

Materia: Diseño de Sistemas

Código de Materia: 082028

Curso: K-3054

Docentes:

Acevedo, Fabiana
Böse, Santiago
Mendez, Bruno
Mur, Pablo
Procopio, Demian
Rico Mendoza, René
Sosa, Ezequiel

Trabajo Práctico: “¿Qué comemos?”

Tipo: Grupal / Individual

GRUPO N° 05	
NOMBRE Y APELLIDO	LEGAJO N°
G Leonel, Gutierrez	137731-0
Matias Aguirre (ex Grupo 03)	121428-7
	-
	-
	-

Fecha prevista de entrega: 22/ 11/ 2015

Fecha real de entrega: 05/ 02/ 2016

Calificación..... Firma.....

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Historia de revisión

Fecha	Descripción	Autor	Versión
30/07/2015	Entrega 3	Leonel Gutierrez	1.0
10/08/2015	Entrega 3 Correcciones	Leonel Gutierrez	1.1
27/08/2015	Entrega 4	Leonel Gutierrez	2.0
28/10/2015	Entrega 5	Matias Aguirre	3.0
05/02/2016	Entrega 6	Grupo 5	4.0

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Tabla de Contenidos

Patrones de Diseño.....	4
Unión con Grupo 03.....	5
Arquitectura Web.....	6
Diagrama de Objetos.....	7
Diagrama de Dispersión.....	8
Mapa del sitio.....	9
Diagrama de Clases.....	10
Diagrama de Entidad Relación.....	11
Diagrama de Secuencias.....	12
MODIFICAR RECETA.....	12
Documentación.....	13
Anexo de Pruebas.....	14
Anexo de Pruebas Entrega 6.....	16

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Patrones de Diseño

Patrón de Diseño Prototype: Se lo utilizó en la función modificarReceta de la clase Usuario para hacer un clon de la receta, y así conservar la receta original.

Patrón de Diseño Template Método: Para el manejo de usuario común y usuario administrador, se cambió la clase Usuario por una Abstrac class y se crearon las dos clases UsuarioComun y UsuarioAdministrador que extienden de la clase Usuario y así estas clases pueden especializarse.

Patrón de Diseño Strategy: Para los filtros de los Reportes, se usó este patrón donde se implementó la interface FiltroStrategy y cada filtro reescribe la función principal Filtrar. También se usó este patrón para las Recomendaciones.

Como parte del 'merge' con entregas previas del Grupo 03, tanto RepositorioUsuarios como RepositorioRecetas y RepositorioGrupos son objetos Singleton (patrón Singleton), ya que su instancia debe ser única y no tiene sentido propagar un puntero entre todas las clases que deberían conocer dicha instancia.

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Unión con Grupo 03

Como se mencionó en el apartado anterior, con la unión de los grupos 03 y 05 se realizó un "merge" entre ambas soluciones. Si bien el grupo 03 iba retrasado con respecto al 05 que se tomó como base, algunas partes de este último fueron cambiadas por considerarse mejor resueltas en la versión del grupo 03:

Repositorios:

Todos los repositorios (Grupos, Usuarios y Recetas), fueron convertidos en singleton por considerarse que su instancia debe ser única, y el hecho de tener que propagar un puntero a dicha instancia aumentaría inapropiadamente el acoplamiento entre objetos.

Para la entrega final se planea que los repositorios (que son básicamente colecciones de Grupos, Usuarios y Recetas con algún otro comportamiento), sean los que se encarguen de la persistencia. Por esto mismo los repositorios no fueron representados con ninguna entidad en la Base de Datos ni persistidos.

Ingredientes

Los ingredientes dejaron de ser elementos de un Enum para convertirse en clases. De esta forma para agregar nuevos ingredientes solo será necesario crear una instancia sin necesidad de modificar y recompilar el código.

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Arquitectura Web

Tecnología de Presentación

La tecnología de presentación utilizada es AngularJS.

El motivo de esta elección fue el tener experiencia previa con este framework.

La otra tecnología de presentación utilizada fue el framework Bootstrap, con el cual nos permiten tener un sitio web “Responsive” (disponible para todo tipo de pantallas).

API Rest

La API Rest seleccionada es el framework Jersey que para su funcionamiento requiere algunos frameworks secundarios adicionales tales como log4j, json, gson, etc.

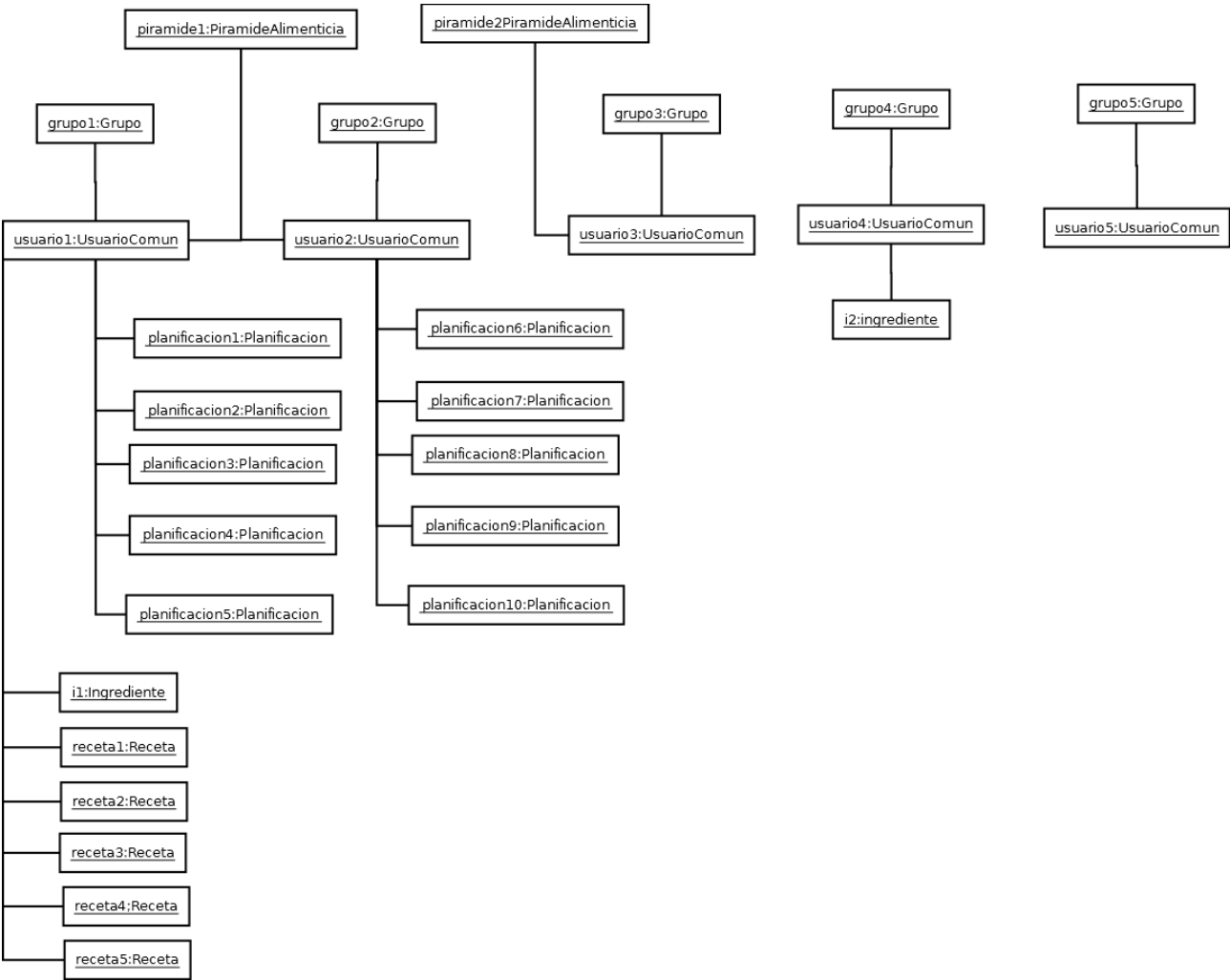
Controllers

loginCtrl
principalCtrl
RecetasCtrl
usuarioCtrl
gruposCtrl
planificarCtrl
EstadisticasCtrl
reportesCtrl
recomendacionCtrl

Recursos

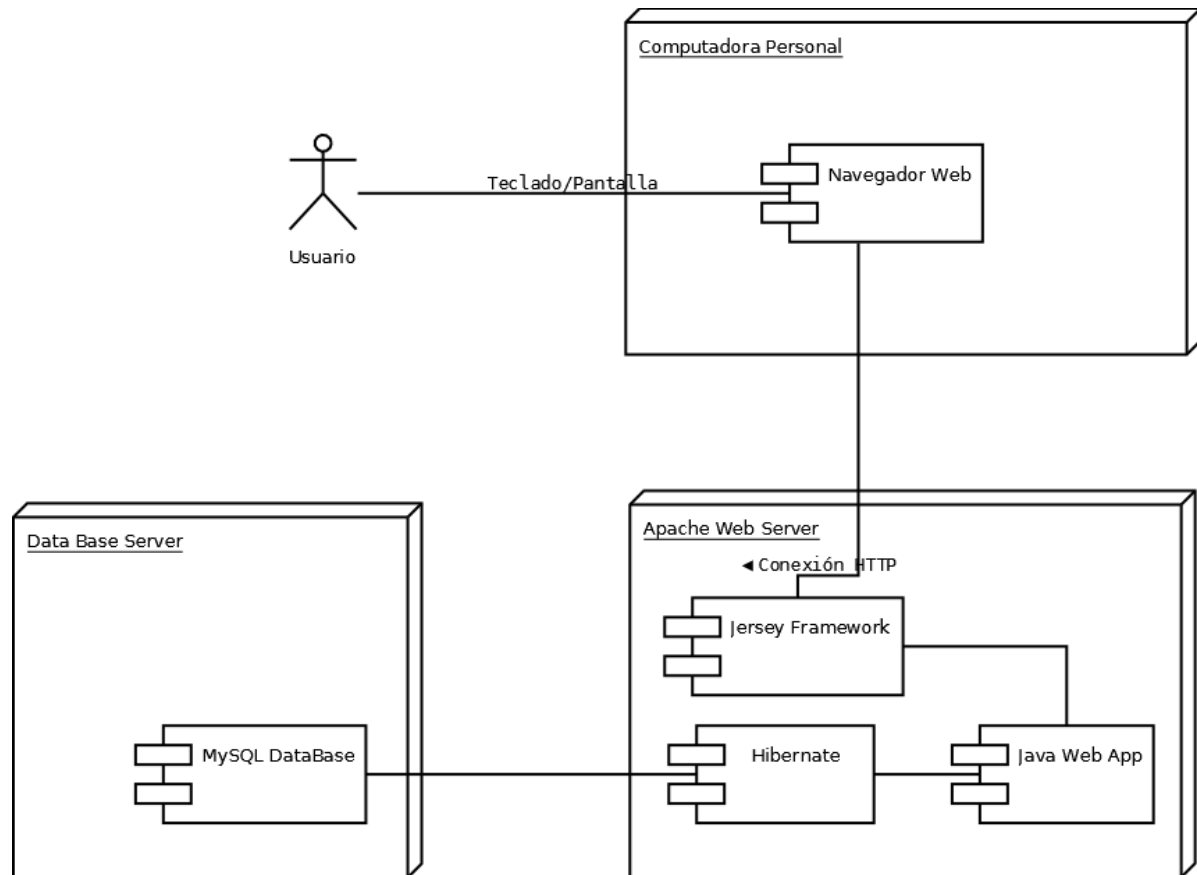
Calificaciones
Estadísticas
Grupos
Login
Planificaciones
Recetas
Recomendaciones
Reporte
Usuarios

Diagrama de Objetos



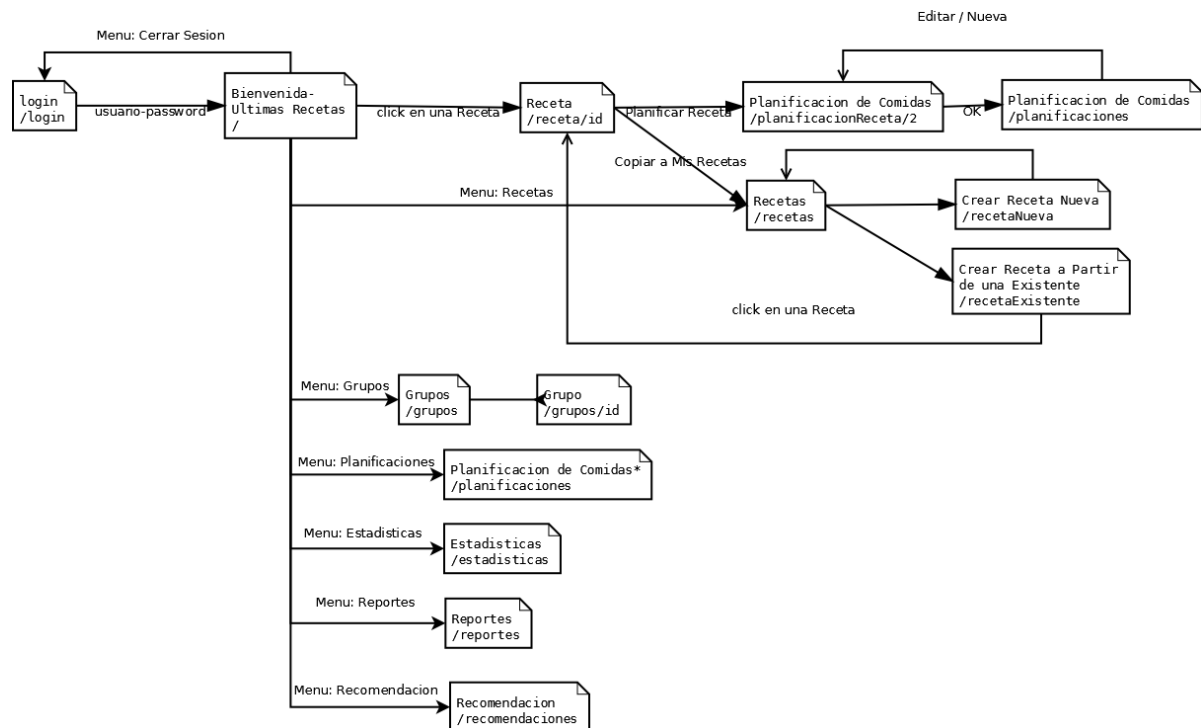
Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Diagrama de Dispersión



Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Mapa del sitio



Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Diagrama de Clases

El Diagrama de Clases se entrega adjunto en formato PNG porque su tamaño impide una buena apreciación.

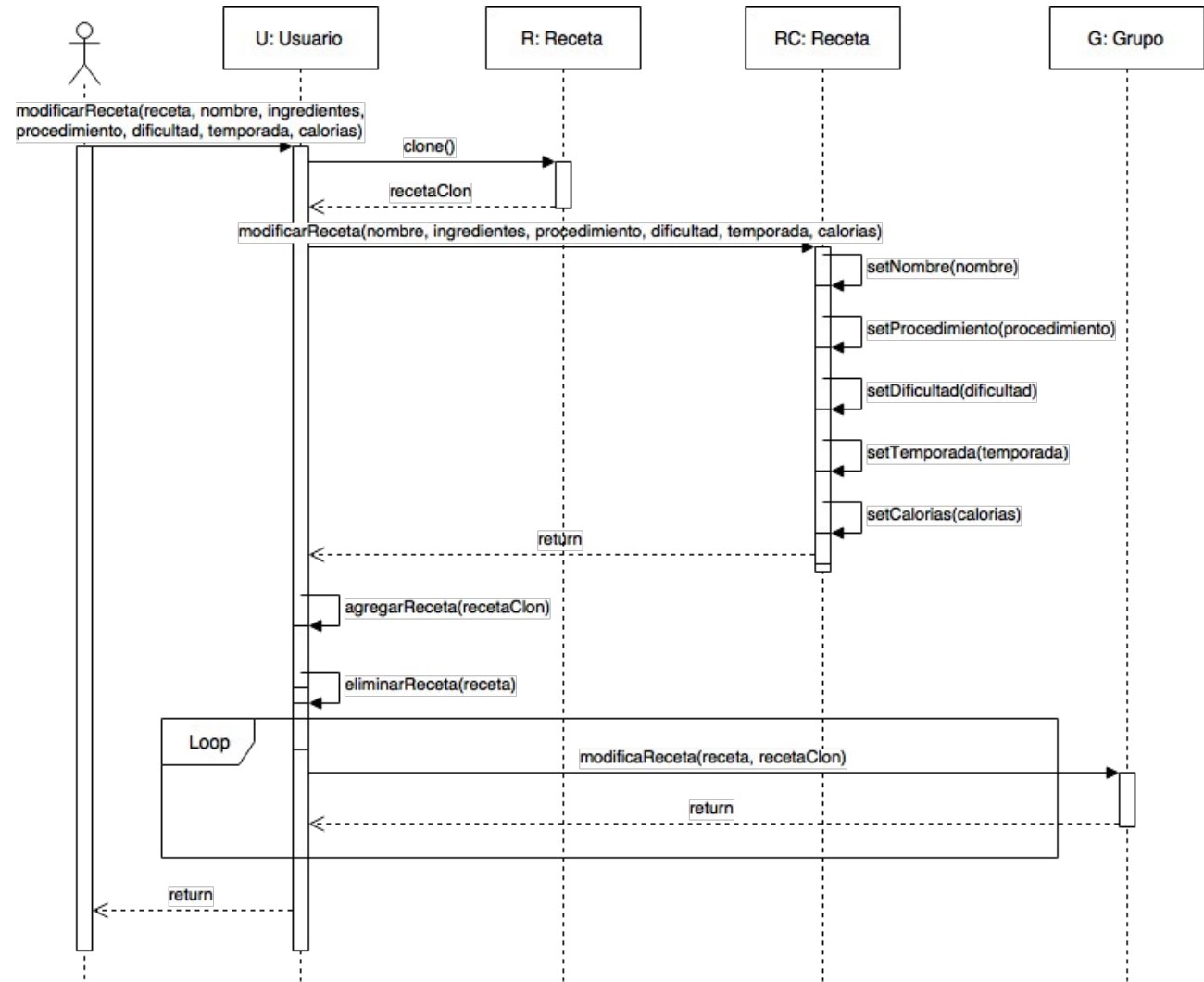
Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Diagrama de Entidad Relación

El Diagrama de Entidad Relación se entrega adjunto en formato PNG porque su tamaño impide una buena apreciación.

Diagrama de Secuencias

Modificar Receta



Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Documentación

En el diagrama de clases principal solamente se representaron las clases principales y mas relevantes, quedando afuera algunas clases secundarias que no tienen mayor impacto en la estructura de la solución.

Tal como en entregas anteriores y como se acostumbra, los métodos setters y getters, los métodos redefinidos de superclases de la librería de Java y algunos constructores no fueron representados.

En la entrega 2, la validación de datos de una receta pide que tenga el atributo: Dificultad de la receta (lista de valores: 1 a 5 estrellas), y en el archivo de Contexto del tp, para la dificultad utiliza otro criterio: Dificultad Receta: Fácil, Media, Difícil, Muy difícil. Para resolver este conflicto se utilizó el segundo criterio, y la lista de valores de 1 a 5 se valida en la calificación de la receta.

En la entrega 4, el criterio que se usó en las Recomendaciones por Balanceo es un sistema de puntos, donde por cada ingrediente (tipo de alimento) que se encuentre en la pirámide, suma un punto, y si las planificaciones ya cubrieron la cantidad estipulada por la pirámide, no suma puntos.

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Anexo de Pruebas

Estadísticas: Para las estadísticas, se hizo un test por semana, uno por mes y otro por ranking de recetas más copiadas. Además diferencia éstas, de acuerdo a qué usuario la ejecuta (ej: un usuario administrador puede ejecutar las estadísticas, mientras que un usuario normal no).

1. TestEstadisticasPorEstacionYSemana
2. TestEstadisticasPorEstacionYMes
3. TestEstadisticasRankingRecetasMasCopiadas

Reportes: Para los tests de reportes, se hizo un test por cada filtro para verificar que funcione correctamente. Además se hizo un test adicional donde se incluyen dos filtros juntos.

1. TestReportePorPeriodo
2. TestReportePorIngredientes
3. TestReportePorCalorias
4. TestReportePorNombre
5. TestReportePorGrupo
6. TestReportePorUsuario
7. TestReporteVariosFiltros
8. TestReporteRecetasNuevas

Recomendaciones: Para las recomendaciones se hizo un test por cada recomendación, para más detalle, se comentaron en los tests el funcionamiento de cada uno.

1. TestPorDiaHorario
2. TestPorBalanceo

Persistencia: Se desarrollaron los siguientes tests para la persistencia en la DB:

- testPersistenciaUsuario
- testPersistenciaGrupo
- testPersistenciaGeneracionReceta
- testPersistenciaRestriccion
- testComparteReceta
- testUsuarioSeUneAGrupo
- testUsuarioSeBorraDeGrupo
- testCalificaReceta
- testPlanificaComida
- testReplanificaComida

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

- testBorraReceta
- testObtenerRankingRecetas

Todos los test pueden ser ejecutados conjunta o individualmente.

Diseño de Sistemas	Curso: K-3054 – Año 2015
Trabajo Práctico: ¿Qué comemos?	Grupo: 05 – Versión 4.0

Anexo de Pruebas Entrega 6

Para poder probar la aplicación principal, debe existir un esquema en la BD llamado 'dds'.

Luego se cargan los datos desde el archivo CargaDatos:

src/main/java/ar.edu.utn.d2s.cargadatos/CargaDatos.java (Run as - Java Application)

Luego se modifica el archivo de Hibernate:

src/main/resources/hibernate.cfg.xml

La línea (n.º 14):

```
<property name="hbm2ddl.auto">create-drop</property>
```

Se la reemplaza por:

```
<property name="hbm2ddl.auto">update</property>
```

Con esto evitamos que se vuelva a crear las tablas y se puedan trabajar los datos ingresados.

Luego se corre la aplicación principal en el servidor de Java (Run as - Run on Server).

Si se quiere volver a hacer las pruebas de JUNIT, hay que volver a modificar el archivo de Hibernate y dejarlo como estaba.