

Taller 1 Diseño e Implementación de Pruebas Unitarias y de Integración

Computación en Internet 192

Objetivos:

- Diseñar e implementar pruebas unitarias y de integración utilizando TDD y Mockito.
- Implementar pruebas usando TestNG.
- Implementar pruebas en Spring Framework.

Actividades:

Preparación:

1. Crear un proyecto Spring Boot incluyendo `spring-boot-starter-data-jpa`, `testng` y `mockito` (no puede utilizar Junit para esta actividad).
2. Crear un paquete modelo e importar las clases entregadas para enunciado del proyecto.
3. Crear los paquetes para los servicios, los repositorios y sus pruebas.
4. Implementar los repositorios para las rutas, conductores, buses y servicios con las operaciones básicas. Puede ignorar las clases PK y 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 las pruebas d y e, de tal forma que se puedan encontrar las rutas, buses y conductores necesarios. (0.5 adicional si se realiza utilizando Stubs):
 - a. (0.5) Proveer un servicio para guardar un *Bus* sin ninguna asociación, pero garantizando que capacidad sea mayor a cero y el tipo sea T,P, o A y que el bus no sea nulo.
 - b. (0.5) Proveer un servicio para guardar un *Conductor* sin ninguna asociación, pero garantizando que las fechas de contratación y nacimiento se consistentes, es decir, no lo contraten antes de nacer y que el conductor no sea nulo.
 - c. (0.5) Proveer un servicio para guardar una *Ruta* sin ninguna asociación, pero garantizando que las horas de inicio y fin y el día de inicio y fin sean consistentes, es decir, no inicien antes de terminar y que la ruta no sea nula.
 - d. (0.5) Proveer un servicio para guardar una *Servicio* que garantice que se asocie tanto un conductor existe, como una ruta y un bus en el repositorio (mock inicialmente).
 - e. (0.5) Modificar el servicio anterior para garantizar que las fechas del servicio son consistentes entre sí y con la fecha de contratación del conductor.

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) Buses
 - b. (0.5) Conductores
 - c. (0.5) Rutas
 - d. (0.5) Servicios 1
 - e. (0.5) Servicios 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 transporte Masivo - MIO

El Sistema de transporte masivo de la ciudad de Cali requiere un sistema que le permita optimizar la utilización de sus buses y conductores, de tal forma que puedan ser asignados a los diferentes servicios que se deben prestar por las rutas definidas para el sistema. La solución debe validar que, al momento de crear un nuevo servicio, tanto el bus como el conductor se encuentren disponibles, es decir que no se encuentren asignados a otro servicio que coincida con las fechas, días y horarios establecidos.

Para ello, se debe almacenar el bus, el conductor y la ruta establecida y adicionalmente debe almacenar la fecha de inicio y fin establecida para el servicio.

En el sistema, se pueden ingresar nuevos buses con la única restricción de tener placa no repetida. De la misma forma se pueden ingresar nuevos conductores, en cuyo caso el identificador único es la cédula.

Se tiene definidas unas rutas que pueden funcionar determinados días de la semana entre unos horarios establecidos. Para ello, se han codificado los días de la semana con números, donde 1 corresponde al lunes, 2 al martes y sucesivamente hasta llegar al domingo. Al igual que en otros casos, los horarios de inicio y fin se manejan en segundos. Las rutas se identifican por un código (columna número) que por convención se ha definido como una letra y dos dígitos.

Existen algunas rutas que operan de lunes a viernes, de lunes a sábado, solo sábados y domingos y finalmente otras rutas operan solamente los sábados o solamente los domingos.

En la etapa inicial se han definido unas tablas que deben ser suficientes para almacenar la información referente a los actores 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.