

# Ejercicio 05 - Autenticación y Autorización

---

Este ejercicio hace referencia a los siguientes ejemplos:

- `o8_express_autenticacion_json_web_token` ([https://github.com/albertsgrc/curso-mean-jedi/tree/master/ejemplos/08\\_express\\_autenticacion\\_json\\_web\\_token](https://github.com/albertsgrc/curso-mean-jedi/tree/master/ejemplos/08_express_autenticacion_json_web_token))

Imaginad que tenemos naves espaciales que quieren aterrizar en un puerto espacial. El planeta en el que queremos aterrizar tiene 3 tipos diferentes de puertos espaciales, cada uno con sus correspondientes coordenadas.

El puerto espacial al que nos redireccionen dependerá del peso de nuestra nave, que podrá ser 'pequeño', 'medio' o 'grande'.

La nave espacial tendrá un identificador y un código de seguridad, que lo autenticarán a la hora de aterrizar en un puerto espacial.

De este modo, el modelo de la nave espacial sería:

```
{
  identificador: String,
  codigo_seguridad: String,
  peso: String
}
```

Y el modelo del puerto espacial sería:

```
{
  peso_aceptado: String,
  coordenada_x: Number,
  coordenada_y: Number
}
```

Tendremos 2 rutas:

## **/authenticate**

Acepta una petición de tipo POST con un identificador y un código de seguridad pasados por el body, y devuelve un token que garantiza acceso al sistema de puertos espaciales.

## **/aterrizaje**

Acepta sólo una petición de tipo GET, y devuelve las coordenadas del puerto espacial que le corresponde a la nave que quiere aterrizar, dependiendo del peso de la nave y el peso que acepte el puerto espacial, que deben coincidir. Se requiere autenticación para realizar esta petición.

En caso de que no haya ningún puerto espacial que acepte el peso de la nave que quiere aterrizar, se responderá informando de esta situación.

Implementad el código que ayudará a manejar el aterrizaje de naves espaciales.