

# Proyecto agencia de viajes

Se pretende desarrollar una aplicación para reserva de viajes. Las reservas de viajes las pueden realizar usuarios autenticados, que podrán registrarse a través de la propia aplicación.

También podrán consultar sus reservas realizadas.

## Descripción del funcionamiento

Inicialmente, la aplicación mostrará este aspecto:

Acceso	Reservar	Mis reservas
(logo o imagen de la agencia)		

Unicamente estará habilitado el botón de "Acceso". Al pulsarlo, se mostrará la siguiente página:

<b>Acceso</b>	Reservar	Mis reservas
---------------	----------	--------------

## Autenticación

Usuario:

Password:

Entrar

[Regístrese](#)

El usuario se tendrá que autenticar. Si no está registrado, podrá registrarse:

<b>Acceso</b>	Reservar	Mis reservas
---------------	----------	--------------

## Registro

Usuario:

Password:

Dirección:

DNI:

Tarjeta:

Registrar

Tras registrarse, se mostrará de nuevo la página de acceso. Al introducir sus credenciales se habilitarán los otros dos enlaces y se le llevará a la página de reservas, pero si no son válidos los credenciales, se mostrará en la misma página un mensaje de error

El aspecto de la página de reservas será:

Acceso	Reservar	Mis reservas
<h3>Datos de reserva</h3>		
Destino:	<input type="text"/>	↓
Hotel:	<input type="text"/>	↓
Personas:	<input type="text" value="1 - 4"/>	↓
Vuelo:	<input type="text"/>	↓
<div>Reservar</div>		

Primeramente, elegirá un destino entre los existentes. Al hacerlo, una lista de hoteles disponibles en ese destino. Después, seleccionará el número de plazas (mínimo 1 máximo 4). Tras ello, la lista de vuelos disponibles para ese destino y que tengan al menos el número de plazas libres seleccionadas. Al pulsar el botón reservar la reserva quedará grabada.

Al realizar una reserva, se tendrá en cuenta que no solo se registra la reserva en la tabla correspondiente, también se actualizarán las plazas libres del vuelo

La opción "Mis reservas" mostrará una página con las reservas realizadas por ese usuario:

Acceso	Reservar	Mis reservas
--------	----------	--------------

Reservas de ---

Destino	Hotel	Fecha	Total

Desconectar

Al desconectar se nos lleva de nuevo a la página principal.

## Descripción técnica

El backend se implementará mediante microservicios Spring Boot. Y se crearán, al menos, tres microservicios:

### Microservicio de hoteles.

Expone al menos los siguientes recursos, todos ellos de acceso libre:

- Datos de hotel por identificador
- Lista de hoteles por localización

### Microservicio de vuelos

Expone al menos los siguientes recursos:

- Lista de vuelos a partir de un destino y número de plazas
- Actualización de plazas a partir de identificador de vuelo y plazas reservadas (securizado mediante token JWT para que solo usuarios determinados puedan utilizarlo)
- Datos de Vuelo por identificador

### Microservicio de reservas

Expone al menos los siguientes recursos:

- Alta de la misma con los datos de reserva. securizado mediante token JWT para que solo usuarios determinados puedan utilizarlo).
- Lista de reservas por cliente

Microservicio de cliente

Expondrá al menos los siguientes recursos:

- Devolución de un Cliente a partir de la combinación usuario y contraseña
- Registro de un Cliente a partir de los datos del mismo
- Devolución de un Cliente a partir del usuario

## Cliente front

El front se desarrollará enteramente en Angular.

## Base de datos

Base de datos en MySQL (se proporciona)

## Aplicación Administrador

Por otro lado, se realizará una pequeña aplicación Web clásica para un hipotético usuario administrador de hoteles , consistente en una página que le mostrará todos los hoteles y podrá habilitar/deshabilitar los que necesite

## Consideraciones:

- Todos los microservicios serán registrados en Eureka y la interacción entre los mismos se realizará mediante descubrimiento
- El cliente front accederá a los microservicios a través de un gateway y estará desplegado en un servidor Tomcat
- La configuración de microservicios y eureka estará centralizada en un servidor de configuración
- Los microservicios serán desplegados en contenedores docker