Examen 1º EVA Lucía

# Lo que puede entrar:

* El configuration en un txt y un xml
* Textfiles
* Ejercicios en el main
* XML Files
* SQL Queries y modelos
* Inyección de dependencias
* JDBC

## Configurations

* TXT y xml:

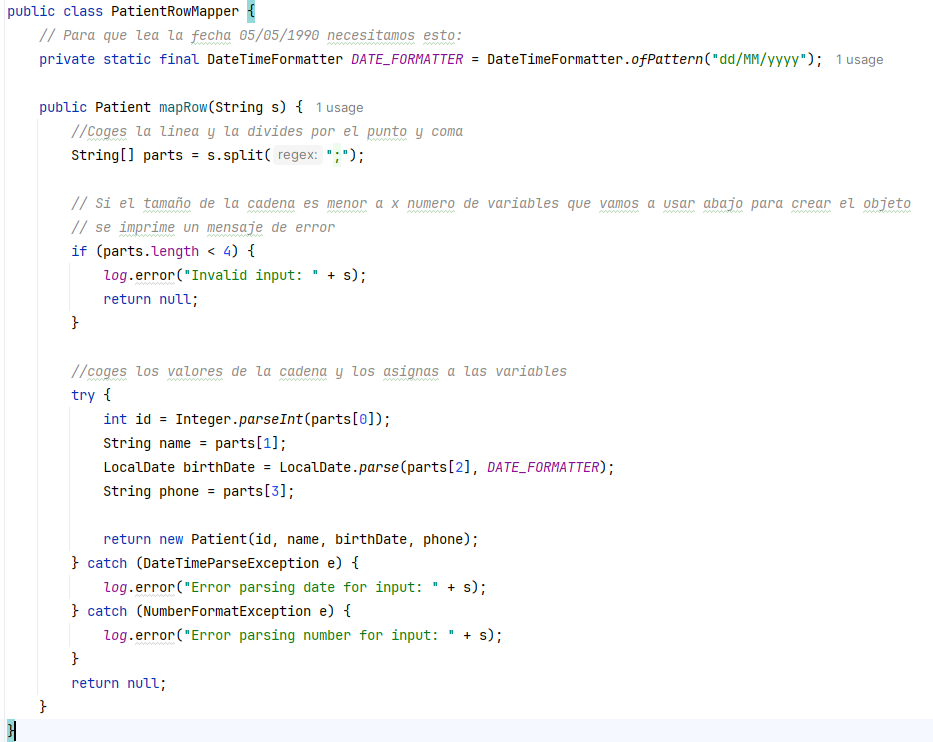
Hay dos clases en las que se configura, la clase configuration y el archivo txt o xml del que tira:

* + Clase configuration: en el paquete common/config:
  + El txt o xml del que se basa, con las variables que luego accede:

## Text Files

Son archivos csv que se guardan en la carpeta data en la ruta raíz, los llamas desde el dao.

* **Cargar contenido a ram desde el archivo de texto**

Antes de nada necesitas una manera de pasar cada línea que veas a un objeto. Se puede hacer de varias maneras, pero lo hacemos con un rowmapper:

Ahora, una vez con el rowmapper configurado:

*// Metodo que usa el rowmapper*private Patient mapToPatient(String line) {  
 return rowMapper.mapRow(line);  
}

*// Metodo que lee el archivo de pacientes*private List<String> readFile() {  
 List<String> lines = new ArrayList<>();

*// en la configuracion*  
 Path filePath = Paths.*get*(config.getPathPatients());

*// Se lee con bufferedReader en este caso porque es mas*

*eficiente para archivos grandes* try (BufferedReader reader =

Files.*newBufferedReader*(filePath))

{  
 String line;  
 while ((line = reader.readLine()) != null) {  
 lines.add(line);  
 }

} catch (IOException e) {  
 throw new InternalServerErrorException("Error

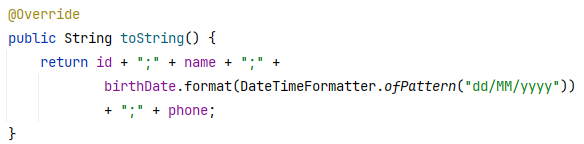
reading the patient file "+ e);

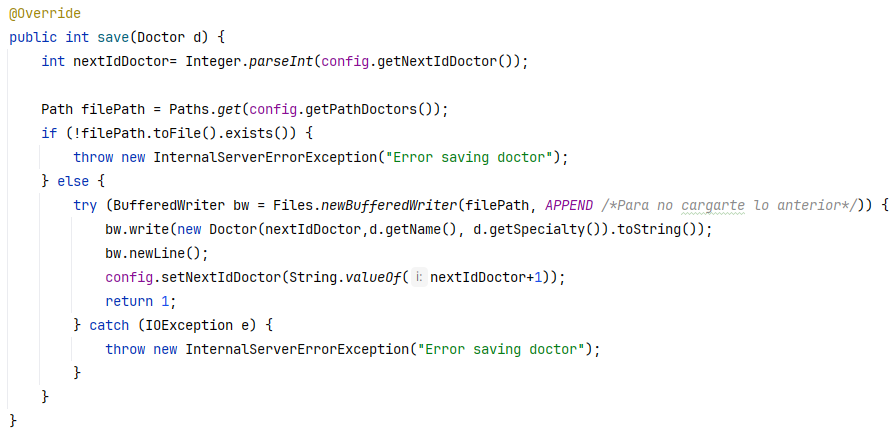
}  
 return lines;  
}

Otro ejemplo que usa readAllLines para archivos pequeños:

* **Guardar contenido a el archivo hacia el archivo de texto**

Antes de nada tienes que tener el tostring bien configurado en el objeto que quieras guardar, hazlo con un tostring que no sea el de por defecto por si acaso, aunque lo haga asi yo aqui:



Luego, iteras sobre la lista y vas agregandolos uno a uno, la unica complicacion esá en sacar el ultimo id, pero eso se guarda en una variable cada vez que haces un getAll, llamas al metodo guardarLastId y lo guardas en una variable o algo así.

* **Eliminar contenido del archivo de texto**