

PRÁCTICA 1

IMPORTACIÓN Y EXPORTACIÓN DE DATOS EN PIZZERIA APP

Curso 2024 - 2025

UNIDAD 1. GESTIÓN DE FICHEROS Y SERIALIZACIÓN DE OBJETOS

ACCESO A DATOS – 2º DAM

I.E.S THIAR

Implementa en Java los siguientes ejercicios, utilizando como base la práctica anterior. Al finalizar, adjunta el proyecto comprimido en un zip en la tarea con esta nomenclatura: Apellido1-Nombre-PracticaUD1.zip. Para la realización de esta práctica hay que seguir las buenas prácticas de programación:

- **Seguir las “naming conventions” de Java.**
- **Gestionar de forma eficiente los recursos externos.**
- **Control de posibles excepciones.**
- **Encapsulamiento.**
- **Principio de Responsabilidad Única.**
- **Baja complejidad ciclomática. Mantener simplicidad de las funciones.**

(3 puntos) Actividad 1. Gestión básica de ficheros. La aplicación tendrá una serie de usuarios, llamados administradores, que tendrán más privilegios que los clientes. La aplicación debe permitir su lectura sin utilizar librerías externas a través de un fichero llamado *admin.txt*. Estos administradores deben mapearse a objetos Cliente, de tal manera que los puedas diferenciar como administradores del resto de objetos cliente. Ejemplo de formato del fichero admin.txt:

```
1;12345678A;Juan Pérez ;Calle Falsa 123; 600123456; juan@example.com ;securepass
2 | 98765432B | Ana García | Avenida del Sol 45 | 650987654 | ana@example.com | password123
3,45612378C,Carlos López , Plaza Mayor 10 , 612345678 , carlos@example.com , abcde12345
```

Características del fichero admin.txt:

- Se debe tener en cuenta que los delimitadores pueden ser “;”, “|” o “,” pero son consistentes en toda la línea, es decir, no varía el delimitador en la misma línea.
- Pueden contener espacios sobrantes.

(3 puntos) Actividad 2. JAXB. La aplicación debe permitir la exportación e importación de objetos Cliente a formato XML utilizando la API JAXB. Deben mapearse a elementos xml todos los campos, a excepción del id, que deberá ser un atributo XML.

(1,5 puntos). EXPORTACIÓN. Ejemplo de formato de salida (fichero clientes.xml), dada una lista de objetos cliente.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<clientes>
  <cliente id="1">
    <apellido>Uribe</apellido>
    <direccion>C/Mayor 10</direccion>
    <dni>11102496E</dni>
    <email>j.uribemontoya@edu.gva.es</email>
    <nombre>Javier</nombre>
```

```
<password>password123</password>
<telefono>+34111929992</telefono>
<admin>false</admin>
</cliente>
<cliente id="2">
  <apellido>Gómez</apellido>
  <direccion>C/Sol 20</direccion>
  <dni>12345678Z</dni>
  <email>ana.gomez@example.com</email>
  <nombre>Ana</nombre>
  <password>password456</password>
  <telefono>+34678901234</telefono>
  <admin>false</admin>
</cliente>
</clientes>
```

(1,5 puntos). **IMPORTACIÓN.** La aplicación debe permitir la importación de un fichero XML para transformarlo en una lista de objetos Cliente, utilizando el formato de salida anteriormente indicado.

(4 puntos) Actividad 3. OpenCSV. La aplicación debe permitir la exportación e importación de objetos Ingrediente a formato CSV, utilizando la API OpenCSV.

(2 puntos). **EXPORTACIÓN.** Ejemplo de formato de salida (fichero ingredientes.csv), dada una lista de objetos ingrediente y utilizando el separador punto y coma (",") para separar los campos y coma(",") para separar los alérgenos.

```
"ALERGENOS";"ID";"NOMBRE"
"Lactosa,Glucosa";"0";"Queso mozzarella"
"Sulfitos,Lactosa";"0";"Pepperoni"
"";"0";"Salsa de tomate"
"Gluten";"0";"Masa de pizza"
"";"0";"Champiñones"
```

(2 puntos). **IMPORTACIÓN.** La aplicación debe permitir la importación del fichero CSV para transformarlo en una lista de objetos Ingrediente, utilizando el formato de salida anteriormente indicado.