Práctica 8

Objetivos

- Construir una aplicación basado en el patrón modelo de dominio y con mapeador JPA
- Evolucionar la arquitectura de capas de la versión SL.TS.TDG_0 para integrar el modelo de dominio y el mapeador

Ejercicios

Evolucionar la arquitectura desde de la versión SL.TS.TDG_0

Sigue este orden:

- Carga en el Eclipse el proyecto que acompaña a esta práctica.
- Copia tus clases de modelo del dominio desarrolladas en la práctica anterior al paquete `uo.ri.cws.domain`.
- Levanta la base de datos HSQLDB adjunta con los datos de la aplicación CarWorkshop.
- Observa la distribución de proyectos. Una aplicación de presentación por cada actor del sistema (están ya completas) y el proyecto para los servicios.
- Observa las interfaces de la capa de servicio, son las mismas que las de la práctica anterior. Observa el uso de los patrones servicio, fachada, factory y DTO.
- Rehaz las clases que implementan los *Transaction Script* (las del paquete `uo.ri.cws.application.service.*.*`). Ahora se deben implementar usando las clases del modelo y repositorios. Empieza por los casos CRUD de mecánico, y el último el de la Factura.

Centralizar el control de transacciones

- Añade las interfaces *`Command`* y *`CommandExecutor`*.
- Haz que todos los *Transaction Script* implementen *`Command`*.
- Modifica las clases que implementan las fachadas de forma que cada TS lo ejecute el `CommandExecutor` (una línea de código por cada método).
- Añade la clase de utilidad *`Jpa.java`* a la capa de persistencia (está en la carpeta `./material` del esqueleto). Echa un vistazo a su contenido y asegúrate de que sabes para qué sirve cada uno de los métodos públicos.
- Implementa el `CommandExecutor`. Haz uso de la clase de utilidad *Jpa* recién añadida.
- Añade una clase Factoría en el paquete `uo.ri.conf` para obtener una instancia del `CommandExecutor`.
- Modifica cada TS eliminando el código redundante de control de la transacción. Deben hacer uso de los repositorios, y estos, a su vez, de la clase de utilidad `Jpa.java`.

Complete the repositories

- Observa el paquete `uo.ri.cws.application.repository`, contiene todas las interfaces de los repositories. Más o menos uno por cada entidad. Verás que todos heredan de la interfaz base Repository. Los repositorios usan la metáfora la colección (add y remove, pero no update), no son TDG ni DAO.
- El paquete `uo.ri.cws.infrastructure.persistence.jpa` contiene las implementaciones de esas interfaces. Hay una implementación base para los métodos comunes. Solo hay que completar las queries específicas para cada repositorio. Recuerda recuperar el mapeador con la clase de utilidad.
- Añade las consultas necesarias en las implementaciones de los repositorios.
    * Cada método realiza una consulta expresada en JPQL que debe ir localizada en el fichero `orm.xml`.
    * Por lo menos añade las de *MechanicJpaRepository*, *InvoiceJpaRepository* y *WorkOrderJpaRepository* con los métodos necesarios.