

Mini-Proyecto Persistencia con JPA y Java EE

Computación en Internet

Pre-requisitos:

- Java Development Kit (JDK)
- STS con Database Development y Hibernate
- Postgress y el Driver (Maven)
- Usar el script-mio.zip

Objetivos:

- Implementar las lógica utilizando los DAOs existentes
- Implementar pruebas de unidad para la lógica que den cubrimiento al código y a los valores limite
- Manejar las transacciones dentro de la lógica de negocio con Spring.

Actividades:

1. Crear la lógica para las entidades que corresponden a las entidades: TMIO1_CONDUCTORES, TMIO1_RUTAS Y TMIO1_BUSES.
 - a. Permita que se creen, actualicen y borren las entidades, utilizando objetos de entidad pasados como parámetro.
 - b. Permita que los conductores se puedan buscarse por nombre, apellidos o cédula.
 - c. Permita que las rutas puedan buscarse por rango de días.
 - d. Permita que los buses puedan buscarse por modelo, tipo o capacidad.
2. Crear la lógica para las entidades que corresponden a la entidad TMIO1_SERVICIOS:
 - a. Permita que se creen, actualicen y borren los servicios, utilizando objetos de entidad pasados como parámetro.
 - b. Permita que los servicios puedan buscarse por rango de fechas.
3. Implementar las pruebas de unidad que permitan verificar la lógica de las entidades. Tenga en cuenta lo siguiente:
 - a. No se debe abrir ni cerrar transacciones a la base de datos desde las pruebas de unidad.
 - b. Las pruebas de unidad deben comunicarse sólo con la lógica de negocio
 - c. Los datos se deben persistir.
 - d. Utilizar el patrón de diseño DAO para la persistencia.
 - e. Mostrar la información completa en los logs y consultas.

Enunciado:

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.