Comentarios Correcciones 16/9/2017

A Java hay que decirle el conjunto de directorios para encontrar clases o archivos con el Class Path.

Para Testing se utiliza otro Class Path.

Eclipse sólo maneja un Class Path, entonces a mano hay que decirle el class path correcto.

Hay que importar de vuelta como eclipse Project y va a andar.

Manejando transacciones a mano, idea de los Test:

Base de memoria se descarta al cerrar la máquina virtual pero entre test y test persiste. Entonces los test son dependientes unos de los otros.

A los test hay que agregarle “extends abstractPersistenceTest”.

Cuando arranca otro test empieza otra sección, rollbackea automáticamente al final de cada test. Así los test van a ser realmente independientes.

Mirar annotation @beforeClass, JPA SCHEMAGENERATOR, SCHEMA EXPORT

Cambiar los nombres de las tablas a plural y cambiar los nombres de las columnas para usar guión bajo en lugar de escribir ejemplo “tipoDeDato”.

(no es necesario, solo informativo) Para realizar S = A + 1 \* 2, de la forma implementada son necesarias 5 consultas y un join bastante complejo. Lo mejor sería parsear y desparsear para evitar tantas consultas. Un converter puede serializar como String. Pero esto no es compatible con otra cosa que no sea java, en cambio la forma implementada sí. En la práctica no se persiste el árbol, sólo el string. Esto no hace falta cambiarlo, es sólo para tenerlo en cuenta como experiencia.

JPA EXTRA también tiene converter a Local Date.

En repositorios hay cosas para reutilizar seguro.

Clase singleton, una transacción para toda la aplicación.

Nunca una transacción dura la vida de un objeto, mucho menos de un singleton.

Transacción es un objeto transitorio.

Si hay un begin y un commit, debe haber un rollback (en un try catch).

Boiler plate -> código que es necesario pero puede generar problemas.

Mirar “implements transactionalOps” y usar en código withTransaction (en los test de contexto se usa), se le pone el código adentro y es equivalente pero mucho mejor a utilizar créate transaction, commit y rollback.

Lazy initialization sólo tiene sentido para objetos pesados y complejos.

Sincronizado es para acceso concurrente.

A createQuery(… , Empresa.class) se le pasa el tipo de lo esperado. Tenemos código muy parecido pero que cambia por los tipos. Esto se soluciona con generics (ver código de ejemplo recibido por email).

Queries más complejas where metodología.nombre =:nombre

Set parameter(“nombre”, nombre)

(se había realizado este comentario porque en el código en una query sql se utilizaba un alias para referenciar a la única variable que se utilizaba en la consulta).

@column es sólo para escalares, sino hay que usar joinedColumn.

Nunca utilizar arrays (no se puede cambiar el tamaño y tiene pocas operaciones), utilizar listas o collections.

No usar For, ni arrays en el código. En lugar de For utilizar streams.

No utilizar calendar, es un gran dolor de cabeza y encima es mutable. Es mejor local date, le pedís year y te devuelve tipo year, no entero.

Arrays.asList(empresa.longevidad())

El constructor público se hereda por defecto.

Super está implícito.

Minimizar los setters. Desde arena no se modifica las cuentas, entonces usar sólo get.

En aplicar condiciones hay un Fold de libro, hay que cambiarlo y buscar el equivalente en Stream.

Equals hashMap

Generate HashCode

Otra opción Apache commons.

Java objects.equals

Objects.hashcode

Nunca más hacer un hashcode a mano.

Importante borrar el código que no se utiliza (hay mucho código comentado, dijeron que lo iban a controlar en la próxima corrección), y realizar las abstracciones de repositorio.