Ingesta de tareas batch

Desde el sistema de reservas de Viajes El Corte Gallego cada día nos pasan una tarea **batch** con las reservas que han hecho sus agentes comerciales para confirmarlas con nosotros (nosotros somos el tour operador Expedición Mórbida).

Instrucciones

- Nos dan el esqueleto de la aplicación, es **obligatorio** usar este esqueleto MAVEN.
- Hay que modificar el **pom.xml** para que incluya el driver de la base de datos
- Hay que usar la configuración que viene por defecto en el docker-compose que se da (ver carpeta stack) para que conecte con la base de datos. **No se puede modificar el docker-compose.yml**. Atención con los datos que vienen en este archivo.
- Levanta los nuevos contenedores antes de comenzar a programar.
- La base de datos se llamará **reservas**, *NO SE ADMITIRÁ OTRO NOMBRE*.
- Si la base de datos no está creada, nuestro programa debe crearla.
- Si las tablas no están creadas, nuestro programa debe crearlas.
- **Sólo hay un main** todo lo tiene que hacer el mismo programa.
- Las tablas deben tener sus respectivas claves foráneas.
- Se puede elegir hacerlo con XML o con JSON, las dos opciones valen igual.
- Hacer los DAO necesarios para todas las entidades.
- Cada clase entidad en un archivo diferente, no se pueden poner todas en el mismo archivo.
- Hay que hacer una rutina de carga del JSON a MySQL que se ejecute desde maven (ya está configurado el pom.xml). Puedes probar como funciona ejecutando ***mvn exec:java***. Antes de meter los datos en la base de datos, si no existen, por ejemplo, algunos clientes, deberán ser introducidos previamente a la inserción, o bien modificar el DAO para que arregle, que si da una excepción que no está esa clave foránea, de de alta el cliente para poder seguir (como veremos que se puede hacer con herramientas ORM).

La **tarea batch** no es más que un JSON con este formato:

Version JSON

```
{
   "reservas": [
      {
         "reserva": {
            "cliente": {
               "id": 1,
               "nombre": "Juan Perez",
               "contacto": {
```

```
                    "email": "juan.perez@email.com",
                    "telefono": "+123456789"
                }
            },
                "alojamiento": {
                    "id": 2,
                    "tipo": "hotel",
                    "nombre": "Hotel Ejemplo",
                    "direccion": "Calle Principal 123, Ciudad",
                    "telefono": "+987654321"
                },
                "entrada": "2023-01-01",
                "salida": "2023-01-05",
                "pension": "pension_completa"
        }
    }
]
}
```

Version XML

```
<?xml version="1.0" encoding="UTF-8"?>
<reservas>
   <reserva>
      <cliente>
         <id>1</id>
         <nombre>Juan Pérez</nombre>
         <contacto>
               <email>juan.perez@email.com</email>
               <telefono>+123456789</telefono>
         </contacto>
      </cliente>
      <alojamiento>
         <id>2</id>
         <tipo>hotel</tipo>
         <nombre>Hotel Ejemplo</nombre>
         <direccion>Calle Principal 123, Ciudad</direccion>
         <telefono>+987654321</telefono>
      </alojamiento>
      <entrada>2023-01-01</entrada>
      <salida>2023-01-05</salida>
      <pension>pension_completa</pension>
   </reserva>
</reservas>
```

Hay que implementar lo siguiente:

1) Nos piden que hagamos un programa que introduzca este JSON en la base de datos. Deberás crear, desde Java, las tablas si no existen, y además crear los clientes y otras tablas necesarias antes de las reservas.

2) Crea un disparador (o impleméntalo en JAVA) que impida que una misma persona haga reservas para fechas que ya tiene otra reserva anterior.

Puedes probar si el punto 2 funciona con esta nueva tarea batch:

Versión JSON:

```json
{
    "reservas": [
      {
      "reserva": {
         "cliente": {
            "id": 1,
            "nombre": "Juan Pérez",
            "contacto": {
               "email": "juan.perez@email.com",
               "telefono": "+123456789"
            }
         },
         "alojamiento": {
               "id": 2,
               "tipo": "hotel",
               "nombre": "Hotel Ejemplo",
               "direccion": "Calle Principal 123, Ciudad",
               "telefono": "+987654321"
         },
         "entrada": "2023-01-02",
         "salida": "2023-01-04",
         "pension": "pension_completa"
      }
      }
    ]
 }
```

Version XML:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<reservas>
    <reserva>
        <cliente>
            <id>1</id>
            <nombre>Juan Pérez</nombre>
            <contacto>
                <email>juan.perez@email.com</email>
                <telefono>+123456789</telefono>
            </contacto>
        </cliente>
        <alojamiento>
            <id>2</id>
            <tipo>hotel</tipo>
            <nombre>Hotel Ejemplo</nombre>
```

```
            <direccion>Calle Principal 123, Ciudad</direccion>
            <telefono>+987654321</telefono>
        </alojamiento>
        <entrada>2023-01-03</entrada>
        <salida>2023-01-04</salida>
        <pension>pension_completa</pension>
    </reserva>
</reservas>
```