

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Controlador;

import HBM.Titulo;
import HBM.TituloId;
import Modelo.Biblioteca;
import Modelo.Catalogo;
import Modelo.CodDewey;
import Modelo.Login;
import Modelo.Usuario;
import java.beans.PropertyChangeEvent;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author nanohp
 */
public class Controlador extends AbstractController implements GestorEventos {

    private Biblioteca biblioteca;

    public Controlador(Biblioteca biblioteca) {
        this.biblioteca = biblioteca;
    }

    public void procesarEvento(Evento evento) {
        Evento eventoAux = null;

        switch (evento.getTipoEvento()) {
            case SALIR:
                System.out.println("Saliendo");
                System.exit(1);
                break;
            case LOGIN:
                if (!evento.getInfo().getClass().getName().equals("Modelo.Login")) {
                    assert false : evento.getInfo().getClass().getName() + " clase invalida";
                } else {
                    Login login = (Login) evento.getInfo();
                    System.out.println("Intento acceso para: " + login.toString());
                    Usuario usuario;
                    try {
                        usuario = biblioteca.login(login.getNombre(), login.getClave());
                        ((Usuario) usuario).setLookAndFeel(login.getLookAndFeel());
                        //System.out.println("---"+usuario.getClass().getSimpleName()+"-");
                        eventoAux = new Evento(TipoEvento.LOGIN_OK, usuario);
                    } catch (Exception ex) {
                        //Logger.getLogger(Controlador.class.getName()).log(Level.SEVERE, null, ex);
                        System.err.println("ERROR");
                        eventoAux = new Evento(TipoEvento.LOGIN_FALL0);
                    }
                    evento.getDestinoRespueta().procesarEvento(eventoAux);
                }
                break;
            case LOGOUT:
                evento.getDestinoRespueta().procesarEvento(new Evento(TipoEvento.LOGOUT));
                break;
            case OBTENER_CAT_DEWEY:
                evento.getDestinoRespueta().procesarEvento(new Evento(TipoEvento.OBTENER_CAT_DEWEY,
                    biblioteca.getCategoriasDewey()));
                break;
            case CONSULTA_CATALOGO_GENERAL:

                System.out.println("CONSULTA_CATALOGO_GENERAL");
                try {

                    Catalogo catalogo = biblioteca.getAlberga();
```

```

        eventoAux = new Evento(TipoEvento.CONSULTA_CATALOGO_GENERAL, catalogo);

    } catch (Exception ex) {
        Logger.getLogger(Controlador.class.getName()).log(Level.SEVERE, null, ex);
        System.err.println("CONSULTA_CATALOGO_GENERAL: ERROR");
        eventoAux = new Evento(TipoEvento.ERROR);
    }
    evento.getDestinoRespueta().procesarEvento(eventoAux);
    break;
case CONSULTA_CATALOGO_CONCRETA:
    System.out.println("CONSULTA_CATALOGO_CONCRETA" + evento.getInfo().getClass().getName
());
    if (!evento.getInfo().getClass().getName().startsWith("HBM.TituloId")) {
        assert false : evento.getInfo().getClass().getName() + " clase invalida";
    }

    try {
        TituloId tituloId = (TituloId) evento.getInfo();

        Titulo titulo = biblioteca.getAlberga().getCatalogo().get(tituloId);
        eventoAux = new Evento(TipoEvento.CONSULTA_CATALOGO_CONCRETA, titulo);

    } catch (Exception ex) {
        Logger.getLogger(Controlador.class.getName()).log(Level.SEVERE, null, ex);
        System.err.println("CONSULTA_CATALOGO_CONCRETA: ERROR");
        eventoAux = new Evento(TipoEvento.ERROR);
    }
    evento.getDestinoRespueta().procesarEvento(eventoAux);
    break;
}

}

public static final String ELEMENT_TEXT_PROPERTY = "Texto";

public void changeElementText(String newText) {
    setModelProperty(ELEMENT_TEXT_PROPERTY, newText);
}

@Override
public void modelPropertyChange(PropertyChangeEvent evt) {
    throw new UnsupportedOperationException("Not supported yet.");
}
}

```