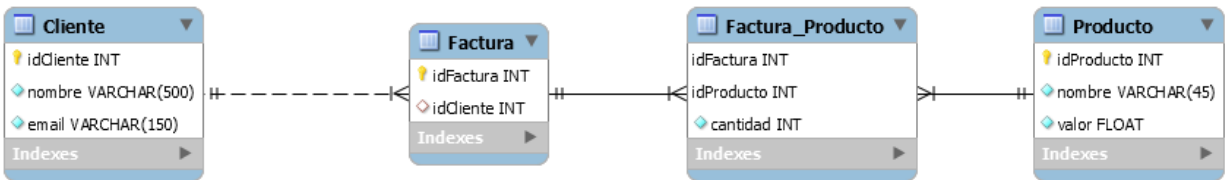


Trabajo Práctico No. 1: Persistencia

- 1) En base al Video 1: JDBC-Java, realizar la configuración de Derby y testear la base de datos con algunas consultas sobre la tabla Persona.
Video 1: <https://www.loom.com/share/a9904b69ecc04892827b740b7d78432a>
- 2) En base al Video 2: JDBC + MySQL, realizar la configuración de MySQL (como sugerencia, utilizar también Docker) y testear la base de datos con las mismas consultas sobre la tabla Persona realizadas en el ejercicio 1.
Video 2: <https://www.loom.com/share/1014c5b740d5491690676ff448209959>
- 3) Refactorizar lo realizado en los Ejercicios 1 y 2 utilizando un patrón DAO que permita abstraer los detalles de la conexión a cada tipo de base de datos y las consultas. Crear una clase Persona para encapsular los datos de la misma, e implementar un servicio que retorne una lista de todas las personas cargadas en la base de datos.
- 4) Enumere brevemente las ventajas de la implementación basada en DAO del ejercicio anterior, respecto a las implementaciones de los ejercicios 1 y 2. Adicionalmente, considere posibles desventajas.
- 5) En base al Video 3: JPA-MySQL, re-implementar las consultas del ejercicio 2, pero esta vez utilizando una implementación basada en JPA y Hibernate.
Video 3: <https://www.loom.com/share/fea19216d50f42aeb9ae2fc7e924ec4b>
- 6) En base al Video 4: JPA-Derby, re-implementar las consultas del ejercicio 1, pero esta vez utilizando una implementación basada en JPA y Hibernate.
Video 4: <https://www.loom.com/share/5d807aa6d03349958f7be2a487ae51a6>
- 7) Enumere brevemente las ventajas de la implementación basada en JPA de los ejercicios 5 y 6, respecto a las implementaciones sin JPA de los ejercicios 1 y 2. Adicionalmente, considere posibles desventajas.
- 8) Extienda el diseño basado en DAO del Ejercicio 3 para incorporar las implementaciones basadas en JPA y Hibernate de los ejercicios 5 y 6. Crear una clase Persona para encapsular los datos de la misma, e implementar un servicio que retorne una lista de todas las personas cargadas en la base de datos.

Ejercicio Integrador

Considere el siguiente diagrama de base de datos:



- 1) Cree un programa utilizando JDBC que cree el esquema de la base de datos.
- 2) Considere los CSV dados y escriba un programa JDBC que cargue los datos a la base de datos. Considere utilizar la biblioteca Apache Commons CSV, disponible en Maven central, para leer los archivos.

```
CSVParser parser = CSVFormat.DEFAULT.withHeader().parse(new
    FileReader("productos.csv"));
for(CSVRecord row: parser) {
    System.out.println(row.get("idProducto"));
    System.out.println(row.get("nombre"));
    System.out.println(row.get("valor"));
}
```

- 3) Escriba un programa JDBC que retorne el producto que más recaudó. Se define "recaudación" como cantidad de productos vendidos multiplicado por su valor.
- 4) Escriba un programa JDBC que imprima una lista de clientes, ordenada por a cuál se le facturó más.