Taller 4 y Proyecto Final - Capa RESTFull e Integración
Computación en Internet

Objetivos:

- Implementar la presentación utilizando Thymeleaf.
- Implementación de servicios RESTFull separando la lógica del proyecto anterior para implementar el cliente y el servidor separados.
- Implementar patrón de diseño Business Delegate como cliente para la capa RESTFull.
- Implementar pruebas de unidad para la lógica que den cubrimiento al código y a los valores límite.
- Manejar las transacciones dentro de la lógica de negocio con Spring.

Se debe entregar la aplicación Web utilizando Thymeleaf, REST y Spring Data / JPA, integrando el código de uno de los dos participantes entregado previamente.

Este trabajo tendrá 2 notas, correspondientes al taller 4 y el proyecto final. A continuación, se detallan los aspectos a evaluar en cada una de las entregas.

Para esta actividad, no se requiere separar en proyectos independientes el front y el back, aunque si debe garantizarse que la única comunicación entre los dos es por medio de los llamados REST. El paquete con el modelo puede ser compartido entre las capas del front y el back.

Actividades Taller 4:

Separación de las capas del Front y del Back para las funcionalidades desarrolladas previamente:

1. (1.5) Implementación del cliente RESTFull como un delegado de negocio (Business Delegate):
    a. (0.6) El cliente está completamente implementado para las funcionalidades requeridas por la presentación (controladores frontales).
    b. (0.3) No hay ningún llamado directo de los controladores frontales a los servicios RESTFull o a la capa de la lógica.
    c. (0.6) Los controladores frontales utilizan los Bean delegados y funcionan correctamente.
2. (1.5) Pruebas para el delegado:
    a. (0.5) Se encuentran implementadas las pruebas unitarias para todas las funcionalidades del delegado.
    b. (1.0) Las pruebas para el delegado utilizan Mocks.
3. (1.5) Servicios RESTFull:
    a. (0.6) El back-end está completamente implementado para las funcionalidades requeridas por la presentación (controladores) por medio del delegado.
    b. (0.3) La definición de los servicios cumple con los requisitos a **nivel 2** de RESTFull (URLs, verbos y nombres).
    c. (0.6) Los servicios REST se pueden utilizar correctamente.
4. (0.5) Despliegue utilizando una base de datos Postgres.

Actividades para proyecto final:

Implementación para cubrir la funcionalidad referente a las revisiones de productos (productreview), las cuentas de materiales (billofmaterials) y consultas previamente desarrolladas:

1. (2.0) Realizar una gestión de revisiones de productos:
    a. (1.0) Una pantalla que permita que se creen, actualicen y borren las revisiones de productos, incluyendo que desde el producto se pueda acceder a las revisiones.
    b. (1.0) Implementar la lógica y repositorios/DAOs necesarios para soportar las revisiones de productos.
2. (2.0) Realizar una gestión para las cuentas de materiales:
    a. (1.0) Una pantalla que permita que se creen, actualicen y borren las cuentas de materiales con la unidad de medida y el producto, incluyendo que desde el producto se puedan consultar la cuenta de materiales.
    b. (1.0) Implementar la lógica y repositorios/DAOs necesarios para soportar l gestión de la cuenta de materiales.
3. (0.5) Implementar las pantallas que permitan ejecutar las consultas del taller anterior
    a. (0.25) Mostrar los encabezados de órdenas de producto (s) con la suma de los detalles de órdenes ordenados por la fecha de la órden, permitiendo seleccionar el rango de fechas.
    b. (0.25) Mostrar el listado de encabezados de órdenes que tienen al menos dos detalles de órdenes de compra.
4. (0.5) Despliegue utilizando una base de datos Postgres.