```java
/**
 * ComplexDialoguePanel.java
 * <p>
 * Diálogo dinámico emergente con campos de texto.
 * <p>
 * Radu Constantin Robu y Alberto Pérez
 */
package P3.Vista;

import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.util.HashMap;
import java.util.Map;
import javax.swing.*;

/**
 * Clase que permite crear una ventana para introducir varios datos
 * por teclado.
 */
public class ComplexDialoguePanel extends JPanel {
    private static final String ETIQUETA_VENTANA = "Introducir información";
    private static final String ETIQUETA_CONTRASEÑA = "Contraseña";
    private String[] etiquetasDatos;
    private Map<String, JTextField> camposTexto = new HashMap<>();

    /**
     * Construye un ComplexDialogPanel.
     *
     * @param textoVentana
     * @param etiquetasDatos
     * @param tamanyoCamposTexto
     */
    public ComplexDialoguePanel(String textoVentana, String[] etiquetasDatos,
                                int tamanyoCamposTexto) {
        setLayout(new GridBagLayout());
        inicializar(etiquetasDatos, tamanyoCamposTexto);
        setBorder(BorderFactory.createTitledBorder(textoVentana));
        this.etiquetasDatos = etiquetasDatos;
    }

    /**
     * Inicializa el ComplexDialoguePanel.
     *
     * @param etiquetasDatos
     * @param tamanyo
     */
    private void inicializar(String[] etiquetasDatos, int tamanyo) {
        for (int i = 0; i < etiquetasDatos.length; i++) {
            String etiqueta = etiquetasDatos[i];
            add(new JLabel(etiqueta), createGbc(0, i));

            if (etiqueta != ETIQUETA_CONTRASEÑA) {
                JTextField textField = new JTextField(tamanyo);
                camposTexto.put(etiqueta, textField);
```

```
56                  add(textField, createGbc(1, i));
57              } else if (etiqueta == ETIQUETA_CONTRASEÑA) {
58                  JPasswordField passwordField = new JPasswordField(tamanyo);
59                  camposTexto.put(etiqueta, passwordField);
60                  add(passwordField, createGbc(1, i));
61              }
62          }
63      }
64
65      /**
66       * Devuelve el camposTexto introducido en el campo correspondiente
67       * a la etiqueta.
68       *
69       * @param etiqueta
70       * @return
71       */
72      public String getText(String etiqueta) {
73          JTextField campoTexto = camposTexto.get(etiqueta);
74          if (campoTexto != null) {
75              return campoTexto.getText();
76          } else {
77              throw new IllegalArgumentException(etiqueta);
78          }
79      }
80
81      /**
82       * Crea las dimendiones del GridBagLayout.
83       *
84       * @param x
85       * @param y
86       * @return
87       */
88      public static GridBagConstraints createGbc(int x, int y) {
89          GridBagConstraints gbc = new GridBagConstraints();
90          gbc.gridx = x;
91          gbc.gridy = y;
92          gbc.weightx = 1.0;
93          gbc.weighty = gbc.weightx;
94          if (x == 0) {
95              gbc.anchor = GridBagConstraints.LINE_START;
96              gbc.fill = GridBagConstraints.BOTH;
97              gbc.insets = new Insets(5, 5, 5, 8);
98          } else {
99              gbc.anchor = GridBagConstraints.LINE_END;
100             gbc.fill = GridBagConstraints.HORIZONTAL;
101             gbc.insets = new Insets(5, 5, 5, 5);
102         }
103         return gbc;
104     }
105
106     /**
107      * Obtiene los datos del viajero a través de un ComplexDialoguePanel.
108      *
109      * @param opciones
110      * @return
```

```
111      */
112     public String[] obtenerTextoCampos(String[] opciones) {
113         int respuesta = -1;
114         int numDatos = this.etiquetasDatos.length;
115         boolean datosIncorrectos = true;
116         while (datosIncorrectos) {
117             String[] datos = new String[numDatos];
118 
119             int tipoOpcion = JOptionPane.DEFAULT_OPTION;
120             int tipoMensaje = JOptionPane.PLAIN_MESSAGE;
121             Icon icono = null;
122             Object valorInicial = opciones[0];
123 
124             respuesta = JOptionPane.showOptionDialog(null,
125                     this, ETIQUETA_VENTANA,
126                     tipoOpcion, tipoMensaje, icono,
127                     opciones, valorInicial);
128 
129             if (respuesta == 0) {
130                 for (int i = 0; i < numDatos; i++) {
131                     datos[i] = this.getText(this.etiquetasDatos[i]);
132                 }
133 
134                 datosIncorrectos = comprobarDatosIntroducidos(datos);
135 
136                 if (!datosIncorrectos) {
137                     return datos;
138                 }
139             } else {
140                 return null;
141             }
142         }
143         return null;
144     }
145 
146     /**
147      * Método que devuelve si los datos introducidos son correctos o no.
148      *
149      * @param datos
150      * @return
151      */
152     private boolean comprobarDatosIntroducidos(String[] datos) {
153         int datosValidos = 0;
154         boolean datosIncorrectos = true;
155 
156         for (String dato : datos) {
157             if ((dato != null) && (dato.length() > 0)) {
158                 datosValidos++;
159             }
160         }
161 
162         if (datosValidos == datos.length) {
163             datosIncorrectos = false;
164         }
165 
```

```
166         return datosIncorrectos;
167     }
168 }
```