

Universidad Rafael Landívar
Facultad de Ingeniería
Manejo e implementación de Archivos
Sección: 01
Catedrática: Ing. David Luna



Comunidad de reciclaje
Programa de administración de datos

Gálvez Arriaza, Andres Sebastian [1024718]
Villatoro Muñoz, Alexander Gabriel [1182118]
López Monserratt, Mauricio Alejandro [1270818]

Guatemala, 18 de noviembre de 2021

Requerimientos mínimos

Requerimientos de hardware

- RAM con al menos 4GB de capacidad.
- Al menos 2GB de almacenamiento disponibles.

Requerimientos de software

- Apache NetBeans IDE +12.5.
- JDK Java Development Kit +16.
- Windows +8.

Alcance del proyecto

El programa de administración de datos consta de un inicio de sesión si ya se tiene un usuario o de un registro si es la primera vez que se ingresan datos. El administrador puede hacer backup de los usuarios seleccionando la ruta destino y, de igual forma, puede editar información de los mismos. También puede agregar, modificar o eliminar materiales, los cuales se almacenan de forma optimizada empleando un árbol binario. Así mismo, es capaz de generar reportes de los materiales donados por los demás usuarios y visualizarlos en la aplicación. Un usuario estándar solo es capaz de editar sus propios datos, así como agregar donaciones.

La ruta en el equipo para encontrar los archivos empleados es C:/MEIA/. En esta es posible encontrar una recopilación de archivos de texto, los cuales contienen cada registro de datos empleados en el programa. El almacenamiento de usuarios se realiza con la implementación de un archivo de tipo secuencial, en este se incluyen a usuarios de rol administrador y rol estándar. Además, la contraseña de los usuarios se almacena cifrada, con el fin de no comprometer la seguridad de los datos.

Estructuras, clases y métodos

- Clases

AdminDashboard	
Atributos	
+ String[] currentUser	
Métodos	
	<pre>1 private void initComponents() { ...164 lines } // </editor-fold> 2 3 private void backupBtnActionPerformed(java.awt.event.ActionEvent evt) { ...22 lines } 4 5 private void userFieldActionPerformed(java.awt.event.ActionEvent evt) { ...3 lines } 6 7 private void searchBtnActionPerformed(java.awt.event.ActionEvent evt) { ...61 lines } 8 9 private void formMouseMoved(java.awt.event.MouseEvent evt) { ...4 lines } 10 11 private void banUserActionPerformed(java.awt.event.ActionEvent evt) { ...61 lines } 12 13 private void registerBtnActionPerformed(java.awt.event.ActionEvent evt) { ...5 lines } 14 15 private void MaterialesBtnActionPerformed(java.awt.event.ActionEvent evt) { ...5 lines } 16 17 private void DonacionesBtnActionPerformed(java.awt.event.ActionEvent evt) { ...5 lines } 18 19 public void MakeBackup(String path) 20 { ...9 lines } 21 22 public static void getFiles(File original, File copy) 23 { ...16 lines } 24 25 public static void getDirectorys(File original, File copy) 26 { ...17 lines }</pre>

Archivo

Atributos

- + Integer BloqueActual
- + String Administrador
- + Integer MaxReorganizar

Métodos

```
41  □ private String HayEspacio(String PathArchivo, boolean Retornar, String Usuario) throws IOException {...53 lines }
94
95  □ private void ActualizarLinea(String ParaActualizar, String Cambio, String PathArchivo) throws IOException {...14 lines }
109 □ private String ObtenerLinea(Integer n, String PathArchivo) {...18 lines }
127
128 □ private Integer ObtenerNumeroLinea(String Linea2, String archivoMaestro) throws IOException {...16 lines }
144
145 □ private Integer ObtenerCantidadLineas(String archivoMaestro) throws IOException {...12 lines }
157
158 □ private void OrdenarIndice(String archivo) throws IOException {...75 lines }
233
234 □ public void Inicializar(String Administrador) throws IOException {...56 lines }
290
291 □ public void Insertar(String Usuario, String Fecha, String Nombre_Material, Integer Peso, String Descripcion,
292   String Evento, String Usuario_transaccion) throws IOException {...45 lines }
337
338 □ public List<String> Buscar(String Usuario) throws IOException {...38 lines }
376
377 □ public List<String> Buscar(String Usuario, String NombreMaterial) throws IOException {...39 lines }
416
417 □ public List<String> Buscar(String Usuario, String NombreMaterial, String Fecha) throws IOException {...40 lines }
457
458 □ private List<String> BuscarIndice(String Usuario, String NombreMaterial, String Fecha) throws IOException {...35 lines }
493
494 □ public List<String> BuscarSecuencial(String Campo) throws IOException
495   {...32 lines }
527
528 □ public Boolean Eliminar(String Usuario, String NombreMaterial, String Fecha) throws IOException
529   {...40 lines }
569
570 // 0 = Usuario || 1 = NombreMaterial || 2 = Fecha || 3 = Peso || 4 = Descripcion || 5 = Evento || 6 = UsuarioTransaccion
571 // El estatus no lo modifiko porque para eso esta eliminar
572 □ public boolean Actualizar(String Usuario, String NombreMaterial, String Fecha, Integer NCampo, String NuevoCampo) throws IOException
573   {...47 lines }
620
621 □ private List<String> FiltroPorFecha(String Fecha1, String Fecha2) throws IOException
622   {...32 lines }
654
655 □ public class Donacion {...26 lines }
681 }
```

Donaciones

Atributos

- + String UsuarioBuscado
- + String NameBuscado
- + String FechaBuscado
- + String[] Buscado

Métodos

```
189  □ private void SearchBtnActionPerformed(java.awt.event.ActionEvent evt) {...65 lines }
255  □ private void EditBtnActionPerformed(java.awt.event.ActionEvent evt) {...29 lines }
285  □ private void EliminarBtnActionPerformed(java.awt.event.ActionEvent evt) {...26 lines }
312
```

Login	
Atributos	
Métodos	
15	<code>private void exitBtnActionPerformed(java.awt.event.ActionEvent evt) { ...5 lines }</code>
19	<code>private void loginBtnActionPerformed(java.awt.event.ActionEvent evt) { ...116 lines }</code>
16	<code>private void registerBtnActionPerformed(java.awt.event.ActionEvent evt) { ...19 lines }</code>
17	<code>private void passFieldKeyPressed(java.awt.event.KeyEvent evt) { ...7 lines }</code>
14	<code>private void loginBtnKeyPressed(java.awt.event.KeyEvent evt) { ...7 lines }</code>
15	

Materiales	
Atributos	
Métodos	
	<code>private void formWindowOpened(java.awt.event.WindowEvent evt) { ...9 lines }</code>
	<code>private void IngresarBtnActionPerformed(java.awt.event.ActionEvent evt) { ...64 lines }</code>
	<code>private void SearchBtnActionPerformed(java.awt.event.ActionEvent evt) { ...18 lines }</code>
	<code>private void EliminiBtnActionPerformed(java.awt.event.ActionEvent evt) { ...53 lines }</code>
	<code>private void EditBtnActionPerformed(java.awt.event.ActionEvent evt) { ...68 lines }</code>
	<code>private void ImagBtnActionPerformed(java.awt.event.ActionEvent evt) { ...13 lines }</code>
	<code>public void Retornar_vacio () { ...8 lines }</code>
	<code>public String Buscar(String Usuario, String NombreMaterial) { ...26 lines }</code>

RecycleDashboard	
Atributos	
+ String User	
Métodos	
1	<code>private void PesoFieldKeyPressed(java.awt.event.KeyEvent evt) { ...3 lines }</code>
2	<code>private void formWindowOpened(java.awt.event.WindowEvent evt) { ...10 lines }</code>
3	<code>private void RecyBtnActionPerformed(java.awt.event.ActionEvent evt) { ...15 lines }</code>
4	<code>private void FechaFieldActionPerformed(java.awt.event.ActionEvent evt) { ...3 lines }</code>
5	<code>public String ObtenerAdmin() { ...45 lines }</code>

Register	
Atributos	
Métodos	
+	<code>private void passFieldKeyPressed(java.awt.event.KeyEvent evt) {...61 lines }</code>
	<code>//Used to register users.</code>
+	<code>private void registerBtnActionPerformed(java.awt.event.ActionEvent evt) {...229 lines }</code>
	<code>//Used to select an image.</code>
+	<code>private void photoBtnActionPerformed(java.awt.event.ActionEvent evt) {...13 lines }</code>
+	<code>private void formWindowClosing(java.awt.event.WindowEvent evt) {...3 lines }</code>
+	<code>private void exitBtnActionPerformed(java.awt.event.ActionEvent evt) {...5 lines }</code>
	<code>//Used to get scores of password security.</code>
+	<code>public String[] getScores() {...34 lines }</code>
	<code>//Used to get results of password security range.</code>
+	<code>public String[] getResults() {...34 lines }</code>
	<code>//Used to cipher password with AES.</code>
+	<code>public String cipherPass(String pass, String ENCRYPT_KEY) {...23 lines }</code>
	<code>//Pasar bitacora a usuarios</code>
+	<code>public String ObtenerAdmin() {...45 lines }</code>
	<code>//Pasar bitacora a usuarios</code>
+	<code>public void BitToUser(String Usuario_Modificacion) {...135 lines }</code>
	<code>//Retorna contenido no completado</code>
+	<code>public String[] ReturnContent(String Archivo, int length) {...26 lines }</code>
	<code>//Pasar bitacora a usuarios</code>
+	<code>public String fecha_modificacion(String Archivo) {...22 lines }</code>
	<code>//Pasar bitacora a usuarios</code>
+	<code>public String usuario_modificacion(String Archivo) {...22 lines }</code>
	<code>//Pasar bitacora a usuarios</code>
+	<code>public String Registros_totales() {...56 lines }</code>

UserDashboard	
Atributos	
+ String[] currentUser	
Métodos	
117	<code>private void banBtnActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }</code>
121	<code>private void RecyBtnActionPerformed(java.awt.event.ActionEvent evt) {...5 lines }</code>

- Métodos

- Insertar: está la acción de insertar el usuario, nombre_material, fecha, peso, descripción, evento, usuario_transacción, fecha_creacion, estatus.
- Búsqueda: se realiza 3 tipos de búsqueda, la primera se realiza con el usuario, la segunda por el usuario y el nombre del material, y por último se realiza por medio del usuario, nombre del material y la fecha. Si no existe una llave se hace una búsqueda secuencial por medio de cualquier campo.
- Eliminación: en este método por medio del usuario, nombre material y la fecha se hace la eliminación de cada archivo.
- Modificación: se manda el usuario, nombre material, fecha, número del campo para poder saber qué campo se tiene que modificar y por último el nuevo campo o lo que se va a estar modificando.
- Reporte si no se tiene ningún parámetro, va a agarrar un rango de fecha default.
- Reporte con parámetros de fecha se tendrá un rango de estos, en los cuales se van a poder demostrar el reporte que esté dentro de ese rango.