

Estructuras de Datos

Proyecto Primera Evaluación – 2023 PAO 2
Sistema para Gestión de Contactos

Integrantes:

Kevin Gianfranco Ibarra Prado
Luis Alejandro Barrera Muñoz
Luis Adrián Litardo Calderón

Ventana de Gestión de Contactos



Ilustración 1 -Imagen referencial de la pantalla de gestión de contactos

En esta ventana (ilustración 1) llamada gestión de contactos, se muestra la pantalla principal o de inicio de nuestra aplicación.

Usuario: en esta caja de texto colocaremos el usuario con los contactos que se desea visualizar. Cada usuario será validado si se encuentra registrado, si lo está mostrará un registro de contactos único, para cada usuario que registre sus contactos.

Log in: dicho botón, dará inicio a la lista contactos del usuario indicado en la parte superior del mismo, siempre que se encuentre registrado.

Register: dicho botón, creará un documento .txt con el nombre indicado en el cuadro de texto del usuario. En este documento se guardarán todos los contactos creados por dicho usuario para que posteriormente pueda visualizarlos, editarlos o eliminarlos. Cada usuario registrado es único y los contactos registrados en el son privados, ningún otro usuario sin acceso a él podrá interactuar con ellos.

Ventana de Lista de Contactos

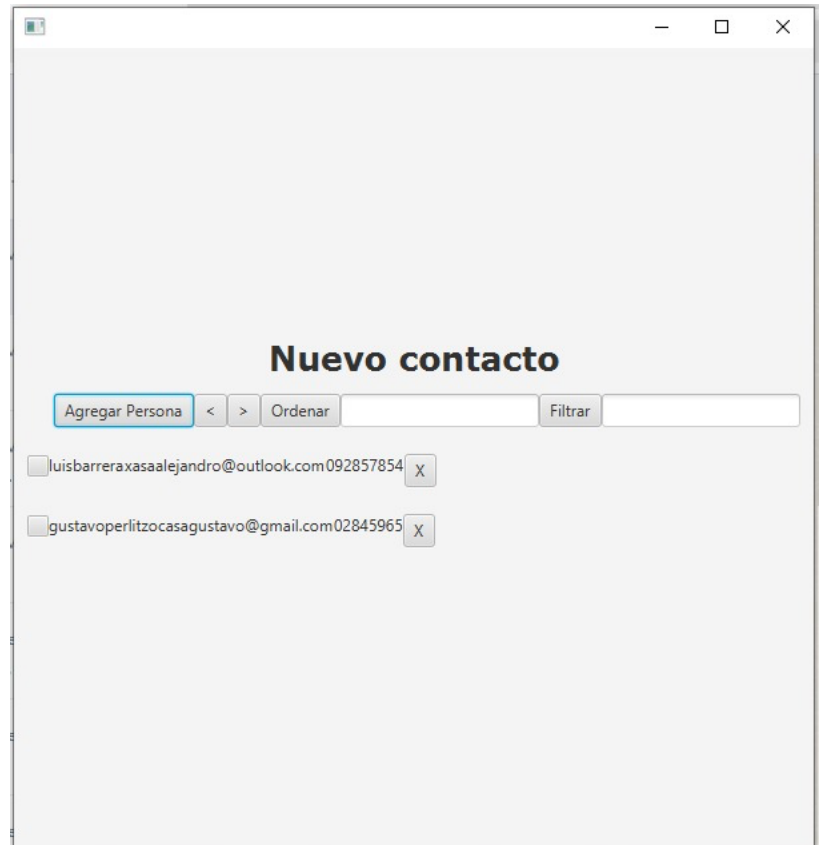
En esta ventana (ilustración 2) se muestran la lista de todos los contactos agregados por el usuario, los cuales se pueden visualizar de manera más detallada al pulsar el nombre de cada contacto. El cual desplegará otra ventana (ilustración 3) en la que el usuario podrá modificar la información que requiera del actual contacto preseleccionado.

Agregar persona: dicho botón, despliega la ventana de nuevo contacto (ilustración 3) en la cual podremos agregar todos los datos de una nueva persona o empresa la cual queramos agregar a la lista de contactos.

< / > : dichos botones, servirán para recorrer la lista de los contactos creados por el usuario. Dentro de la funcionalidad de esta, se realiza un recorrido a una LinkedList personalizada de Objetos <Contacto> la cual es circular doblemente enlazada. Por la cual una vez se culmina de mostrar el último contacto presentado en la lista, nuevamente se presenta el primer contacto, siendo alusión a una lista de contactos circular.

Ordenar: dicho botón ordenará la lista de manera específica y detallada de acuerdo al argumento presentado en la caja de texto que se encuentra contigua a la misma.

Filtrar: dicho botón, filtrará una busca de contactos según el argumento presentado en la caja de texto que se encuentra contigua a la misma. Las filtraciones pueden ser por categorías como nombre, apellido, correo, etc.



Ventana de Agregar Contactos

En esta ventana (ilustración 2) se muestran la opción de agregar contactos, con todos sus requerimientos y validaciones.

Nuevo contacto

Nombre: luis

Apellido: barrera

Dirección: xasa

Email: alejandro@outlook.com

Numero telefónico: 092857854

Agregar Foto

< >

X

Brasil

Agregar

```
NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help 92 % Mar 5 dic 10:49
<default-config> 386.0/1664MB Search (⌘+)
Start Page x App.java x VistaContactos.java x
Source History
1 package proyecto.mavenproject1;
2
3
4 import datos.CONSTANTES;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.io.InputStream;
8 import java.io.ObjectOutputStream;
9 import java.util.ListIterator;
10 import javafx.application.Application;
11 import javafx.geometry.Insets;
12 import javafx.geometry.Pos;
13 import javafx.scene.Scene;
14 import javafx.scene.control.Alert;
15 import javafx.scene.control.Button;
16 import javafx.scene.control.Label;
17 import javafx.scene.control.TextField;
18 import javafx.scene.image.Image;
19 import javafx.scene.image.ImageView;
20 import javafx.scene.layout.GridPane;
21 import javafx.scene.layout.VBox;
22 import javafx.scene.text.Font;
23 import javafx.scene.text.FontWeight;
24 import javafx.stage.Stage;
25
26 /**
27  * JavaFX App
28  */
29 public class App extends Application {
30
31     public Stage stage;
32     TextField txtusuario;
33     VBox root = new VBox();
34     VBox lineaBotones = new VBox();
35     GridPane datos = new GridPane();
36     Label label = new Label(string: "Gestión de contactos");
37     private ImageView imageView;
38
39     @Override
40     public void start(Stage stage) {
41         Font fuente = Font.font(string: "Verdana", fw: FontWeight.EXTRA_BOLD, d: 25);
42     }
43 }
```

```
NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help 92 % Mar 5 dic 10:49
<default-config> 588.9/1664MB Search (⌘+)
Start Page x App.java x VistaContactos.java x
Source History
37
38 @Override
39 public void start(Stage stage) {
40     Font fuente = Font.font(string: "Verdana", fw: FontWeight.EXTRA_BOLD, d: 25);
41     label.setFont(font: fuente);
42     label.setAlignment(pos: Pos.TOP_CENTER);
43
44     //Creacion Botones
45     Button btnlog = new Button(string: "Log In");
46     Button btnreg = new Button(string: "Register");
47
48     datos.setVgap(d: 10); // Espaciado vertical entre filas
49     datos.setHgap(d: 10); // Espaciado horizontal entre columnas
50     datos.setAlignment(pos: Pos.CENTER);
51     lineaBotones.setAlignment(pos: Pos.CENTER);
52
53     // Configurar márgenes para los elementos (opcional)
54     lineaBotones.setSpacing(d: 20); // 20 píxeles de espacio vertical
55
56     GridPane.setMargin(node: btnlog, new Insets(d: 20));
57     GridPane.setMargin(node: btnreg, new Insets(d: 20));
58
59     Label lblusuario = new Label(string: "Usuario: ");
60     txtusuario = new TextField();
61
62     datos.add(node: lblusuario, i: 0, i1: 0);
63     datos.add(node: txtusuario, i: 1, i1: 0);
64     try {
65         Image imagen = new Image(string: "file:src/main/java/datos/contactos.jpg", d: 200, d1: 200, bin: true, bin1: true);
66         imageView = new ImageView(image: imagen);
67     } catch (Exception exception) {
68         imageView = new ImageView();
69     }
70
71     lineaBotones.getChildren().addAll(es: btnlog, es: btnreg);
72     root.setSpacing(d: 40);
73     root.setAlignment(pos: Pos.CENTER);
74     root.getChildren().add(c: label);
75     root.getChildren().add(c: imageView);
76 }
```

```
NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help 92 % Mar 5 dic 10:50
<default-config> 178.7/1664MB
Start Page x App.java x VistaContactos.java x
Source History
81 Scene scene = new Scene(parent: root, d: 640, d1: 480);
82 stage.setTitle(string: "Gestión de Contactos");
83 stage.setScene(scene);
84 stage.show();
85
86 }
87
88 public static void main(String[] args) {
89     launch 0;
90 }
91
92 public void nextImage(List<Image> it) {
93
94     // FileChooser f = new FileChooser();
95     // f.setInitialDirectory(new File("C:\\Users\\LuisA\\OneDrive\\Escritorio\\images"));
96     // File selectedFile = f.showOpenDialog(stage);
97     //
98     // Path from = Paths.get(selectedFile.toURI());
99     // System.out.println(selectedFile.toURI());
100     // System.out.println(Paths.get("C:\\Users\\LuisA\\OneDrive\\Escritorio\\images"));
101     // Path to = Paths.get("C:\\Users\\LuisA\\OneDrive\\Escritorio\\images");
102     //
103     // selectedFile.renameTo(new File("C:\\Users\\LuisA\\OneDrive\\Escritorio\\images\\newjpg.jpg"));
104 }
105
106 public void mostrarContactos() {
107     try {
108         VistaContactos ventanaContactos = new VistaContactos(archivo: txtusuario.getText());
109         ventanaContactos.start(new Stage());
110     } catch (Exception ex) {
111         ex.printStackTrace();
112     }
113 }
114
115 private void registrarUsuario() {
116     // Obtiene el código ingresado en el campo txtcodigo
117     String usuario = txtusuario.getText();
118
119     // Verifica que el código no esté vacío
120 }
```

```
NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help 92 % Mar 5 dic 10:50
<default-config> 372.7/614.0MB
Start Page x App.java x VistaContactos.java x
Source History
113 }
114
115 private void registrarUsuario() {
116     // Obtiene el código ingresado en el campo txtcodigo
117     String usuario = txtusuario.getText();
118
119     // Verifica que el código no esté vacío
120     if (usuario.isEmpty()) {
121         // Define un nombre de archivo basado en el código
122         Usuario user = new Usuario(codigo: usuario);
123         String archivo = "U" + usuario + ".txt";
124         try {
125             // Abre un archivo para escribir el objeto
126             ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(name: archivo));
127             // Escribe el código en el archivo
128             oos.writeObject(obj: user);
129             oos.flush();
130             oos.close();
131             // Muestra una alerta de éxito
132             mostrarAlerta(titulo: "Éxito", mensaje: "El código se ha grabado en el archivo: " + archivo, tipo: Alert.AlertType.INFORMATION);
133         } catch (IOException ex) {
134             // Muestra una alerta en caso de error
135             mostrarAlerta(titulo: "Error", mensaje: "Hubo un error al grabar el código en el archivo.", tipo: Alert.AlertType.ERROR);
136         }
137     } else {
138         // Muestra una alerta si el campo está vacío
139         mostrarAlerta(titulo: "Error", mensaje: "El campo de código está vacío.", tipo: Alert.AlertType.ERROR);
140     }
141 }
142
143 // Método auxiliar para mostrar una alerta
144 private void mostrarAlerta(String titulo, String mensaje, Alert.AlertType tipo) {
145     Alert alert = new Alert(tipo);
146     alert.setTitle(titulo);
147     alert.setContentText(mensaje);
148     alert.showAndWait();
149 }
150
151 }
152 }
```

Tabla de Co-Evaluación de equipo de proyecto

Los valores de la siguiente Coevaluación van clasificados de la siguiente manera:

1 – No aporte en nada, 2 – Aporte insuficiente, 3 – Aporte lo mínimo, 4 – Un aporte significativo, 5- Aporte completo y eficiente.

Criterios de evaluación	KEVIN GIANFRANCO IBARRA PRADO	LUIS ALEJANDRO BARRERA MUÑOZ	LUIS ADRIAN LITARDO CALDERON
<u>Participación</u>	5	5	5
<u>Responsabilidad</u>	5	5	5
<u>Colaboración</u>	5	5	5
<u>Calidad de trabajo</u>	5	5	5
<u>Comunicación</u>	5	5	5
<u>Aportes al equipo</u>	5	5	5

1 Tabla de Co-evaluación