

Problemática

La aerolínea NORTE-SUR (NR) desea que sus clientes puedan reservar vuelos privados en su flota de aviones, para lo cual le han solicitado a su equipo de trabajo que desarrollen una aplicación robusta y que pueda ser utilizada en diferentes dispositivos.

En primera instancia se desarrollará el backend bajo los siguientes requisitos:

- Desarrollar una API que permita hacer un CRUD sobre los datos de catálogo de vuelos
- Desarrollar una API que permita hacer un CRUD sobre la reserva de vuelos
- Desarrollar una API para administrar clientes (usuarios) de la aplicación

Requerimiento no funcional se debe tener en cuenta:

- Utilizar el lenguaje de programación java y el framework Spring
- Utilizar el motor de base de datos relacional PostgreSQL 11 o superior
- Para la seguridad de las API se debe utilizar el estándar RFC 7519 (JWT)

Requerimientos de arquitectura:

- Implementar el modelo de N capas utilizado en clases
 - Capa de persistencia (Repository y Entity)
 - Capa de lógica de negocios (Service)
 - Capa de controlador (Controller y DTO)
 - Capa de manejo de excepciones centralizadas (@ControllerAdvice)
 - Capa de seguridad (JWT)

Modelo de datos

Se propone utilizar el siguiente modelo de datos, este modelo puede ser mejorado y varios aspectos, por lo tanto es libre de modificarlo:

Flight

id: int
departureDate: String
departureAirportCode: String
departureAirportName: String
departureCity: String
departureLocale: String
arrivalDate: String
arrivalAirportCode: String
arrivalAirportName: String
arrivalCity: String
arrivalLocale: String
ticketPrice: Int
ticketCurrency: String
flightNumber: Int
seatCapacity: Int

Booking

id: int
status: BookingStatus
outboundFlight: Flight
paymentToken: String
checkedIn: Boolean
customer: User
createdAt: String
bookingReference: String

BookingStatus (enum)

UNCONFIRMED
CONFIRMED
CANCELLED

User

id: int
fullname: String
username: String (email)
password: String

Endpoints para la API Rest Catalog

Endpoint	HTTP Method	Query Parameter	Path Parameter	Request payload	Security	Description
/catalog/	GET	departureAirportCode arrivalAirportCode departureDate Ejemplo: /catalog/?departureAirportCode=LON&arrivalAirportCode=ARG&departureDate=2022-04-29				Obtiene la lista de todos los vuelos que cumplan con los criterios de búsqueda establecidos en el query parameter
/catalog/	POST			Flight JSON Object	SI Header Param: Authorization	Permite crear un nuevo vuelo
/catalog/:id	PUT		id	Flight	SI	Actualiza un

			ejemplo: /catalog/1	JSON Object	Header Param: Authorization	vuelo del catálogo por el id
/catalog/:id	DELETE		id ejemplo: /catalog/1			Elimina un vuelo
/catalog/:airportCode	GET	departureDate ejemplo: /catalog/MAD/?departureDate=2022-05-12	airportCode			Obtiene un listado de los vuelos que salen de un aeropuerto en una fecha determinada

Endpoints para la API Rest Booking

Endpoint	HTTP Method	Query Parameter	Path Parameter	Request payload	Security	Description
/booking/:id	GET		id			Obtiene una reserva por el id
/booking/	GET	status customerName Ejemplo: /booking/ /booking/?status=CONFIRMED /booking/?customerName=juan /booking/?status=UNCONFIRMED&customerName=miguel				Obtiene un listado de reserva según el parámetro de búsqueda indicado, puede ser sin parámetro, un sólo parámetro o los dos parámetros a la vez
/booking/flight/:idflight/user/:iduser	POST		idflight iduser Ejemplo: /booking/flight/1/user/5	Booking JSON Object	SI Header Param: Authorization	Permite crear una reserva de un vuelo y un usuario dado en el path parámetro
/booking/:id	DELETE		id		SI	Elimina una reserva por el id
/booking/flight/:idflight	GET		idflight			Obtiene todas las reservas

						para un vuelo especificado
--	--	--	--	--	--	----------------------------

Endpoints para la API Rest User

Endpoint	HTTP Method	Query Parameter	Path Parameter	Request payload	Security	Description
/users/:id	GET		id Ejemplo: /users/5			Obtiene un usuario por el id
/users/	GET					Listar todos los usuarios
/users/	POST			User JSON Object		Crear usuario
/users/:id	PUT		id Ejemplo: /users/5	User JSON Object	SI Header Param: Authorization	Actualizar usuario
/users/:id	DELETE		id Ejemplo: /users/5		SI Header Param: Authorization	Eliminar usuario

Endpoints para la API Rest Auth

Endpoint	HTTP Method	form-data	Request payload	Description
/login/	GET	username password		permite la validación de un usuario registrado y retorna un token

Entregables

- URL del repositorio **github** donde se encuentre el código fuente