

Boletín 3. Creación de la capa DAO

BDSW Tema 4 –La capa de acceso a los datos (DAO)

12 de febrero de 2014

Índice

1	Introducción	1
2	Creación del proyecto Eclipse	1
2.1	Los DTOs	1
3	La interface <code>EmpleadoDAO</code>	2
4	La factoría abstracta <code>DAOFactory</code>	2
5	La factoría <code>MySQLDAOFactory</code>	2
6	La clase <code>MySQLEmpleadoDAO</code>	3
7	Actividades	4
7.1	Actividad 1	4
7.2	Actividad 2	5
7.3	Actividad 3	5

Índice de figuras

1	Panel de creación de la clase <code>MySQLDAOFactory</code>	3
2	Panel de creación de la clase <code>MySQLEmpleadoDAO</code>	4

Índice de listados

1	Código de la interface <code>EmpleadoDAO</code>	2
2	Código de la clase <code>DAOFactory</code>	2
3	Código de la clase <code>MySQLDAOFactory</code>	2
4	Código parcial de la clase <code>MySQLEmpleadoDAO</code>	3

1 Introducción

Siguiendo con el ejemplo de la actividad empresarial simple y teniendo en cuenta que:

- En el Tema 1 se introdujeron los VOs (DTOs, DTs) `Empleado` y `Departamento`.
- En el Tema 2 de creo la base de datos `Empresa`.

El objetivo del siguiente ejercicio es implementar el DAO de la clase `Empleado` utilizando el driver JDBC de MySQL para las operaciones básicas CRUD de la base de datos.


2 Creación del proyecto Eclipse

Para la realización de este boletín, crearemos un nuevo proyecto Eclipse abriendo el menú **File** y seleccionando **New** → **Java Project**. En el panel emergente, creamos el proyecto `DAO`. Seleccionamos con el botón derecho el icono del proyecto y en el menú contextual seleccionamos **New** → **Package** y creamos el paquete `sssi.tasi.capitulo4.dto`.

2.1 Los DTOs

Desempaquete el archivo `boletin_t4b3_ficheros.zip` y mueva los ficheros `*.java` (que implementan las clases `Empleado` y `Departamento`) al directorio `DAO/src/sssi/tasi/capitulo4/dto` bajo su espacio de trabajo (*workspace*). Recuerde refrescar (*Refresh*) el catálogo de Eclipse para que tenga en cuenta los nuevos ficheros. Para ello puede seleccionar la entrada correspondiente bajo el menú **File** o pulsar la tecla **F5**.

3 La interface EmpleadoDAO

Para crear la interface EmpleadoDAO entraremos en eclipse. En el proyecto TASI crearemos un nuevo paquete haciendo click sobre el icono . En el menú contextual seleccionar **New** → **Package** y crear el paquete `sssi.tasi.capitulo4.dao`.

Seleccionamos ahora este paquete con el botón derecho y en el menú contextual seleccionamos **New** → **Interface** para crear la interface EmpleadoDAO cuyo código se muestra en el listado 1

Listado 1: Código de la interface EmpleadoDAO.

```
1 package sssi.tasi.capitulo4.dao;
2
3 import sssi.tasi.capitulo4.dto.Empleado;
4
5 public interface EmpleadoDAO {
6     public void insert(Empleado e);
7     public Empleado read(Empleado e);
8     public void update(Empleado e);
9     public void delete(Empleado e);
10 }
```

4 La factoría abstracta DAOFactory

La factoría abstracta DAOFactory se comporta como la fábrica genérica de DAO para nuestro programa. El código de la clase se muestra parcialmente en el listado 2

Listado 2: Código de la clase DAOFactory.

```
1 package sssi.tasi.capitulo4.dao;
2
3 import sssi.tasi.capitulo4.dto.Empleado;
4
5 public abstract class DAOFactory {
6     // Lista de DAOs soportados
7     public static final int ORACLE = 1;
8     public static final int MYSQL = 2;
9     public static final int XML = 3;
10
11     // Metodos para cada DAO creado
12     public abstract EmpleadoDAO getEmpleadoDAO();
13
14     public static DAOFactory getDAOFactory(int which) {
15         switch (which) {
16             case MYSQL:
17                 return new MySQLDAOFactory();
18             default:
19                 return null;
20         }
21     }
22 }
```

Aviso: No se preocupe si Eclipse muestra un error indicando que no puede resolver el nombre MySQLDAOFactory. Ese mensaje desaparecerá tan pronto creamos la clase MySQLDAOFactory en el siguiente apartado.

5 La factoría MySQLDAOFactory

Creamos la clase MySQLDAOFactory seleccionando el ratón con el botón derecho y eligiendo **New** → **class** en el menú contextual. Cumplimentar el panel de creación de la clase como se muestra en la figura 1

El código de la clase se muestra parcialmente en el listado 3.

Listado 3: Código de la clase MySQLDAOFactory.

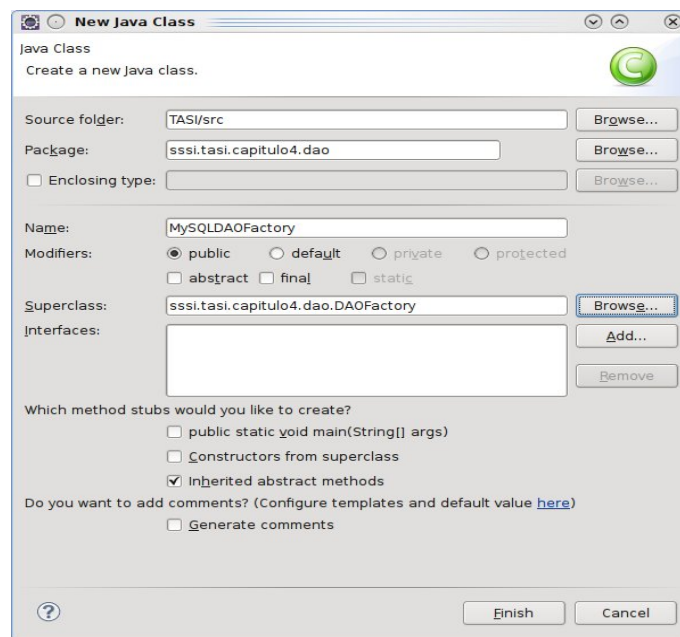


Figura 1: Panel de creación de la clase MySQLDAOFactory.

```

1 package sssi.tasi.capitulo4.dao;
2
3 import java.text.SimpleDateFormat;
4 import java.util.*;
5 import java.sql.*;
6
7 public class MySQLDAOFactory extends DAOFactory {
8     public static final String DRIVER="com.mysql.jdbc.Driver";
9     public static final String DBURL="jdbc:mysql://localhost/empresa";
10    public static final String USERNAME="usuario";
11    public static final String PASSWORD="usupw";
12
13    public static Connection createConnection() throws Exception {
14        Class.forName(DRIVER).newInstance();
15        return DriverManager.getConnection(DBURL, USERNAME, PASSWORD);
16    }
17
18    @Override
19    public EmpleadoDAO getEmpleadoDAO() {
20        return new MySQLEmpleadoDAO();
21    }

```

Aviso: No se preocupe si Eclipse muestra un error indicando que no puede resolver el nombre MySQLEmpleadoDAO. Ese mensaje desaparecerá tan pronto como creemos esta clase en el siguiente apartado.

6 La clase MySQLEmpleadoDAO

Creamos la clase MySQLEmpleadoDAO seleccionando el ratón con el botón derecho y eligiendo **New** → **class** en el menú contextual. Cumplimentar el panel de creación de la clase como se muestra en la figura 2.

El código parcial de esta clase se muestra en el listado 4.

Listado 4: Código parcial de la clase MySQLEmpleadoDAO.

```

1 package sssi.tasi.capitulo4.dao;
2
3 import sssi.tasi.capitulo4.dto.Empleado;
4
5 public class MySQLEmpleadoDAO implements EmpleadoDAO {

```

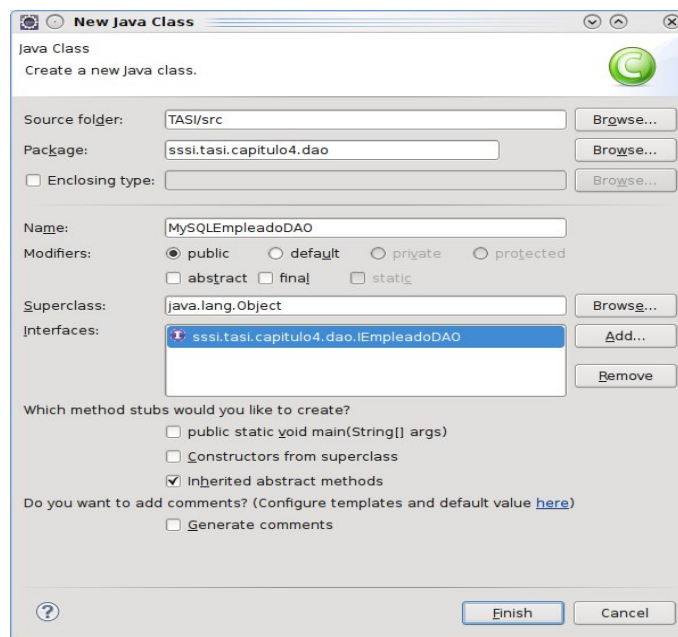


Figura 2: Panel de creación de la clase MySQLEmpleadoDAO.

```

6
7  @Override
8  public void delete(Empleado e) {
9      // TODO Auto-generated method stub
10
11  }
12
13  @Override
14  public void insert(Empleado e) {
15      // TODO Auto-generated method stub
16
17  }
18
19  @Override
20  public Empleado read(Empleado e) {
21      // TODO Auto-generated method stub
22      return null;
23  }
24
25  @Override
26  public void update(Empleado e) {
27      // TODO Auto-generated method stub
28
29  }
30  }

```

7 Actividades

7.1 Actividad 1

Complete la clase EmpleadoDAO. Pruebe el funcionamiento de la clase DAO construyendo un pequeño banco de pruebas en Java que:

- Lea (read) y escriba en la consola los empleados de código '7', '24', '33' y '51'
- Cree dos usuarios con su nombre y el nombre de un compañero. Léalos y muéstrelos por pantalla. Deje el atributo departamento puesto a NULL.
- Modifique (update) los empleados creados y los asigne al departamento '2' (Planificación).
- Borre los usuarios creados dejando la base de datos en el mismo estado en que la encontró.

7.2 Actividad 2

Cree el DAO para la clase `Departamento`. Para ello deberá crear la interface `DepartamentoDAO`, crear la clase `MySQLEmpleadoDAO` que implemente la interface y ampliar las clase `DAOFactory` y `MySQLDaoFactory` adecuadamente. Elabore un pequeño banco de pruebas de modo similar al descrito en la actividad anterior y que permita validar todas las funciones del DAO.

7.3 Actividad 3

Construya el DAO para la clase `Proyecto` siguiendo el mismo procedimiento descrito hasta ahora. Deberá generar la clase DTO `Proyecto` en el paquete `sssi.tasi.capitulo4.dto`. Elabore un pequeño banco de pruebas de modo similar al descrito en la actividad anterior y que permita validar todas las funciones del DAO.