

Práctica 2. Relaciones y Aggregation Framework.

Apartado 1: relaciones y proyecciones

Se desea implementar la capa de persistencia para una aplicación de gestión de asistencia online. En concreto, se trata de almacenar los datos sobre chats que los técnicos y clientes realizan sobre los productos que se venden. La implementación se hará con Java y Spring Data (Java 8) sobre una base de datos MySQL.

Este apartado de la práctica parte del escenario descrito en la versión 1 de la práctica 1, donde se tiene una implementación completamente relacional. En la base de datos se definían las siguientes entidades, con los datos que se indican a continuación:

- Cliente: nombre, apellidos, correo electrónico y ciudad.
- Producto: nombre, marca, precio.
- Técnico: nombre, nivel (dato numérico de 1 a 10).
- Chat: fecha, autor de la entrada (cliente o técnico, puede ser un campo similar a “binario” o *booleano*), texto de la entrada, producto sobre el que se chatea, cliente que lo compró (con quien se está chateando) y técnico que le atiende.

Se desea implementar una jerarquía con las siguientes características:

- Todos los productos tienen nombre y precio.
- Existe un tipo de producto denominado **servicio** que incluye el nombre del proveedor, las fechas de inicio y fin de su contratación, y un campo numérico denominado promoción que indica el porcentaje de descuento aplicado sobre el precio inicial.
- Existe otro tipo de producto denominado **retail** que incluye la marca del artículo y el país de origen.

Es necesario además almacenar una relación entre clientes y productos, de manera que un cliente pueda comprar muchos productos y un producto lo puedan comprar muchos clientes. En esa relación se debe incluir la fecha de compra.

Se pide crear las correspondientes entidades que modelen este escenario de jerarquía y relaciones, así como insertar datos de prueba y proporcionar un código que muestre el correcto almacenamiento de la información.

También se pide implementar y mostrar los resultados de las siguientes consultas:

- Ciudad cuyos clientes han realizado más compras.
- Para cada cliente, mostrar su nombre, apellidos y el número de servicios que ha comprado.

Apartado 2: Aggregation Framework

A partir del fichero de ejemplo denominado “Provincias.json” disponible en Aula virtual, se debe indicar el código que resuelva las siguientes consultas desde el terminal cliente de MongoDB:

- Año con mayor población en España (suma de todas las provincias), indicando tanto el año como el valor.
- Para mostrar cómo ha evolucionado la población, se pide mostrar, en un documento por provincia, la población del primer año almacenado, el último y la diferencia.

Formato de entrega

La práctica se entregará teniendo en cuenta los siguientes aspectos:

- La práctica se entregará como un fichero .zip que tendrá un directorio con el proyecto Maven que resuelve el apartado 1 y un fichero de texto con las consultas del apartado 2.
 - En el directorio del proyecto solamente hay que incluir *pom.xml* y el directorio *src*.
 - El proyecto se puede crear con cualquier editor o IDE.
- El nombre del fichero .zip será el correo URJC del alumno (sin @alumnos.urjc.es).
- La práctica se entregará por Aula Virtual según la fecha indicada.

Las prácticas se podrán realizar de forma individual o por parejas. En caso de que la práctica se haga por parejas:

- Sólo será entregada por uno de los alumnos.
- El nombre del fichero .zip contendrá el correo de ambos alumnos separado por guión. Por ejemplo *p.perezf2019-z.gonzalez2019.zip*