```java
/**
 * AplicacionVista.java
 * <p>
 * Clase que representa la vista de la aplicación implementada.
 * <p>
 * Radu Constantin Robu y Alberto Pérez
 */
package P3.Vista;

import javax.swing.*;
import javax.swing.border.Border;
import javax.swing.border.EmptyBorder;
import javax.swing.border.EtchedBorder;
import javax.swing.text.SimpleAttributeSet;
import javax.swing.text.StyleConstants;
import javax.swing.text.StyledDocument;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;

import P3.Control.OyenteVista;
import P3.Modelo.Tupla;

public class AplicacionVista implements ActionListener,
        PropertyChangeListener {
    private static final int ANCHO = 600;
    private static final int ALTO = 800;
    private static final int ANCHO_PANEL_BOTONES = 200;
    private static final int ALTO_PANEL_BOTONES = 200;
    private static final int ANCHO_PANEL_TAREAS = 400;
    private static final int ALTO_PANEL_TAREAS = 200;
    private static final int INDENT_TAREAS = 70;
    private static final int TAMANYO_NOMBRE_TAREA = 16;
    private static final int TAMANYO_DESCRIPCION_TAREA = 32;
    private static final int TAMANYO_CAMPOS_TEXTO = 10;

    private static final String FORMATO_FECHA = "^[0-9]{2} [0-9]{2} [0-9]{4}$";
    private static final String PATRON_CADENA_DIGITOS = "^[0-9]+$";

    private OyenteVista oyenteVista;
    private static final String NUEVO_FICHERO = "Nuevo fichero";
    private static final String ANYADIR_TAREA = "Añadir tarea";
    private static final String ELIMINAR_TAREA = "Eliminar tarea";
    private static final String BUSCAR_TAREA = "Buscar tareas";
    private static final String LISTAR_TAREAS = "Listar tareas";
    private static final String GUARDAR_TAREAS = "Guardar tareas";
    private static final String SALIR = "Salir";
    private static final String MAINFRAME_WRAPPER = "Mainframe Wrapper";
    private static final String ACCION_NUEVO_FICHERO = "NUEVO_FICHERO";
    private static final String ACCION_ANYADIR_TAREA = "ANYADIR_TAREA";
    private static final String ACCION_ELIMINAR_TAREA = "ELIMINAR_TAREA";
    private static final String ACCION_BUSCAR_TAREAS = "BUCAR_TAREAS";
    private static final String ACCION_LISTAR_TAREAS = "LISTAR_TAREAS";
```

```
56      private static final String ACCION_GUARDAR_TAREAS = "GUARDAR_TAREAS";
57      private static final String ACCION_SALIR = "SALIR";
58      private static final String TITULO_GUARDAR_SALIR = "Guardar y salir";
59      private static final String MENSAJE_GUARDAR_CAMBIOS =
60              "¿Desea guardar los cambios?";
61
62      private static final String TITULO_VENTANA_INICIAR_SESION =
63              "Conexión con el mainframe";
64      private static final String[] CAMPOS_INICIAR_SESION =
65              {"Host", "Usuario", "Contraseña"};
66      private static final String[] OPCIONES_INICIAR_SESION =
67              {"Iniciar sesión", "Cancelar"};
68
69      private static final String TITULO_VENTANA_ANYADIR_TAREA =
70              "Añadir una tarea";
71      private static final String[] CAMPOS_ANYADIR_TAREA =
72              {"Id", "Nombre", "Descripción", "Fecha"};
73      private static final String[] OPCIONES_ANYADIR_TAREA =
74              {"Añadir", "Cancelar"};
75
76      private static final String TITULO_VENTANA_BUSCAR_TAREAS = "Buscar tareas";
77      private static final String[] CAMPOS_BUSCAR_TAREAS = {"Fecha"};
78      private static final String[] OPCIONES_BUSCAR_TAREAS =
79              {"Buscar", "Cancelar"};
80
81      private static final String TITULO_VENTANA_ELIMINAR_TAREA =
82              "Eliminar tarea";
83      private static final String[] CAMPOS_ELIMINAR_TAREA = {"Id"};
84      private static final String[] OPCIONES_ELIMINAR_TAREA =
85              {"Eliminar", "Cancelar"};
86
87
88      private static final String ETIQUETA_VENTANA_ERROR_FECHA =
89              "Fecha incorrecta";
90      private static final String MENSAJE_FECHA_INCORRECTA =
91              "El formato de la fecha introducida no es correcto.\n" +
92              "Formato aceptado:\n" + "DD MM AAAA\n";
93
94      private static final String ETIQUETA_VENTANA_ID_INCORRECTO =
95              "ID incorrecto";
96      private static final String MENSAJE_ID_INCORRECTO =
97              "El id de una tarea debe contener únicamente números.";
98
99      private static final String ETIQUETA_VENTANA_NOMBRE_INCORRECTO =
100             "Nombre incorrecto";
101     private static final String MENSAJE_NOMBRE_INCORRECTO =
102             "El nombre de la tarea debe tener menos de " +
103             TAMANYO_NOMBRE_TAREA + " caracteres.";
104
105     private static final String ETIQUETA_VENTANA_DESCRIPCION_INCORRECTA =
106             "Descripción incorrecta";
107     private static final String MENSAJE_DESCRIPCION_INCORRECTA =
108             "La descripción de la tarea debe tener menos de " +
109             TAMANYO_DESCRIPCION_TAREA + " caracteres.";
110
```

```
111     private String[] datosInicioSesion;
112
113     private JButton botonNuevoFichero;
114     private JButton botonAnyadirTarea;
115     private JButton botonEliminarTarea;
116     private JButton botonBuscarTarea;
117     private JButton botonListarTareas;
118     private JButton botonGuardarTareas;
119     private JButton botonSalir;
120     private JPanel panelTareas;
121     private JTextPane areaTextoTareas;
122
123     private enum CodigoRespuesta {
124         OK, ERROR_ID_TAREA, ERROR_LONG_NOMBRE, DATOS_VACIOS,
125         ERROR_FECHA_INCORRECTA, ERROR_LONG_DESCRIPCION
126     }
127
128     /**
129      * Constructor de la clase.
130      */
131     public AplicacionVista(OyenteVista oyenteVista) {
132         datosInicioSesion = iniciarSesion();
133         this.oyenteVista = oyenteVista;
134     }
135
136     public String[] obtenerDatosInicioSesion() {
137         return datosInicioSesion;
138     }
139
140     /**
141      * Crea la ventana principal de la interfaz y sus elementos.
142      */
143     public void crearElementosVentanaPrincipal() {
144         JFrame ventanaPrincipal = new JFrame(MAINFRAME_WRAPPER);
145         ventanaPrincipal.setSize(ANCHO, ALTO);
146         ventanaPrincipal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
147         ventanaPrincipal.setLocationRelativeTo(null);
148         ventanaPrincipal.setLayout(new BorderLayout());
149         ventanaPrincipal.add(crearPanelBotones());
150         ventanaPrincipal.add(crearPanelMostrarTareas(), BorderLayout.EAST);
151         ventanaPrincipal.setVisible(true);
152
153     }
154
155     /**
156      * Crea el panel de tareas de la ventana principal.
157      *
158      * @return
159      */
160     public JPanel crearPanelMostrarTareas() {
161         panelTareas = new JPanel(new BorderLayout());
162         Border bordePanelTareas =
163                 BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);
164         panelTareas.setBorder(bordePanelTareas);
165         panelTareas.setPreferredSize(new Dimension(ANCHO_PANEL_TAREAS,
```

```
166                 ALTO_PANEL_TAREAS));
167
168         JScrollPane panelDeslizable = new JScrollPane();
169         areaTextoTareas = new JTextPane();
170         areaTextoTareas.setEditable(false);
171         areaTextoTareas.setAlignmentX(70);
172         panelDeslizable.setViewportView(areaTextoTareas);
173         panelTareas.add(panelDeslizable);
174
175         StyledDocument doc = areaTextoTareas.getStyledDocument();
176
177         SimpleAttributeSet center = new SimpleAttributeSet();
178         StyleConstants.setLeftIndent(center, INDENT_TAREAS);
179         doc.setParagraphAttributes(0, doc.getLength(), center, false);
180
181         return panelTareas;
182     }
183
184
185     /**
186      * Crea el panel de botones de la ventana principal.
187      *
188      * @return
189      */
190     private JPanel crearPanelBotones() {
191         crearBotones();
192         JPanel panelBotones = new JPanel(new BorderLayout());
193         panelBotones.setBorder(new EmptyBorder(2, 3, 2, 3));
194
195         JPanel layoutBotones = new JPanel(new GridBagLayout());
196         layoutBotones.setBorder(new EmptyBorder(5, 5, 5, 5));
197         Border bordePanelBotones =
198                 BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);
199         layoutBotones.setBorder(bordePanelBotones);
200
201         JPanel contenedorBotones =
202                 new JPanel(new GridLayout(10, 1, 10, 20));
203
204         contenedorBotones.add(botonNuevoFichero);
205         contenedorBotones.add(botonAnyadirTarea);
206         contenedorBotones.add(botonEliminarTarea);
207         contenedorBotones.add(botonBuscarTarea);
208         contenedorBotones.add(botonListarTareas);
209         contenedorBotones.add(botonGuardarTareas);
210         contenedorBotones.add(botonSalir);
211         layoutBotones.add(contenedorBotones);
212         layoutBotones.setPreferredSize(new Dimension(ANCHO_PANEL_BOTONES,
213                 ALTO_PANEL_BOTONES));
214         panelBotones.setPreferredSize(new Dimension(ANCHO_PANEL_BOTONES,
215                 ALTO_PANEL_BOTONES));
216         contenedorBotones.setPreferredSize(new Dimension(ANCHO_PANEL_BOTONES,
217                 ALTO_PANEL_BOTONES));
218         panelBotones.add(layoutBotones, BorderLayout.WEST);
219
220         return panelBotones;
```

```
221     }
222
223     /**
224      * Crea los botones del panel de botones.
225      */
226     private void crearBotones() {
227         botonNuevoFichero = new JButton(NUEVO_FICHERO);
228         botonNuevoFichero.addActionListener(this);
229         botonNuevoFichero.setActionCommand(ACCION_NUEVO_FICHERO);
230
231         botonAnyadirTarea = new JButton(ANYADIR_TAREA);
232         botonAnyadirTarea.addActionListener(this);
233         botonAnyadirTarea.setActionCommand(ACCION_ANYADIR_TAREA);
234
235         botonEliminarTarea = new JButton(ELIMINAR_TAREA);
236         botonEliminarTarea.addActionListener(this);
237         botonEliminarTarea.setActionCommand(ACCION_ELIMINAR_TAREA);
238
239         botonBuscarTarea = new JButton(BUSCAR_TAREA);
240         botonBuscarTarea.addActionListener(this);
241         botonBuscarTarea.setActionCommand(ACCION_BUSCAR_TAREAS);
242
243         botonListarTareas = new JButton(LISTAR_TAREAS);
244         botonListarTareas.addActionListener(this);
245         botonListarTareas.setActionCommand(ACCION_LISTAR_TAREAS);
246
247         botonGuardarTareas = new JButton(GUARDAR_TAREAS);
248         botonGuardarTareas.addActionListener(this);
249         botonGuardarTareas.setActionCommand(ACCION_GUARDAR_TAREAS);
250
251         botonSalir = new JButton(SALIR);
252         botonSalir.addActionListener(this);
253         botonSalir.setActionCommand(ACCION_SALIR);
254     }
255
256     /**
257      * Método de tratamiento de eventos.
258      *
259      * @param e
260      */
261     @Override
262     public void actionPerformed(ActionEvent e) {
263         switch (e.getActionCommand()) {
264             case ACCION_NUEVO_FICHERO:
265                 oyenteVista.eventoProducido(OyenteVista.Evento.NUEVO_FICHERO,
266                         null);
267                 break;
268
269             case ACCION_ANYADIR_TAREA:
270                 String[] datosAnyadirTarea = anyadirTarea();
271                 if (datosAnyadirTarea != null) {
272                     Tupla<Tupla, Tupla> tupla = new Tupla(
273                         new Tupla(datosAnyadirTarea[0], datosAnyadirTarea[1]),
274                         new Tupla(datosAnyadirTarea[2], datosAnyadirTarea[3]));
275                     oyenteVista.eventoProducido(
```

```
                                OyenteVista.Evento.ANYADIR_TAREA, tupla);
                }
                break;

            case ACCION_BUSCAR_TAREAS:
                String[] datosBuscarTareas = buscarTareas();
                if (datosBuscarTareas != null) {
                    oyenteVista.eventoProducido(OyenteVista.Evento.BUSCAR_TAREA,
                            datosBuscarTareas[0]);
                }
                break;

            case ACCION_SALIR:
                int resp = JOptionPane.showConfirmDialog(null,
                        MENSAJE_GUARDAR_CAMBIOS,
                        TITULO_GUARDAR_SALIR, JOptionPane.YES_NO_OPTION);
                if (resp == JOptionPane.YES_OPTION) {
                    oyenteVista.eventoProducido(OyenteVista.Evento.SALIR, "y");
                } else if (resp == JOptionPane.NO_OPTION) {
                    oyenteVista.eventoProducido(OyenteVista.Evento.SALIR, "n");
                }
                break;

            case ACCION_ELIMINAR_TAREA:
                String[] datosEliminarTarea = obtenerDatosEliminarTarea();
                if (datosEliminarTarea != null) {
                    oyenteVista.eventoProducido(
                            OyenteVista.Evento.ELIMINAR_TAREA,
                            datosEliminarTarea[0]);
                }
                break;

            case ACCION_LISTAR_TAREAS:
                oyenteVista.eventoProducido(OyenteVista.Evento.LISTAR_TAREAS,
                        null);
                break;

            case ACCION_GUARDAR_TAREAS:
                oyenteVista.eventoProducido(OyenteVista.Evento.GUARDAR_TAREAS,
                        null);
                break;
        }
    }

    /**
     * Método para mostrar tareas en un mainframe.
     */
    public void mostrarTareas(String tareas) {
        areaTextoTareas.setText("");
        areaTextoTareas.setText(tareas);
    }

    /**
     * Método para iniciar sesión en un mainframe.
     */
```

```
331     private String[] iniciarSesion() {
332         ComplexDialoguePanel ventanaIniciarSesion =
333                 new ComplexDialoguePanel(TITULO_VENTANA_INICIAR_SESION,
334                 CAMPOS_INICIAR_SESION, 16);
335         String[] datosInicioSesion =
336                 ventanaIniciarSesion.obtenerTextoCampos(
337                         OPCIONES_INICIAR_SESION);
338 
339         return datosInicioSesion;
340     }
341 
342     /**
343      * Método para añadir una tarea en el mainframe.
344      *
345      * @return
346      */
347     private String[] anyadirTarea() {
348         String[] datosAnyadirTarea = null;
349 
350         ComplexDialoguePanel ventanaAnyadirTarea = new ComplexDialoguePanel(
351                 TITULO_VENTANA_ANYADIR_TAREA, CAMPOS_ANYADIR_TAREA,
352                 TAMANYO_DESCRIPCION_TAREA);
353         datosAnyadirTarea = ventanaAnyadirTarea.obtenerTextoCampos(
354                 OPCIONES_ANYADIR_TAREA);
355 
356         CodigoRespuesta codigo = verificarDatosTarea(datosAnyadirTarea);
357 
358         switch (codigo) {
359             case OK:
360                 return datosAnyadirTarea;
361 
362             case ERROR_FECHA_INCORRECTA:
363                 JOptionPane.showMessageDialog(new JFrame(),
364                         MENSAJE_FECHA_INCORRECTA, ETIQUETA_VENTANA_ERROR_FECHA,
365                         JOptionPane.ERROR_MESSAGE);
366                 break;
367 
368             case ERROR_ID_TAREA:
369                 JOptionPane.showMessageDialog(new JFrame(),
370                         MENSAJE_ID_INCORRECTO, ETIQUETA_VENTANA_ID_INCORRECTO,
371                         JOptionPane.ERROR_MESSAGE);
372                 break;
373 
374             case ERROR_LONG_NOMBRE:
375                 JOptionPane.showMessageDialog(new JFrame(),
376                         MENSAJE_NOMBRE_INCORRECTO,
377                         ETIQUETA_VENTANA_NOMBRE_INCORRECTO,
378                         JOptionPane.ERROR_MESSAGE);
379                 break;
380 
381             case ERROR_LONG_DESCRIPCION:
382                 JOptionPane.showMessageDialog(new JFrame(),
383                         MENSAJE_DESCRIPCION_INCORRECTA,
384                         ETIQUETA_VENTANA_DESCRIPCION_INCORRECTA,
385                         JOptionPane.ERROR_MESSAGE);
```

```
386                 break;
387             }
388             return null;
389         }
390 
391         /**
392          * Comprueba los datos que el usuario ha introducido.
393          *
394          * @param datosTarea
395          * @return
396          */
397         private CodigoRespuesta verificarDatosTarea(String[] datosTarea) {
398             if (datosTarea != null) {
399                 if (datosTarea[0].matches(PATRON_CADENA_DIGITOS)) {
400                     if ((datosTarea[1].length() > 0) &&
401                             (datosTarea[1].length() <= TAMANYO_NOMBRE_TAREA)) {
402                         if ((datosTarea[2].length() > 0) &&
403                                 (datosTarea[2].length() <=
404                                         TAMANYO_DESCRIPCION_TAREA)) {
405                             if (datosTarea[3].matches(FORMATO_FECHA)) {
406                                 return CodigoRespuesta.OK;
407                             } else {
408                                 return CodigoRespuesta.ERROR_FECHA_INCORRECTA;
409                             }
410                         }
411                         return CodigoRespuesta.ERROR_LONG_DESCRIPCION;
412                     } else {
413                         return CodigoRespuesta.ERROR_LONG_NOMBRE;
414                     }
415                 } else {
416                     return CodigoRespuesta.ERROR_ID_TAREA;
417                 }
418             } else {
419                 return CodigoRespuesta.DATOS_VACIOS;
420             }
421         }
422 
423         /**
424          * Método para buscar tareas en el mainframe por fecha.
425          *
426          * @return
427          */
428         private String[] buscarTareas() {
429             ComplexDialoguePanel ventanaBuscarTareas =
430                     new ComplexDialoguePanel(TITULO_VENTANA_BUSCAR_TAREAS,
431                     CAMPOS_BUSCAR_TAREAS, TAMANYO_CAMPOS_TEXTO);
432             String[] datosBuscarTareas =
433                     ventanaBuscarTareas.obtenerTextoCampos(OPCIONES_BUSCAR_TAREAS);
434 
435             return datosBuscarTareas;
436         }
437 
438         /**
439          * Método para ELIMINAR una tarea.
440          *
```

```
441      * @return
442      */
443     private String[] obtenerDatosEliminarTarea() {
444         ComplexDialoguePanel ventanaEliminarTareas =
445                 new ComplexDialoguePanel(TITULO_VENTANA_ELIMINAR_TAREA,
446                 CAMPOS_ELIMINAR_TAREA, TAMANYO_CAMPOS_TEXTO);
447         String[] datoseliminarTareas =
448                 ventanaEliminarTareas.obtenerTextoCampos(
449                         OPCIONES_ELIMINAR_TAREA);
450 
451         return datoseliminarTareas;
452     }
453 
454     /**
455      * Muestra un diálogo con el error que se ha producido.
456      *
457      * @param titulo
458      * @param mensaje
459      */
460     public void notificarMensajeError(String titulo, String mensaje) {
461         JOptionPane.showMessageDialog(new JFrame(), mensaje, titulo,
462                 JOptionPane.ERROR_MESSAGE);
463     }
464 
465     /**
466      * Muestra un diálogo con la confirmación de la tarea.
467      *
468      * @param titulo
469      * @param mensaje
470      */
471     public void notificarMensajeConfirmacion(String titulo, String mensaje) {
472         JOptionPane.showMessageDialog(new JFrame(), mensaje, titulo,
473                 JOptionPane.INFORMATION_MESSAGE);
474     }
475 
476     /**
477      * Método de cambios de propiedades de los elementos de la ventana.
478      *
479      * @param evt
480      */
481     @Override
482     public void propertyChange(PropertyChangeEvent evt) {
483     }
484 }
485 
486 
```