Destino`.
 - id
 - rut
 - nombre
 - apellido
 - edad
 - ciudadNatal
3. Crear modelo `Destino` con los siguientes atributos, en carpeta `model` la cual debe contener anotaciones de JPA y tener una relación de N:1 con `Pasajero`.
 - id
 - ciudad
 - pais
 - fecha
4. Crear interfaz llamada `DestinoRepository` en carpeta `repository`.
5. Crear interfaz llamada `DestinoService` en carpeta `service`.
 - Método `save` para crear destino.
 - Método `findAll` para listar destinos.

6. Crear clase llamada `DestinoServiceImpl` en carpeta `service`.
 - Método `save` para crear destino.
 - Método `findAll` para listar destinos.
7. Crear interfaz llamada `PasajeroRepository` en carpeta `repository`.
8. Crear interfaz llamada `PasajeroService` en carpeta `service`.
 - Método `save` para crear pasajero.
 - Método `findAll` para listar pasajeros.
9. Crear clase llamada `PasajeroServiceImpl` en carpeta `service`.
 - Método `save` para crear pasajeros.
 - Método `findAll` para listar pasajeros.
10. Crear clase `DestinoController`.
 - Método `crearDestino` (GET) mostrar formulario de destino.
 - Método `crearDestino` (POST) para crear destino.
 - Método `destinos` para mostrar vista de destinos.
11. Crear vista `crear-destino.html` en carpeta `resources/templates`, para agregar destino a pasajero.
12. Crear vista `destinos.html` en carpeta `resources/templates`, para listar destinos registrados.
13. Crear clase `PasajeroController`.
 - Método `crearPasajero` (GET) mostrar formulario de pasajero.
 - Método `crearPasajero` (POST) para crear pasajero.
 - Método `pasajeros` para mostrar vista de pasajeros.
14. Crear vista `crear-pasajero.html` en carpeta `resources/templates`, para agregar pasajero.
15. Crear vista `pasajeros.html` en carpeta `resources/templates`, para listar pasajeros registrados.

Consideraciones

El formulario de ingreso de pasajero, debe lucir así:

PasajerosDestinosNuevo PasajeroNuevo Destino

Ingreso Pasajero

Rut

Nombre

Apellido

Edad

Ciudad Natal

Agregar

El formulario de ingreso de destino, debe lucir así:

PasajerosDestinosNuevo PasajeroNuevo Destino

Ingreso Destino

Ciudad

País

Fecha

mm / dd / yyyy

Pasajero

✓ Martin

Richard

Agregar

El sitio que muestra la lista de destinos ingresado debe lucir así:

PasajerosDestinosNuevo PasajeroNuevo Destino

Destinos			
Ciudad	País	Fecha	Pasajero
Laussane	Suiza	2019-11-25	Martin

El sitio que muestra la lista de pasajeros ingresados debe lucir así:

PasajerosDestinosNuevo PasajeroNuevo Destino

Pasajeros					
Rut	Nombre	Apellido	Edad	Ciudad Natal	Destinos
1-9	Martin	Odersky	61	IdN	Laussane
2-6	Richard	Hickey	56	IdNN	