



Bases de Datos II

Trabajo Práctico Integrador: Etapa 3

Introducción

Esta segunda etapa del trabajo consistirá en implementar una alternativa de persistencia sobre el mismo modelo de la etapa 1 (Bithub), pero esta vez utilizando MongoDB como almacenamiento. Nuevamente, la aplicación consistirá en una arquitectura multicapa clásica, pero requerirá de la implementación de un nuevo **servicio** y **repositorio**, ambos adaptados a la tecnología MongoDB.

El nuevo servicio debe implementar la misma interfaz de la etapa 1, **BithubService**. Uno de los desafíos a la hora de mapear el mismo modelo a dos bases de datos con estructuras y formas de funcionar diferentes es cómo las mismas identifican a sus objetos. Hibernate, si bien es flexible, en general utiliza una estrategia de números enteros autoincrementales que la mayoría de los RDBMS permite. MongoDB, en cambio, implementa una estrategia particular que llama **ObjectId**. Debido a estas diferencias, mientras que la estructura y semántica de los tests a ser implementados se han mantenido, se ha parametrizado el modo en que se definen y obtienen los **ids** de los objetos a ser persistidos. La clase `BithubServiceTestCase` ahora tiene un tipo paramétrico permite definir el tipo particular de los ids y hay dos subclases (`HibernateBithubServiceTestCase` y `MongoDBBithubServiceTestCase`) que establecen el tipo concreto de dichos ids, así como también inyectan el service concreto a utilizar.

Finalmente, todos los objetos del modelo deberán heredar de una clase `PersistentObject`, que define ambos ids para todos los objetos de modo que todos puedan ser mapeados con ambas estrategias.

La cátedra entregará

- Una nueva clase de tests, que en términos de funcionalidad es exactamente igual que la de la etapa 1, pero con adaptaciones para ciertos conceptos de MongoDB (como tratamiento de **null** o uso de **id**).
- Una nueva configuración de Maven para incluir las librerías necesarias para trabajar con MongoDB con **PojoCodec**, un mapeador básico que convierte los objetos en documentos de MongoDB y viceversa.

Al igual que en la etapa 1, desde la línea de comandos debe obtenerse un build exitoso en donde pasen todos los tests.



Requisitos detallados que deben satisfacerse en esta etapa

1. Escribir todo el código necesario para que la aplicación compile y todos los tests de las subclases de `BithubServiceTestCase` pasen, tanto en su implementación de Hibernate como en la de MongoDB. Esto implicará codificar una nueva implementación concreta de la interfaz `BithubService`.
2. Correr `mvn clean install` debe resultar en un build exitoso en donde todos los tests pasen.
3. El código tiene que estar subido en el mismo repositorio que la etapa previa, pero en nuevo branch llamado `practica3`.
4. La forma de entrega va a ser solamente a través del repositorio privado, que deberá compartirse con el ayudante designado para el grupo.

Requisitos técnicos para esta etapa

1. Tener instalado JDK 8 (`javac -version` desde la línea de comandos debe funcionar)
2. Tener instalado Maven 3.5.0 (`mvn --version` desde la línea de comando debe funcionar)
3. Tener instalado MongoDB 4.0