Ejercicio No. 5. Categoria-Producto Hibernate

Barrera Pérez Carlos Tonatihu
Profesor: José Asunción Enríquez Zárate
Web Application Development
Grupo: 3CM9

3 de junio de 2019

Índice

1. Introducción **3**

2. Desarrollo **4**

3. Pruebas **23**

4. Conclusiones **29**

1. Introducción

Este ejercicio tuvo como objetivo desarrollar un proyecto web para el CRUD de tiendita que se había trabajado con anterioridad, la base que se trabajo fue 1.

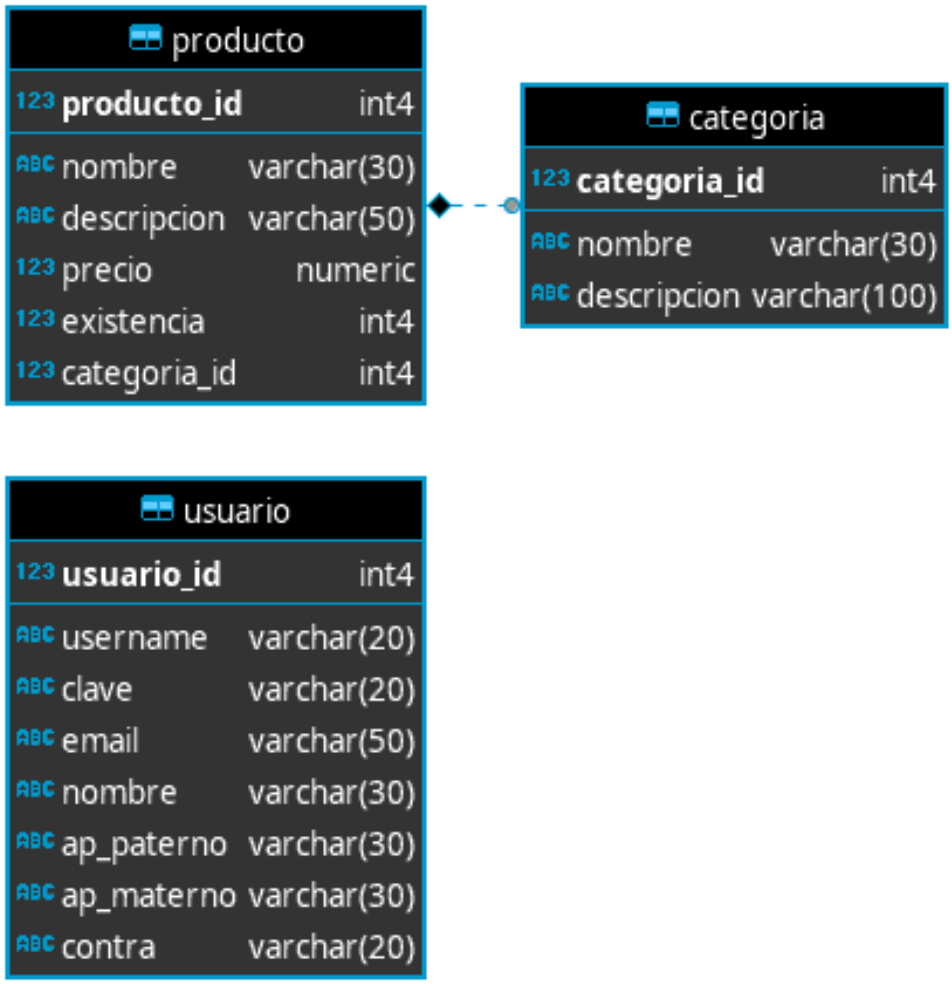

Figura 1: Base de datos que se trabajo en PostgreSQL

También se agrego la funcionalidad de inicio de sesión, generación de reporte, de gráficas y envió de estos dos archivos por correo.

2. Desarrollo

```java
package me.tonatihu.util;

import org.hibernate.HibernateException;
import org.hibernate.Metamodel;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.query.Query;

import javax.persistence.metamodel.EntityType;
import java.util.logging.Level;
import java.util.logging.Logger;

public class HibernateUtil {
    private static final SessionFactory ourSessionFactory;

    static {
        try {
            Configuration configuration = new Configuration();
            configuration.configure();

            ourSessionFactory = configuration.buildSessionFactory();
        } catch (Throwable ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static Session getSession() throws HibernateException {
        return ourSessionFactory.openSession();
    }
}
```

Archivo 1: HibernateUtil.java

```java
package me.tonatihu.controller;

```

```
3 import me.tonatihu.dao.UsuarioDao;
4 import me.tonatihu.dao.impl.UsuarioDaoImpl;
5 import me.tonatihu.dto.Usuario;
6 import me.tonatihu.entity.UsuarioEntity;
7
8 import javax.servlet.ServletException;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import javax.servlet.http.HttpSession;
13 import java.io.IOException;
14 import java.util.logging.Level;
15 import java.util.logging.Logger;
16
17 //@WebServlet(name = "LoginServlet")
18 public class LoginServlet extends HttpServlet {
19     private static final Logger LOGGER =
20 Logger.getLogger(LoginServlet.class.getName());
21     protected void doPost(HttpServletRequest request,
22             HttpServletResponse response) throws
    ServletException, IOException {
23         processRequest(request, response);
24     }
25
26     protected void doGet(HttpServletRequest request,
27             HttpServletResponse response) throws
    ServletException, IOException {
28             processRequest(request, response);
29     }
30
31     private void processRequest(HttpServletRequest request,
    HttpServletResponse
32 response)
33             throws ServletException, IOException {
34         request.setCharacterEncoding("UTF-8");
35         String accion = request.getParameter("logout");
36         if (accion != null)
37             logout(request, response);
38         else {
39             if (request.getMethod().equals("POST"))
40                 login(request, response);
41             else
42                 request.getRequestDispatcher("login.jsp").
```

```
    forward(request,
43 response);
44          }
45      }
46
47      private void login(HttpServletRequest request,
    HttpServletResponse response)
48              throws IOException, ServletException {
49          UsuarioDao dao = new UsuarioDaoImpl();
50          String username = request.getParameter("username");
51          String contra = request.getParameter("password");
52          UsuarioEntity u = (UsuarioEntity) dao.
    findByUsernameAndContra(username,
53 contra);
54          if (u != null) {
55              HttpSession session = request.getSession();
56              Usuario usuario = new Usuario();
57              usuario.setUsername(u.getUsername());
58              usuario.setNombreCompleto(u.getNombre() + " " +
     u.getApPaterno() + "
59 " + u.getApMaterno());
60              session.setAttribute("USUARIO_SESSION", usuario
    );
61              response.sendRedirect("home");
62          } else {
63              LOGGER.log(Level.WARNING, "You shall not pass!"
    );
64              request.getRequestDispatcher("login.jsp").
    forward(request,
65 response);
66          }
67      }
68
69      private void logout(HttpServletRequest request,
    HttpServletResponse
70 response)
71              throws ServletException, IOException {
72          HttpSession session = request.getSession(false);
73          session.invalidate();
74          request.getRequestDispatcher("login.jsp").forward(
    request, response);
75      }
76 }
```

Archivo 2: LoginServlet.java

```java
package me.tonatihu.controller;

import me.tonatihu.dao.impl.UsuarioDaoImpl;
import me.tonatihu.dto.Usuario;
import me.tonatihu.entity.UsuarioEntity;
import me.tonatihu.util.Paginas;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;

//@WebServlet(name = "PerfilServlet")
public class PerfilServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws
    ServletException, IOException {
        processRequest(request, response);
    }

    protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws
    ServletException, IOException {
        processRequest(request, response);
    }

    private void processRequest(HttpServletRequest request,
     HttpServletResponse
response)
            throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        HttpSession session = request.getSession(false);
        Usuario u = (Usuario) session.getAttribute("
    USUARIO_SESSION");
        UsuarioDaoImpl usuarioDao = new UsuarioDaoImpl();
        UsuarioEntity usuario = usuarioDao.findById(u.
    getUsername());
```

```
35          request.setAttribute("usuario", usuario);
36          request.setAttribute("PAGINA", Paginas.PERFIL);
37          request.getRequestDispatcher("perfil.jsp").forward(
    request, response);
38      }
39  }
```

Archivo 3: PerfilServlet.java

```
1  package me.tonatihu.controller;
2
3  import me.tonatihu.entity.CategoriaEntity;
4  import me.tonatihu.service.CategoriaService;
5  import me.tonatihu.util.Paginas;
6
7  import javax.servlet.ServletException;
8  import javax.servlet.annotation.WebServlet;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import java.io.IOException;
13 import java.util.List;
14
15 //@WebServlet(name = "CategoriasServlet")
16 public class CategoriasServlet extends HttpServlet {
17     private void processRequest(HttpServletRequest request,
18             HttpServletResponse response) throws
    ServletException, IOException {
19         request.setCharacterEncoding("UTF-8");
20         String accion = request.getParameter("accion");
21         switch (accion) {
22             case "ver":
23                 listarCategorias(request, response);
24                 break;
25             case "agregar":
26                 crearCategoria(request, response);
27                 break;
28             case "eliminar":
29                 eliminarCategoria(request, response);
30                 break;
31             case "guardar":
32                 almacenarCategoria(request, response);
33                 break;
```

```
34              case "editar":
35                  editarCategoria(request, response);
36                  break;
37          }
38      }
39
40      protected void doPost(HttpServletRequest request,
41              HttpServletResponse response) throws
    ServletException, IOException {
42          processRequest(request, response);
43      }
44
45      protected void doGet(HttpServletRequest request,
46              HttpServletResponse response) throws
    ServletException, IOException {
47          processRequest(request, response);
48      }
49
50      private void editarCategoria(HttpServletRequest request
    ,
51              HttpServletResponse response) throws
    ServletException, IOException {
52          CategoriaService service = new CategoriaService();
53          CategoriaEntity c =
54 service.findById(Integer.parseInt(request.getParameter("id"
    )));
55          request.setAttribute("categoria", c);
56          request.setAttribute("PAGINA", Paginas.
    VER_CATEGORIAS);
57          request.getRequestDispatcher("formCategoria.jsp").
    forward(request,
58 response);
59      }
60
61      private void listarCategorias(HttpServletRequest
    request,
62 HttpServletResponse response)
63              throws ServletException, IOException {
64          CategoriaService categoriaService = new
    CategoriaService();
65          List<CategoriaEntity> lista = categoriaService.
    findAll();
66          request.setAttribute("categorias", lista);
```

```
67          request.setAttribute("PAGINA", Paginas.
    VER_CATEGORIAS);
68          request.getRequestDispatcher("listaCategorias.jsp")
    .forward(request,
69 response);
70      }
71
72      private void crearCategoria(HttpServletRequest request,
73              HttpServletResponse response) throws
    ServletException, IOException {
74          request.setAttribute("PAGINA", Paginas.
    AGREGAR_CATEGORIA);
75          request.getRequestDispatcher("formCategoria.jsp").
    forward(request,
76 response);
77      }
78
79      private void eliminarCategoria(HttpServletRequest
    request,
80 HttpServletResponse response)
81              throws IOException {
82          CategoriaService categoriaService = new
    CategoriaService();
83          int id = Integer.parseInt(request.getParameter("id"
    ));
84          categoriaService.delete(id);
85          response.sendRedirect("categorias?accion=ver");
86      }
87
88      private void almacenarCategoria(HttpServletRequest
    request,
89 HttpServletResponse response)
90              throws IOException {
91          CategoriaEntity c = new CategoriaEntity();
92          CategoriaService service = new CategoriaService();
93          if (request.getParameter("id") == null ||
94 request.getParameter("id").isEmpty()) {
95              c.setNombre(request.getParameter("nombre"));
96              c.setDescripcion(request.getParameter("
    descripcion"));
97              service.create(c);
98          } else {
99              c.setCategoriaId(Integer.parseInt(request.
```

```
        getParameter("id")));
100             c.setNombre(request.getParameter("nombre"));
101             c.setDescripcion(request.getParameter("
    descripcion"));
102             service.update(c);
103         }
104         response.sendRedirect("categorias?accion=ver");
105     }
106 }
```

Archivo 4: CategoriasServlet.java

```
 1 package me.tonatihu.controller;
 2
 3 import me.tonatihu.dao.impl.ProductoDaoImpl;
 4 import me.tonatihu.entity.CategoriaEntity;
 5 import me.tonatihu.entity.ProductoEntity;
 6 import me.tonatihu.service.CategoriaService;
 7 import me.tonatihu.util.Paginas;
 8
 9 import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import java.io.IOException;
14 import java.math.BigDecimal;
15 import java.util.List;
16
17 //@WebServlet(name = "ProductosServlet")
18 public class ProductosServlet extends HttpServlet {
19     protected void doPost(HttpServletRequest request,
20             HttpServletResponse response) throws
    ServletException, IOException {
21         processRequest(request, response);
22     }
23
24     protected void doGet(HttpServletRequest request,
25             HttpServletResponse response) throws
    ServletException, IOException {
26         processRequest(request, response);
27     }
28
29     private void processRequest(HttpServletRequest request,
```

```
      HttpServletResponse
30 response)
31             throws ServletException, IOException {
32         request.setCharacterEncoding("UTF-8");
33         String accion = request.getParameter("accion");
34         switch (accion) {
35             case "ver":
36                 listar(request, response);
37                 break;
38             case "agregar":
39                 crear(request, response);
40                 break;
41             case "eliminar":
42                 eliminar(request, response);
43                 break;
44             case "guardar":
45                 guardar(request, response);
46                 break;
47             case "editar":
48                 editar(request, response);
49                 break;
50         }
51     }
52
53     private void crear(HttpServletRequest request,
     HttpServletResponse response)
54 throws ServletException, IOException {
55         CategoriaService service = new CategoriaService();
56         request.setAttribute("PAGINA", Paginas.
     AGREGAR_PRODUCTO);
57         request.setAttribute("categorias", service.findAll
     ());
58         request.getRequestDispatcher("formProducto.jsp").
     forward(request,
59 response);
60     }
61
62     private void guardar(HttpServletRequest request,
     HttpServletResponse
63 response)
64             throws IOException {
65         ProductoEntity producto = new ProductoEntity();
66         ProductoDaoImpl productoDao = new ProductoDaoImpl()
```

```
    ;
67          CategoriaEntity c = new CategoriaEntity();
68
69          c.setCategoriaId(Integer.parseInt(request.
    getParameter("categoria")));
70
71          producto.setNombre(request.getParameter("nombre"));
72          producto.setDescripcion(request.getParameter("
    descripcion"));
73
74 producto.setExistencia(Integer.parseInt(request.
    getParameter("existencia")));
75
76 producto.setPrecio(BigDecimal.valueOf(Double.valueOf(
    request.getParameter("preci
77 o"))));
78          producto.setCategoria(c);
79          if (request.getParameter("id") == null ||
80 request.getParameter("id").isEmpty()) {
81              productoDao.create(producto);
82          } else {
83
84 producto.setProductoId(Integer.parseInt(request.
    getParameter("id")));
85              productoDao.update(producto);
86          }
87          response.sendRedirect("productos?accion=ver");
88      }
89
90      private void eliminar(HttpServletRequest request,
    HttpServletResponse
91 response)
92              throws IOException {
93          ProductoDaoImpl dao = new ProductoDaoImpl();
94          int id = Integer.parseInt(request.getParameter("id"
    ));
95          ProductoEntity p = dao.findById(id);
96          dao.delete(p);
97          response.sendRedirect("productos?accion=ver");
98      }
99
100     private void editar(HttpServletRequest request,
    HttpServletResponse
```

```
101 response)
102             throws ServletException, IOException {
103         ProductoDaoImpl dao = new ProductoDaoImpl();
104         CategoriaService service = new CategoriaService();
105         ProductoEntity p=
106 dao.findById(Integer.parseInt(request.getParameter("id")));
107         request.setAttribute("producto", p);
108         request.setAttribute("categorias", service.findAll
    ());
109         request.setAttribute("PAGINA", Paginas.
    VER_PRODUCTOS);
110         request.getRequestDispatcher("formProducto.jsp").
    forward(request,
111 response);
112     }
113
114     private void listar(HttpServletRequest request,
    HttpServletResponse
115 response)
116             throws ServletException, IOException {
117         request.setAttribute("PAGINA", Paginas.
    VER_PRODUCTOS);
118         ProductoDaoImpl dao = new ProductoDaoImpl();
119         List<ProductoEntity> lista = ((ProductoDaoImpl) dao
    ).findAll();
120         request.setAttribute("productos", lista);
121         request.getRequestDispatcher("listaProductos.jsp").
    forward(request,
122 response);
123     }
124
125 }
```

Archivo 5: ProductosServlet.java

```
1 package me.tonatihu.dao;
2
3 import me.tonatihu.util.HibernateUtil;
4 import org.hibernate.Session;
5 import org.hibernate.Transaction;
6
7 import java.io.Serializable;
8 import java.util.List;
```

```
 9
10 /**
11  * @author tonatihu
12  * Created on 3/17/19
13  */
14
15 public abstract class GenericDao <T, Id extends
       Serializable>{
16     private Session currentSession;
17     private Transaction currentTransaction;
18
19     public void openCurrentSession() {
20         currentSession = HibernateUtil.getSession();
21     }
22
23     public void openCurrentSessionWithTransaction() {
24         currentSession = HibernateUtil.getSession();
25         currentTransaction = currentSession.
       beginTransaction();
26     }
27
28     public void closeCurrentSession() {
29         currentSession.close();
30     }
31
32     public void closeCurrentSessionwithTransaction() {
33         currentTransaction.commit();
34         currentSession.close();
35     }
36
37     public void rollback() {
38         if (currentTransaction != null)
39             currentTransaction.rollback();
40     }
41
42     protected Session getCurrentSession() {
43         return currentSession;
44     }
45
46     public abstract void create(T entity);
47     public abstract void update(T entity);
48     public abstract T findById(Id id);
49     public abstract void delete(T entity);
```

```
50      public abstract List<T> findAll();
51  }
```

Archivo 6: GenericDao.java

```
1  package me.tonatihu.dao;
2
3  import java.io.Serializable;
4
5  /**
6   * @author tonatihu
7   * Created on 3/17/19
8   */
9
10 public interface UsuarioDao<T, Id extends Serializable> {
11      T findByUsernameAndContra(String username, String
       contra);
12      T findByUsername(String username);
13 }
```

Archivo 7: UsuarioDao.java

```
1  package me.tonatihu.dao.impl;
2
3  import me.tonatihu.dao.GenericDao;
4  import me.tonatihu.dao.UsuarioDao;
5  import me.tonatihu.entity.UsuarioEntity;
6  import org.hibernate.query.Query;
7
8  import java.util.List;
9
10 /**
11  * @author tonatihu
12  * Created on 3/17/19
13  */
14
15 public class UsuarioDaoImpl extends GenericDao<
       UsuarioEntity, String> implements
16 UsuarioDao<UsuarioEntity, String> {
17      private static final String FIND_BY_USERNAME_AND_CONTRA
       = "from
18 UsuarioEntity where username=:u and contra=:p";
```

```
19      private static final String FIND_BY_USERNAME = "from
    UsuarioEntity where
20 username=:u";
21
22      @Override
23      public UsuarioEntity findByUsernameAndContra(String
    username, String contra)
24 {
25          openCurrentSession();
26          Query query =
27 getCurrentSession().createQuery(FIND_BY_USERNAME_AND_CONTRA
    );
28          query.setParameter("u", username);
29          query.setParameter("p", contra);
30          UsuarioEntity usuario = (UsuarioEntity) query.
    uniqueResult();
31          closeCurrentSession();
32          return usuario;
33      }
34
35      @Override
36      public UsuarioEntity findByUsername(String username) {
37          openCurrentSession();
38          Query query = getCurrentSession().createQuery(
    FIND_BY_USERNAME);
39          query.setParameter("u", username);
40          UsuarioEntity usuario = (UsuarioEntity) query.
    uniqueResult();
41          closeCurrentSession();
42          return usuario;
43      }
44
45      @Override
46      public void create(UsuarioEntity entity) {
47
48      }
49
50      @Override
51      public void update(UsuarioEntity entity) {
52
53      }
54
55      @Override
```

```
56      public UsuarioEntity findById(String s) {
57          openCurrentSession();
58          UsuarioEntity usuario = getCurrentSession().get(
        UsuarioEntity.class, s);
59          closeCurrentSession();
60          return usuario;
61      }
62
63      @Override
64      public void delete(UsuarioEntity entity) {
65
66      }
67
68      @Override
69      public List<UsuarioEntity> findAll() {
70          return null;
71      }
72  }
```

Archivo 8: UsuarioDaoImpl.java

```
1  package me.tonatihu.dao.impl;
2
3  import me.tonatihu.dao.GenericDao;
4  import me.tonatihu.entity.CategoriaEntity;
5
6  import java.util.List;
7
8  /**
9   * @author tonatihu
10  * Created on 3/11/19
11  */
12
13 public class CategoriaDaoImpl extends GenericDao<
       CategoriaEntity, Integer> {
14      @Override
15      public void create(CategoriaEntity entity) {
16          getCurrentSession().persist(entity);
17      }
18
19      @Override
20      public void update(CategoriaEntity entity) {
21          getCurrentSession().update(entity);
```

```
22      }
23
24      @Override
25      public CategoriaEntity findById(Integer id) {
26          return getCurrentSession().get(CategoriaEntity.
    class, id);
27      }
28
29      @Override
30      public void delete(CategoriaEntity entity) {
31          getCurrentSession().delete(entity);
32      }
33
34      @Override
35      @SuppressWarnings("unchecked")
36      public List<CategoriaEntity> findAll() {
37          return (List<CategoriaEntity>) getCurrentSession().
    createQuery("from
38 CategoriaEntity ").list();
39      }
40 }
```

Archivo 9: CategoriaDaoImpl.java

```
1 package me.tonatihu.dao.impl;
2
3 import me.tonatihu.dao.GenericDao;
4 import me.tonatihu.dto.Dato;
5 import me.tonatihu.entity.ProductoEntity;
6 import org.hibernate.HibernateException;
7
8 import java.util.ArrayList;
9 import java.util.List;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12
13
14 /**
15  * @author tonatihu
16  * Created on 3/11/19
17  */
18
19 public class ProductoDaoImpl extends GenericDao<
```

```
    ProductoEntity, Integer> {
20      private static final Logger LOGGER =
21  Logger.getLogger(ProductoDaoImpl.class.getName());
22      @Override
23      public void create(ProductoEntity entity) {
24          openCurrentSessionWithTransaction();
25          try {
26              getCurrentSession().save(entity);
27          } catch (HibernateException hehe) {
28              LOGGER.log(Level.SEVERE, "Error en create",
    hehe);
29              rollback();
30          } finally {
31              closeCurrentSessionwithTransaction();
32          }
33      }
34
35      @Override
36      public void update(ProductoEntity entity) {
37          openCurrentSessionWithTransaction();
38          try {
39              getCurrentSession().update(entity);
40          } catch (HibernateException hehe) {
41              LOGGER.log(Level.SEVERE, "Error en update",
    hehe);
42              rollback();
43          } finally {
44              closeCurrentSessionwithTransaction();
45          }
46      }
47
48      @Override
49      public ProductoEntity findById(Integer integer) {
50          openCurrentSession();
51          ProductoEntity productoEntity =
52  getCurrentSession().get(ProductoEntity.class, integer);
53          closeCurrentSession();
54          return productoEntity;
55      }
56
57      @Override
58      public void delete(ProductoEntity entity) {
59          openCurrentSessionWithTransaction();
```

```
60          try {
61              getCurrentSession().delete(entity);
62          } catch (HibernateException hehe) {
63              LOGGER.log(Level.SEVERE, "Error en delete",
    hehe);
64              rollback();
65          } finally {
66              closeCurrentSessionwithTransaction();
67          }
68      }
69
70      @Override
71      @SuppressWarnings("unchecked")
72      public List<ProductoEntity> findAll() {
73          openCurrentSession();
74          List<ProductoEntity> lista = (List<ProductoEntity>)
    getCurrentSession()
75                  .createQuery("from ProductoEntity").list();
76          closeCurrentSession();
77          return lista;
78      }
79
80      public List<Dato> getDatos() {
81          openCurrentSession();
82          List lista = getCurrentSession()
83                  .createQuery("select c.nombre as nombre,
    count(p.nombre) as
84  catidad " +
85                          "from CategoriaEntity as c left
    join c.productos as p "
86  +
87                          "group by c.nombre").getResultList
    ();
88          List<Dato> datos = new ArrayList<>();
89          for (Object o : lista) {
90              Object[] filas = (Object[]) o;
91              datos.add(new Dato((String) filas[0], ((Long)
    filas[1]).intValue()));
92          }
93          closeCurrentSession();
94          return datos;
95      }
96      private static void printResult(Object result) {
```

```
97          if (result == null) {
98              System.out.print("NULL");
99          } else if (result instanceof Object[]) {
100             Object[] row = (Object[]) result;
101             System.out.print("[");
102             for (Object o : row) {
103                 printResult(o);
104             }
105             System.out.print("]");
106         } else if (result instanceof Long || result
    instanceof Double
107                 || result instanceof String) {
108             System.out.print("Chido: " + result.getClass().
    getName() + ": " +
109 result);
110         } else {
111             System.out.print("Resultado: " + result);
112         }
113     }
114
115 }
```

Archivo 10: ProductoDaoImpl.java

3. Pruebas

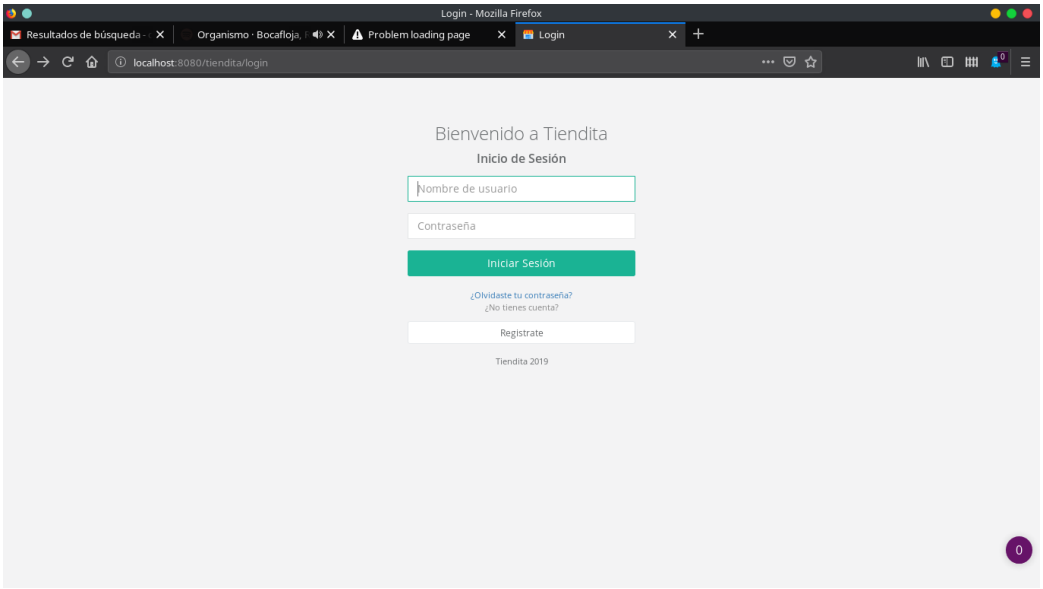

Figura 2: Login

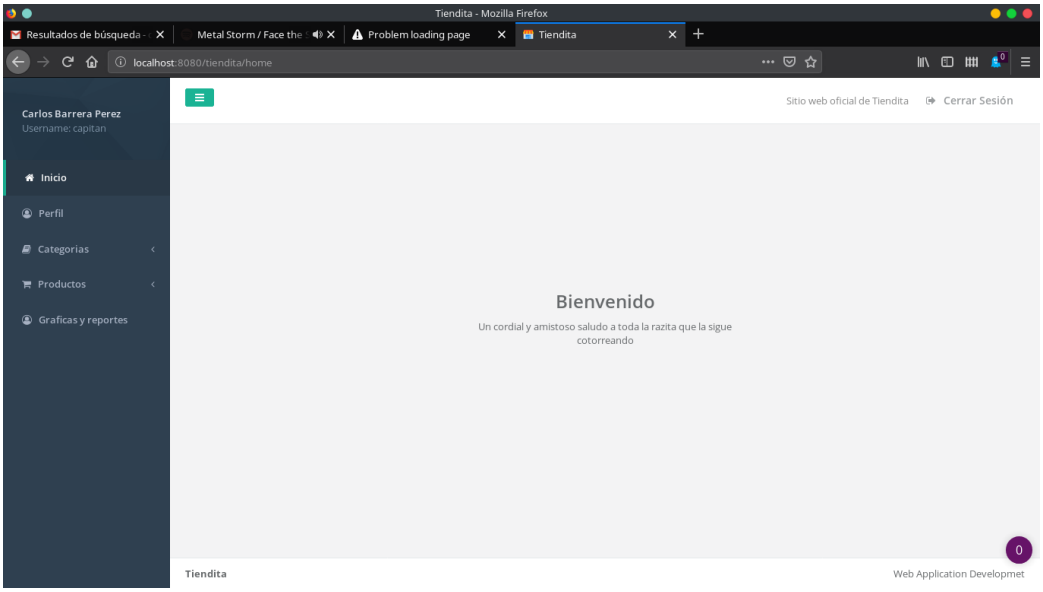

Figura 3: Home

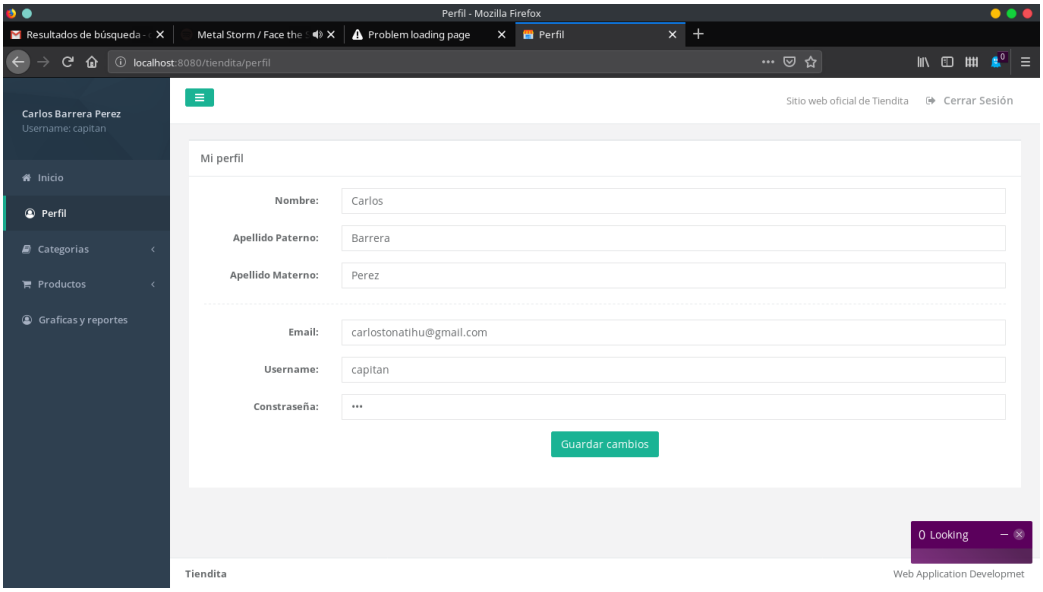

Figura 4: Perfil

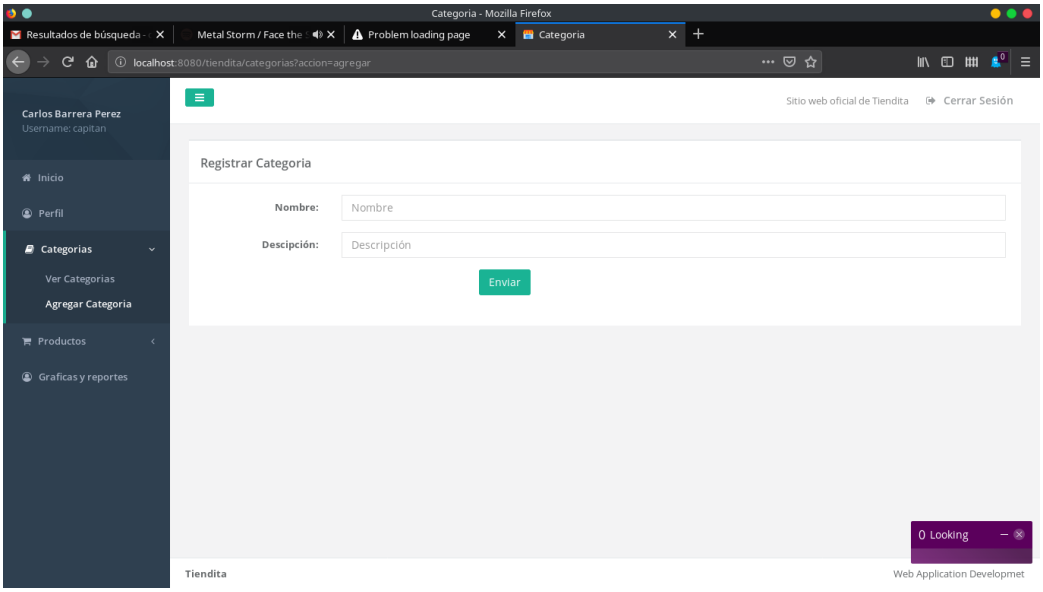

Figura 5: Agregar categoría

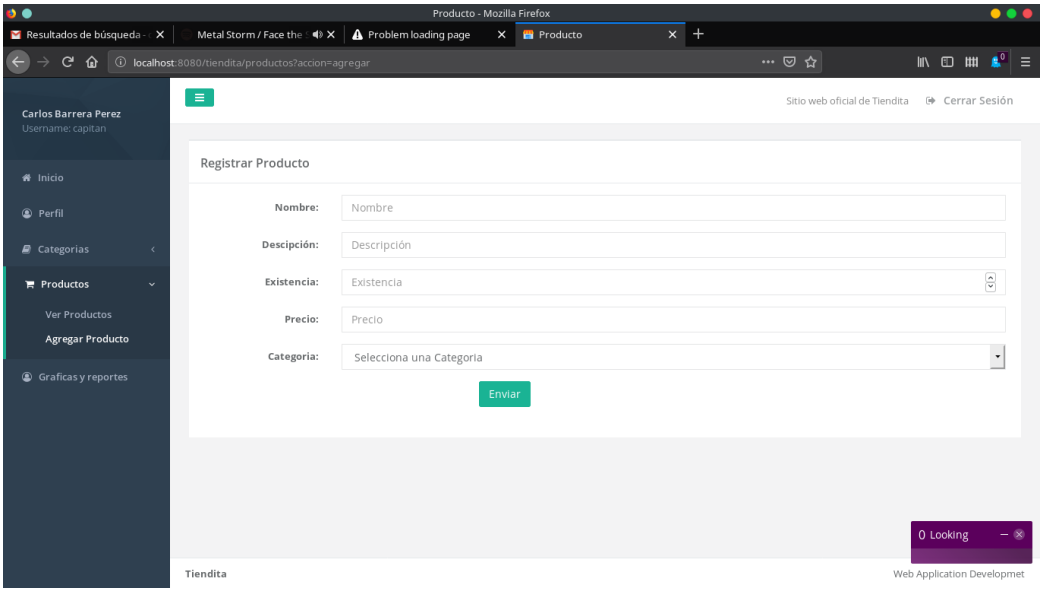

Figura 6: Agregar producto

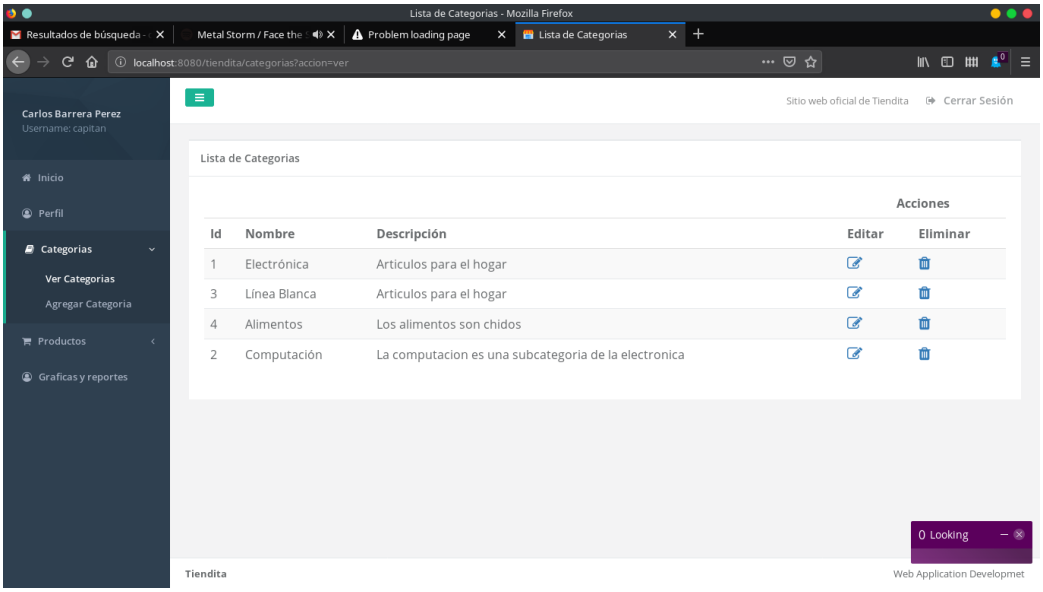

Figura 7: Ver categorías

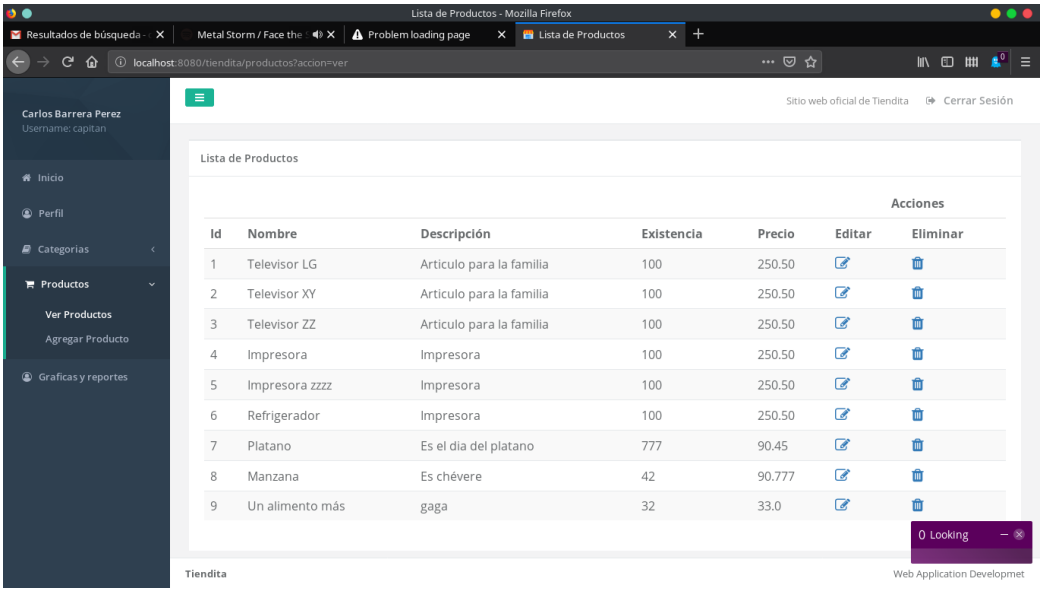

Figura 8: Ver productos

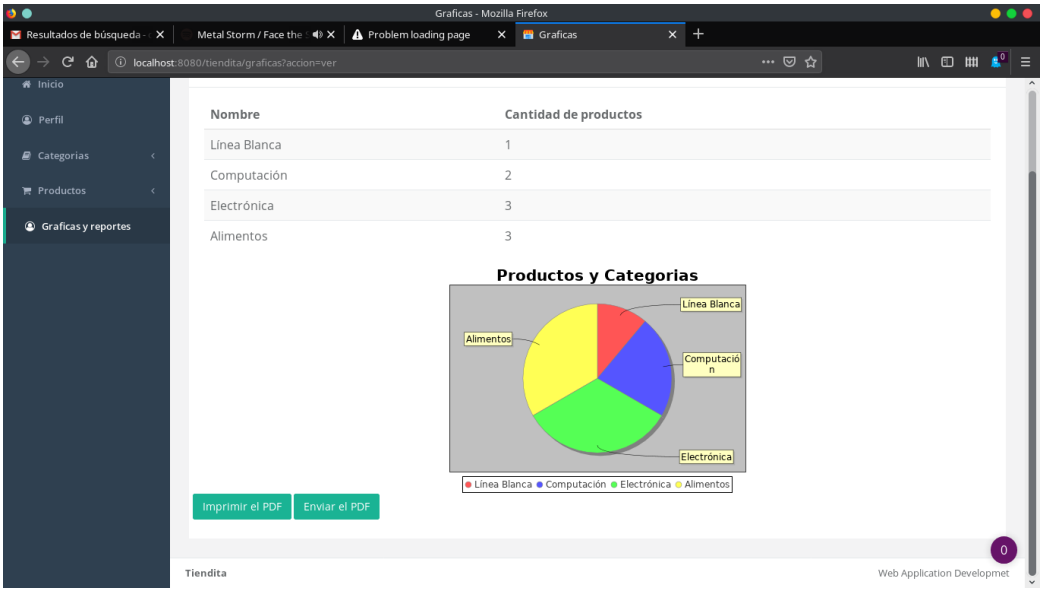

Figura 9: Gráficas

Tiendita

Fecha: Monday 03 June

La neta no recuerdo que era esto

Nombre del producto o algo así	Creo que esto era la cantidad
Línea Blanca	1
Computación	2
Electrónica	3
Alimentos	3

Figura 10: Reporte

Figura 11: Envío del reporte y la gráfica por email

4. Conclusiones

La parte complicada de esta materia fue el generar el reporte debido a que trabajar con jasper no es tan sencillo como parece, por lo cual consumió gran parte de la elaboración de esta practica, el resto no fue tan difícil debido a que ya se había trabajado con cosas similares en ejercicios anteriores