Ejercicio (30 min.)

# SELF-CONTAINED y TRANSACTION-ROLLBACK

1. Revisa el método “find” y entiende su funcionamiento
2. ¿Qué pruebas realizarían para este método?

CHECKPOINT: ¿Pudiste encontrar al menos 4 pruebas?

1. Dentro del eclipse, elige un caso del método “find” que involucre pasar al menos un parámetro diferente de nulo, implementa su prueba.

TIP: ¿No sabes qué valor esperado poner en el Assert porque depende de los registros que se encuentren en la BD? Dentro de la misma prueba, antes del Assert, puedes cargar datos en la BD, de tal manera que la propia prueba conozca con que datos va a trabajar y no depender de los datos de la BD

Puedes utilizar el método “LoadData” que se encuentra dentro de la misma clase que contiene a las pruebas.

TIP: Para instanciar una fecha puedes utilizar “Date.valueOf("2010-12-30")”

1. Ejecuta una vez la prueba y observar que esta pase. (Si falla realiza las correcciones necesarias, elimina manualmente los registros de la BD y vuelve a ejecutar la prueba)
2. Ejecuta nuevamente la prueba. ¿Por qué la prueba falla? ¿Por qué tengo que eliminar los registros antes de ejecutar la prueba nuevamente?
3. Para eliminar los registros automáticamente, vamos a implementar el patrón TRANSACTION-ROLLBACK utilizando SPRING-TEST.

Para esto, primero necesitamos integrar JUNIT con SPRING, agrega las siguientes anotaciones a la clase “EmployeeJDBCTest”.

* @RunWith(SpringJUnit4ClassRunner.class)  
  Nos permite utilizar funcionalidad de spring dentro de JUnit, por ejemplo instanciar e inyectar las dependencias.
* @ContextConfiguration("jdbc-ctx.xml")  
  Nos permite indicar cuál es el archivo de configuración de Spring (“context file”) donde se encontrará la definición de las clases a instanciar e inyectar.

1. Agrega las siguientes anotaciones a la clase “EmployeeJDBCTest”, para habilitar el manejo de transacciones.

* @Transaction  
  Nos permite ejecutar cada método de la clase dentro de su propia transacción; por defecto, realiza un rollback al finalizar la transacción.

1. Ejecuta la prueba consecutivas veces y observar que siempre pase.
2. Dentro del eclipse, elige otro caso para el método “find” que involucre pasar al menos un parámetro diferente de nulo, implementa su prueba.

REFLEXIÓN: Para las 2 pruebas ya implementadas, ¿el número de registros cargados es igual al número de registros esperados? ¿Crees que sería mejor cargar más registros que el número de esperados?

1. Implementa una prueba para el método “créate”. ¿Esta prueba necesita inicializar datos previamente?
2. Implementa una prueba para el método “delete”.

REFLEXION:   
¿Qué pasaría si necesitamos probar el método “find” y no existe un método “create” para inicializar datos?

¿Qué pasaría si necesitamos probar los métodos “create” o “delete” y no existe un método “find” para consultar si el registro existe en la BD?