

SERVICIOS WEBV CON SPRINGBOOT LINEA AEREA



El primer problema que encontré al empezar a desarrollar la aplicación fue con el proyecto del servidor, en este caso al crear los DTOs de cada entity pensé que el valor de estos podía ser auto incremental, así que el servidor cuenta con una variable global para simular el valor autoincremental de una base de datos relacional.

```
//Incrementa el id de forma automática  
2 usages  
private int count = 0;
```

El disparador del servidor daba problemas, pretendía conectarse a una base de datos en lugar de al localhost busque información en la red y detecte que debía importar la clase DataSourceAutoConfiguration del paquete org.springframework.boot.autoconfigure.jdbc. Es parte del framework Spring Boot y se encarga de la configuración automática del origen de datos (DataSource) en la aplicación. Sin embargo, en algunos casos, como cuando la aplicación no necesita acceder a una base de datos, puede ser necesario excluir esta configuración para evitar errores de arranque.

```
package org.educa.airline;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration;  
  
@SpringBootApplication(exclude = {DataSourceAutoConfiguration.class})  
public class Main {  
    public static void main(String[] args) {  
        SpringApplication.run(Main.class, args);  
    }  
}
```

Fuente: <https://stackoverflow.com/questions/36387265/disable-all-database-related-auto-configuration-in-spring-boot>

Al no solicitarse en la práctica métodos para incluir vuelos en el controlador, ha se procedido a crear un método init(){} para agregar un vuelo, y así poder realizar consultas a los repositorios correspondientes creo que esto podría simular una base de datos aislada, resulto no ser la solución más adecuada y ahora los vuelos se pueden crear de forma manual.

```
// Metodo de inicializacion que se ejecuta despues de que se haya construido el controlador  
@PostConstruct  
public void init() {  
    // Crear un vuelo de ejemplo y agregarlo al servicio de vuelos  
    Date date = new Date(); // Fecha actual  
    Flight flight = new Flight(id: "ABC123", origin: "New York", destination: "Los Angeles", date)  
    flightService.addFlight(flight);  
}
```

Problemas con el dto del vuelo a la hora de introducir la fecha lo solucione dando un formato nuevo de tipo string a la fecha del dto, pero a remitirla desde el servidor sigue dando problemas,

```
12 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class ProductFlightDTO {
    private String id;
    private String origin;
    private String destination;
    private String dateString;
}
```

Las partes más importantes de la api están comentadas pero, no tuve tiempo de hacer javadoc.