

Taller 1 - Diseño e Implementación de Pruebas Unitarias y de Integración en Spring

Computación en Internet 20-1

Objetivos de la unidad 3:

- Diseñar e implementar pruebas unitarias y de integración.
- Permitir el acceso básico a un modelo de datos por medio de repositorios.
- Implementar servicios que consuman los repositorios.
- Implementar pruebas usando herramientas como Junit o TestNG.
- Implementar pruebas en Spring Framework.

Planeación de pruebas:

En la parte final del documento encontrará el enunciado del sistema que se propone desarrollar y las tablas actualmente definidas para soportar el sistema. Cualquier cambio en el modelo de datos deberá ser validado con el profesor.

Acorde al enunciado del problema y a los pedidos que se encuentran abajo, debe realizar el diseño de pruebas tanto de integración como unitarias. En el documento a entregar debe ser clara la correspondencia entre el diseño, el tipo de prueba y las pruebas implementadas. Se recomienda enumerarlas y utilizar esta enumeración en los comentarios de los métodos de prueba.

Actividades:

Preparación del proyecto:

1. Crear un proyecto Spring Boot incluyendo `spring-boot-starter-data-jpa`, `Junit 5` y `mockito`.
2. Crear un paquete modelo e importar las clases entregadas acorde al enunciado del proyecto.
3. Crear los paquetes para los servicios, los repositorios y sus pruebas.
4. Implementar los repositorios para los temas, historias, cronómetros, grupos, Sprints y juegos con las operaciones básicas. Puede ignorar las entidades no mencionadas en este punto, para el alcance de este taller.

Pruebas Unitarias (en cada numeral 20% diseño, 80% implementación de las pruebas unitarias con todos los casos relevantes).

5. Realizar las pruebas definiendo los mocks necesarios en cada caso y la posterior implementación acorde al requerimiento en cada punto siguiendo TDD. Debe incluir la información inicial para la prueba, de tal forma que se puedan encontrar el tema, historias y cronogramas necesarios:
 - a. (0.5) Proveer servicios para guardar y editar un *tema* sin ninguna asociación, pero garantizando que la cantidad de Sprints y grupos sea mayor a cero.
 - b. (0.5) Proveer servicios para guardar y editar un *juego* sin ninguna asociación, pero garantizando que la cantidad de Sprints y grupos sea mayor a cero. En caso de recibir un tema, se debe validar que este exista antes de guardarlo (del tema se recibe el identificador y se busca).
 - c. (0.5) Proveer servicios para guardar y editar una *Historia* sin ninguna asociación adicional pero asociada a un *juego*. Debe garantizar que el valor de negocio, el sprint inicial y la prioridad sean mayores que cero. Se debe validar que este exista el juego antes de guardarlo (del juego se recibe el identificador y se busca).
 - d. (1.0) Modificar el servicio de guardar juego que, adicional a lo ya pedido, considere que se crea a partir de un tema de la siguiente forma. Debe asociar las historias y cronogramas que se encontraban asociados al tema al nuevo juego creando una copia

de los mismos. Es decir, las historias y cronogramas originales deben quedar asociados al tema y las copias de ellas al juego. Para efectos del ejercicio cree una copia del anterior y llámelo por ejemplo guardarJuego2.

Pruebas de Integración (en cada numeral 10% diseño, 50% implementación de las pruebas de integración con todos los casos relevantes, 40% implementación del servicio solicitado).

6. Realizar las pruebas de integración, y la posterior implementación acorde a los requerimientos del numeral anterior. Debe conservar las pruebas unitarias, puede realizar una copia de la clase para realizar las pruebas de integración.
 - a. (0.5) Temas
 - b. (0.5) Juegos1
 - c. (0.5) Historias
 - d. (1.0) Juegos 2

Debe diseñar e implementar todos los casos necesarios para cada uno de los requerimientos acorde a las validaciones pedidas para cubrir todos los posibles errores o terminaciones exitosas. El diseño debe entregarse en un documento Word o PDF.

Enunciado: Sistema de gestión de actividades prácticas

El departamento TIC de la Universidad Icesi requiere un sistema que le permita mejorar la gestión de las actividades prácticas que realiza en grupos con estudiantes de pregrado y postgrado. Estas actividades se realizan en diferentes cursos, con diferente cantidad de personas y con unos objetivos y tiempos definidos para cada uno de ellas.

El sistema debe poder soportar grupos de actividades, denominados temas, que pueden tener asociadas historias de usuario y cronómetros para controlar las actividades. Estos grupos de actividades se deben poder “instanciar” en juegos que hereden las historias y cronómetros del tema pero las pueda editar para el juego específico. Adicionalmente, un juego tiene unos sprints y grupos que pueden ser editados y gestionados.

Al momento de crear un juego, este puede crearse desde un tema, en cuyo caso debe copiar la información del tema y crear automáticamente la cantidad de Sprints y grupos acorde a lo definido en el tema. El sistema debe validar que no se creen 2 juegos en el mismo horario para un mismo día. El sistema no va a permitir borrar los juegos, pero si los sprints, grupos, historias y cronómetros.

El sistema debe validar que, al momento de editar una historia o un cronómetro para un juego, este no se encuentre asociado a un tema. En caso de encontrarse asociado, primero debe realizar una copia de la historia o cronómetro, para no modificar la original. Para efectos del presente proyecto, puede asumir que las historias no tienen criterios y que los cronogramas no tienen intervalos cuando están asociadas a un tema.

En la etapa inicial se han definido unas tablas que deben ser suficientes para almacenar la información referente a los elementos mencionados anteriormente. Se le ha proporcionado el diagrama que muestra las tablas y relaciones de modo que sea más fácil entender el alcance de la solución requerida.