

Introducción

En este documento se presenta la segunda parte del proyecto de migración del proyecto de casas rurales desarrollado con tecnología Java Swing, servicios Web JAX-WS y objectDB, a tecnologías JSF e Hibernate. Si se ha concluido con éxito la primera parte del proyecto, ya se dispondrá de un proyecto donde se ha reimplementado utilizando el framework Hibernate la capa de presentación de los casos de uso que permiten consultar y establecer ofertas de casas rurales (Query availability y Set availability).

En esta segunda parte del proyecto se pretende sustituir la lógica del negocio implementada con el SGBDOO objectDB por otra realizada usando el framework Hibernate.

Pasos a seguir

1. Añadir en el proyecto JSF de casas rurales el código fuente del proyecto inicial

En el nuevo proyecto con JSF e Hibernate no se necesitará utilizar el wsRuralHouse.jar correspondiente al proyecto anterior, el cual era necesario para crear la instancia con la lógica del negocio (new FacadeImplementationWS()).

Lo que hay que hacer es incluir en este proyecto de JSF y ahora Hibernate, las clases del proyecto anterior wsRuralHouse-OCW, para que a partir de ellas se pueda modificar lo necesario.

Algunas de estas clases no son necesarias como BusinessLogicServer, BusinessLogicServerConsole, ConfigXML y ObjectdbManagerServer, así como todas las del paquete “gui”, por lo que se pueden eliminar.

2. Implementar la nueva lógica del negocio con Hibernate

Hay que construir una nueva implementación de FacadeImplementationWS que implemente los mismos métodos (que son los definidos en la interfaz ApplicationFacadeInterfaceWS) utilizando Hibernate.

Pero, en realidad, lo único que hay que cambiar en la clase FacadeImplementationWS, es la clase de acceso a datos que utiliza, que en vez de DataAccess (que utiliza una base de datos orientada a objetos objectDB) sea otra clase de acceso a datos que utilice Hibernate.

Aunque se podría no modificar FacadeImplementationWS sino rehacer la clase DataAccess, se creará una nueva clase de acceso a datos, HibernateDataAccess, y las referencias a DataAccess en FacadeImplementationWS se sustituirán por HibernateDataAccess.

De las clases del dominio (Offer y RuralHouse) habría que quitar las anotaciones de objectDB, que son las siguientes:

```
@Entity  
@Id  
@GeneratedValue
```

Al definir las clases del dominio hay que tener en cuenta que Hibernate no funciona bien dando persistencia a atributos o propiedades que son de tipo Vector. En vez de Vector hay que usar colecciones Java (como Set, List,...). Ver: <https://forum.hibernate.org/viewtopic.php?f=6&t=932647>

Por lo tanto, lo que hay que hacer es: **crear los ficheros de enlace o mapping, el fichero de configuración de Hibernate y la implementación de la clase DAO (HibernateDataAccess) usando cuando sea necesario sentencias HQL o instrucciones de inserción, borrado, actualización o recuperación de objetos dado su identificador.**

