```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_CAMPOS_INICIO_SESION = 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 static final String RESPUESTA_SI = "y";
    private static final String RESPUESTA_NO = "n";


    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";
```

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

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

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

```
221         contenedorBotones.add(botonSalir);
222         layoutBotones.add(contenedorBotones);
223         layoutBotones.setPreferredSize(new Dimension(ANCHO_PANEL_BOTONES,
224                 ALTO_PANEL_BOTONES));
225         panelBotones.setPreferredSize(new Dimension(ANCHO_PANEL_BOTONES,
226                 ALTO_PANEL_BOTONES));
227         contenedorBotones.setPreferredSize(new Dimension(ANCHO_PANEL_BOTONES,
228                 ALTO_PANEL_BOTONES));
229         panelBotones.add(layoutBotones, BorderLayout.WEST);
230 
231         return panelBotones;
232     }
233 
234     /**
235      * Crea los botones del panel de botones.
236      */
237     private void crearBotones() {
238         botonNuevoFichero = new JButton(NUEVO_FICHERO);
239         botonNuevoFichero.addActionListener(this);
240         botonNuevoFichero.setActionCommand(ACCION_NUEVO_FICHERO);
241 
242         botonAnyadirTarea = new JButton(ANYADIR_TAREA);
243         botonAnyadirTarea.addActionListener(this);
244         botonAnyadirTarea.setActionCommand(ACCION_ANYADIR_TAREA);
245 
246         botonEliminarTarea = new JButton(ELIMINAR_TAREA);
247         botonEliminarTarea.addActionListener(this);
248         botonEliminarTarea.setActionCommand(ACCION_ELIMINAR_TAREA);
249 
250         botonBuscarTarea = new JButton(BUSCAR_TAREA);
251         botonBuscarTarea.addActionListener(this);
252         botonBuscarTarea.setActionCommand(ACCION_BUSCAR_TAREAS);
253 
254         botonListarTareas = new JButton(LISTAR_TAREAS);
255         botonListarTareas.addActionListener(this);
256         botonListarTareas.setActionCommand(ACCION_LISTAR_TAREAS);
257 
258         botonGuardarTareas = new JButton(GUARDAR_TAREAS);
259         botonGuardarTareas.addActionListener(this);
260         botonGuardarTareas.setActionCommand(ACCION_GUARDAR_TAREAS);
261 
262         botonSalir = new JButton(SALIR);
263         botonSalir.addActionListener(this);
264         botonSalir.setActionCommand(ACCION_SALIR);
265     }
266 
267     /**
268      * Método de tratamiento de eventos.
269      *
270      * @param e
271      */
272     @Override
273     public void actionPerformed(ActionEvent e) {
274         switch (e.getActionCommand()) {
275             case ACCION_NUEVO_FICHERO:
```

```
276                 oyenteVista.eventoProducido(OyenteVista.Evento.NUEVO_FICHERO,
277                         null);
278                 break;
279 
280             case ACCION_ANYADIR_TAREA:
281                 String[] datosAnyadirTarea = anyadirTarea();
282                 if (datosAnyadirTarea != null) {
283                     Tupla<Tupla, Tupla> tupla = new Tupla(
284                             new Tupla(datosAnyadirTarea[0], datosAnyadirTarea[1]),
285                             new Tupla(datosAnyadirTarea[2], datosAnyadirTarea[3
    ]));
286                     oyenteVista.eventoProducido(
287                             OyenteVista.Evento.ANYADIR_TAREA, tupla);
288                 }
289                 break;
290 
291             case ACCION_BUSCAR_TAREAS:
292                 String[] datosBuscarTareas = buscarTareas();
293                 if (datosBuscarTareas != null) {
294                     oyenteVista.eventoProducido(OyenteVista.Evento.BUSCAR_TAREA,
295                             datosBuscarTareas[0]);
296                 }
297                 break;
298 
299             case ACCION_SALIR:
300                 int resp = JOptionPane.showConfirmDialog(null,
301                         MENSAJE_GUARDAR_CAMBIOS,
302                         TITULO_GUARDAR_SALIR, JOptionPane.YES_NO_OPTION);
303                 if (resp == JOptionPane.YES_OPTION) {
304                     oyenteVista.eventoProducido(OyenteVista.Evento.SALIR,
305                             RESPUESTA_SI);
306                 } else if (resp == JOptionPane.NO_OPTION) {
307                     oyenteVista.eventoProducido(OyenteVista.Evento.SALIR,
308                             RESPUESTA_NO);
309                 }
310                 break;
311 
312             case ACCION_ELIMINAR_TAREA:
313                 String[] datosEliminarTarea = obtenerDatosEliminarTarea();
314                 if (datosEliminarTarea != null) {
315                     oyenteVista.eventoProducido(
316                             OyenteVista.Evento.ELIMINAR_TAREA,
317                             datosEliminarTarea[0]);
318                 }
319                 break;
320 
321             case ACCION_LISTAR_TAREAS:
322                 oyenteVista.eventoProducido(OyenteVista.Evento.LISTAR_TAREAS,
323                         null);
324                 break;
325 
326             case ACCION_GUARDAR_TAREAS:
327                 oyenteVista.eventoProducido(OyenteVista.Evento.GUARDAR_TAREAS,
328                         null);
329                 break;
```

```
330         }
331     }
332 
333     /**
334      * Método para mostrar tareas en un mainframe.
335      *
336      * @param tareas
337      */
338     public void mostrarTareas(String tareas) {
339         areaTextoTareas.setText("");
340         areaTextoTareas.setText(tareas);
341     }
342 
343     /**
344      * Método para iniciar sesión en un mainframe.
345      *
346      * @return
347      */
348     private String[] iniciarSesion() {
349         ComplexDialoguePanel ventanaIniciarSesion =
350                 new ComplexDialoguePanel(TITULO_VENTANA_INICIAR_SESION,
351                         CAMPOS_INICIAR_SESION, TAMANYO_CAMPOS_INICIO_SESION);
352         String[] datosInicioSesion =
353                 ventanaIniciarSesion.obtenerTextoCampos(
354                         OPCIONES_INICIAR_SESION);
355 
356         return datosInicioSesion;
357     }
358 
359     /**
360      * Método para añadir una tarea en el mainframe.
361      *
362      * @return
363      */
364     private String[] anyadirTarea() {
365         String[] datosAnyadirTarea = null;
366 
367         ComplexDialoguePanel ventanaAnyadirTarea = new ComplexDialoguePanel(
368                 TITULO_VENTANA_ANYADIR_TAREA, CAMPOS_ANYADIR_TAREA,
369                 TAMANYO_DESCRIPCION_TAREA);
370         datosAnyadirTarea = ventanaAnyadirTarea.obtenerTextoCampos(
371                 OPCIONES_ANYADIR_TAREA);
372 
373         CodigoRespuesta codigo = verificarDatosTarea(datosAnyadirTarea);
374 
375         switch (codigo) {
376             case OK:
377                 return datosAnyadirTarea;
378 
379             case ERROR_FECHA_INCORRECTA:
380                 JOptionPane.showMessageDialog(new JFrame(),
381                         MENSAJE_FECHA_INCORRECTA, ETIQUETA_VENTANA_ERROR_FECHA,
382                         JOptionPane.ERROR_MESSAGE);
383                 break;
384 
```

```
385             case ERROR_ID_TAREA:
386                 JOptionPane.showMessageDialog(new JFrame(),
387                         MENSAJE_ID_INCORRECTO, ETIQUETA_VENTANA_ID_INCORRECTO,
388                         JOptionPane.ERROR_MESSAGE);
389                 break;
390 
391             case ERROR_LONG_NOMBRE:
392                 JOptionPane.showMessageDialog(new JFrame(),
393                         MENSAJE_NOMBRE_INCORRECTO,
394                         ETIQUETA_VENTANA_NOMBRE_INCORRECTO,
395                         JOptionPane.ERROR_MESSAGE);
396                 break;
397 
398             case ERROR_LONG_DESCRIPCION:
399                 JOptionPane.showMessageDialog(new JFrame(),
400                         MENSAJE_DESCRIPCION_INCORRECTA,
401                         ETIQUETA_VENTANA_DESCRIPCION_INCORRECTA,
402                         JOptionPane.ERROR_MESSAGE);
403                 break;
404         }
405         return null;
406     }
407 
408     /**
409      * Comprueba los datos que el usuario ha introducido.
410      *
411      * @param datosTarea
412      * @return
413      */
414     private CodigoRespuesta verificarDatosTarea(String[] datosTarea) {
415         if (datosTarea != null) {
416             if (datosTarea[0].matches(PATRON_CADENA_DIGITOS)) {
417                 if ((datosTarea[1].length() > 0) &&
418                         (datosTarea[1].length() <= TAMANYO_NOMBRE_TAREA)) {
419                     if ((datosTarea[2].length() > 0) &&
420                             (datosTarea[2].length() <=
421                                     TAMANYO_DESCRIPCION_TAREA)) {
422                         if (datosTarea[3].matches(FORMATO_FECHA)) {
423                             return CodigoRespuesta.OK;
424                         } else {
425                             return CodigoRespuesta.ERROR_FECHA_INCORRECTA;
426                         }
427                     }
428                     return CodigoRespuesta.ERROR_LONG_DESCRIPCION;
429                 } else {
430                     return CodigoRespuesta.ERROR_LONG_NOMBRE;
431                 }
432             } else {
433                 return CodigoRespuesta.ERROR_ID_TAREA;
434             }
435         } else {
436             return CodigoRespuesta.DATOS_VACIOS;
437         }
438     }
439 
```

```
440     /**
441      * Método para buscar tareas en el mainframe por fecha.
442      *
443      * @return
444      */
445     private String[] buscarTareas() {
446         ComplexDialoguePanel ventanaBuscarTareas =
447                 new ComplexDialoguePanel(TITULO_VENTANA_BUSCAR_TAREAS,
448                         CAMPOS_BUSCAR_TAREAS, TAMANYO_CAMPOS_TEXTO);
449         String[] datosBuscarTareas =
450                 ventanaBuscarTareas.obtenerTextoCampos(OPCIONES_BUSCAR_TAREAS);
451 
452         return datosBuscarTareas;
453     }
454 
455     /**
456      * Método para eliminar una tarea.
457      *
458      * @return
459      */
460     private String[] obtenerDatosEliminarTarea() {
461         ComplexDialoguePanel ventanaEliminarTareas =
462                 new ComplexDialoguePanel(TITULO_VENTANA_ELIMINAR_TAREA,
463                         CAMPOS_ELIMINAR_TAREA, TAMANYO_CAMPOS_TEXTO);
464         String[] datoseliminarTareas =
465                 ventanaEliminarTareas.obtenerTextoCampos(
466                         OPCIONES_ELIMINAR_TAREA);
467 
468         return datoseliminarTareas;
469     }
470 
471     /**
472      * Muestra un diálogo con el error que se ha producido.
473      *
474      * @param titulo
475      * @param mensaje
476      */
477     public void notificarMensajeError(String titulo, String mensaje) {
478         JOptionPane.showMessageDialog(new JFrame(), mensaje, titulo,
479                 JOptionPane.ERROR_MESSAGE);
480     }
481 
482     /**
483      * Muestra un diálogo con la confirmación de la tarea.
484      *
485      * @param titulo
486      * @param mensaje
487      */
488     public void notificarMensajeConfirmacion(String titulo, String mensaje) {
489         JOptionPane.showMessageDialog(new JFrame(), mensaje, titulo,
490                 JOptionPane.INFORMATION_MESSAGE);
491     }
492 
493     /**
494      * Método de cambios de propiedades de los elementos de la ventana.
```

```
495       *
496       * @param evt
497       */
498      @Override
499      public void propertyChange(PropertyChangeEvent evt) {
500      }
501  }
502
503
```