

```

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

import Controlador.Controlador;
import Controlador.Evento;
import Controlador.GestorEventos;
import Modelo.Biblioteca;
import Modelo.MenuData;
import Modelo.Usuario;
import java.beans.PropertyChangeEvent;
import java.util.Observable;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import javax.swing.plaf.metal.DefaultMetalTheme;
import javax.swing.plaf.metal.MetalLookAndFeel;
import javax.swing.plaf.metal.OceanTheme;
/**
 *
 * @author nanohp
 */
public class VistaGrafica implements IVista, GestorEventos {

    private int estado;
    private Controlador controlador;
    private VistaGLogin vGLogin;
    private VistaGPrincipal vGPrincipal;
    private Observable modelo;
    private Biblioteca biblioteca;
    private String lookAndFeel;
    private final static String LOOKANDFEEL = "Metal";
    private final static String THEME = "Ocean";

    public VistaGrafica(Biblioteca biblioteca) {
        this.biblioteca = biblioteca;
        inicializaLookAndFeelInicial();
    }

    public VistaGrafica(MenuData menu) {
        inicializaLookAndFeelInicial();
    }

    private void inicializaLookAndFeelInicial() {
        try {
            //lookAndFeel = "com.sun.java.swing.plaf.motif.MotifLookAndFeel";
            String lookAndFeel = "com.sun.java.swing.plaf.gtk.GTKLookAndFeel";
            UIManager.setLookAndFeel(lookAndFeel);
        } catch (UnsupportedLookAndFeelException ex) {
            Logger.getLogger(VistaGrafica.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(VistaGLogin.class.getName()).log(Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            Logger.getLogger(VistaGLogin.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            Logger.getLogger(VistaGLogin.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public void arranca() {
        vGLogin = new VistaGLogin(this, controlador);
        vGLogin.setVisible(true);
    }

    public void operacion(Usuario usuario) {
        vGPrincipal = new VistaGPrincipal(this, controlador, usuario);
        vGPrincipal.setVisible(true);
    }

```

```

    }

    @Override
    public void setModelo(Observable observable) {
        modelo = observable;
    }

    @Override
    public String getValor(String etiqueta) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void mostrarMensaje(String mensaje) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

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

    @Override
    public void procesarEvento(Evento evento) {

        Object info = evento.getInfo();

        switch (evento.getTipoEvento()) {

            case FIJAR_LYL:
                lookAndFeel = (String)info;
                break;
            case LOGIN_OK:
                if (info.getClass().getName().startsWith("Modelo.Usuario")) {
                    //vGLogin.setVisible(false);
                    //vGLogin = null;
                    Usuario usuarioActivo = (Usuario) info;
                    //this.initLookAndFeelAplicacion(lookAndFeel);
                    this.initLookAndFeelAplicacion(usuarioActivo.getLookAndFeel());

                    operacion(usuarioActivo);
                } else {
                    assert false : info.getClass().getName() + " clase no valida";
                }
                break;
            case LOGOUT:
                vGPrincipal.setVisible(false);
                //vGPrincipal = null;
                vGLogin.limpiarModelo();
                vGLogin.setVisible(true);

                break;
            default:
                System.err.println("eventoRespuesta() no tratado");
        }
    }

    @Override
    public void setControlador(Controlador controlador) {
        this.controlador = controlador;
    }

    private void initLookAndFeelAplicacion(String lookAndFeelStr) {

        if (lookAndFeel != null) {
            if (lookAndFeelStr.equals("Metal")) {
                lookAndFeel = UIManager.getCrossPlatformLookAndFeelClassName();
                // an alternative way to set the Metal L&F is to replace the
                // previous line with:
                // lookAndFeel = "javax.swing.plaf.metal.MetalLookAndFeel";
            }
        }
    }

```

```

    } else if (lookAndFeelStr.equals("System")) {
        lookAndFeel = UIManager.getSystemLookAndFeelClassName();
    } else if (lookAndFeelStr.equals("Motif")) {
        lookAndFeel = "com.sun.java.swing.plaf.motif.MotifLookAndFeel";
    } else if (lookAndFeelStr.equals("GTK")) {
        lookAndFeel = "com.sun.java.swing.plaf.gtk.GTKLookAndFeel";
    } else {
        System.err.println("Unexpected value of LOOKANDFEEL specified: "
            + lookAndFeel);
        lookAndFeel = UIManager.getCrossPlatformLookAndFeelClassName();
    }

    try {

        UIManager.setLookAndFeel(lookAndFeel);

        // If L&F = "Metal", set the theme

        if (lookAndFeel.equals("Metal")) {
            if (THEME.equals("DefaultMetal")) {
                MetalLookAndFeel.setCurrentTheme(new DefaultMetalTheme());
            } else if (THEME.equals("Ocean")) {
                MetalLookAndFeel.setCurrentTheme(new OceanTheme());
            } else {
                MetalLookAndFeel.setCurrentTheme(new TestTheme());
            }

            UIManager.setLookAndFeel(new MetalLookAndFeel());
        }
    } catch (ClassNotFoundException e) {
        System.err.println("Couldn't find class for specified look and feel:"
            + lookAndFeel);
        System.err.println("Did you include the L&F library in the class path?");
        System.err.println("Using the default look and feel.");
    } catch (UnsupportedLookAndFeelException e) {
        System.err.println("Can't use the specified look and feel ("
            + lookAndFeel
            + ") on this platform.");
        System.err.println("Using the default look and feel.");
    } catch (Exception e) {
        System.err.println("Couldn't get specified look and feel ("
            + lookAndFeel
            + "), for some reason.");
        System.err.println("Using the default look and feel.");
        e.printStackTrace();
    }
}
}
}
}

```