Universidad Rafael Landivar 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.

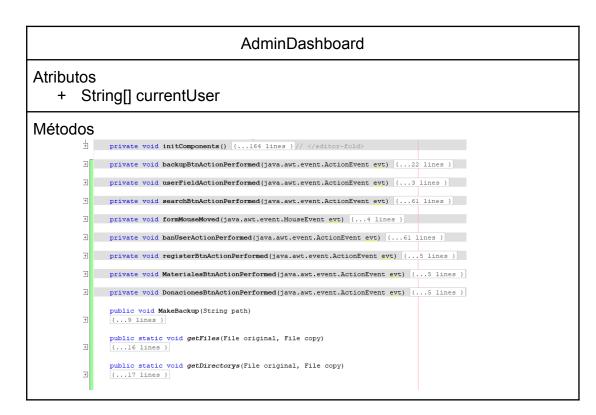
## Alcance del proyecto

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 aplicaicón. 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



## Archivo

### **Atributos**

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

#### Métodos 41 + private String HayEspacio(String PathArchivo, boolean Retornar, String Usuario) throws IOException (...53 lines } private void ActualizarLinea(String ParaActualizar, String Cambio, String PathArchivo) throws IOException [...14 lines] private String ObtenerLinea(Integer n, String PathArchivo) [...18 lines] 128 🛨 private Integer ObtenerNumeroLinea(String Linea2, String archivoMaestro) throws IOException [...16 lines } private Integer ObtenerCantidadLineas(String archivoMaestro) throws IOExcept ion [...12 lines ] 158 🛨 private void OrdenarIndice(String archivo) throws IOException {...75 lines } 234 🛨 public void Inicializar(String Administrador) throws IOException {...56 lines } public void Insertar(String Usuario, String Fecha, String Nombre Material, Integer Feso, String Descripcion, 291 292 🛨 String Evento, String Usuario\_transaccion) throws IOException {...45 lines 337 338 ± public List<String> Buscar(String Usuario) throws IOException {...38 lines } 377 + 416 public List<String> Buscar(String Usuario, String NombreMaterial) throws IOException [...39 lines ] 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}] public List<String> BuscarSecuencial(String Campo) throws IOException [{...32 lines }] 494 495 ± 528 public Boolean Eliminar(String Usuario, String NombreMaterial, String Fecha) [...40 lines ] 529 ± 569 // 0 = Usuario || 1 = NombreMaterial || 2 = Fecha || 3 = Peso || 4 = Descripcion || 5 = Evento || 6 = UsuarioTransaccion |/ El estatus no lo modifico porque para eso esta eliminar public boolean Actualizar(String Usuario, String NombreMaterial, String Fecha, Integer NCampo, String NuevoCampo) throws IOException [...47 lines ] 573 ± 620 <u>Q.</u> 622 ± private List<String> FiltroPorFecha(String Fechal, String Fechal) throws IOException 654 655 + public class Donacion [...26 lines ]

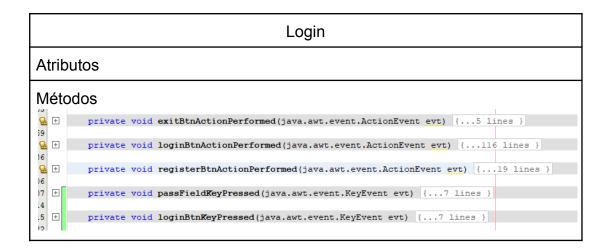
## **Donaciones**

## **Atributos**

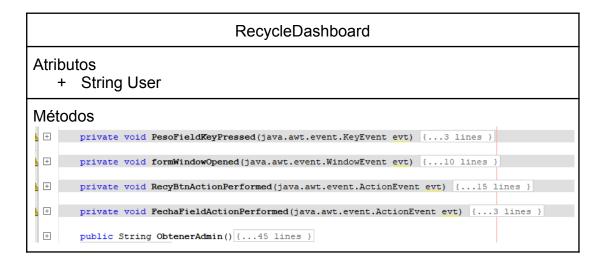
- String UsuarioBuscado
- + String NameBuscado
- String FechaBuscado
- + String[] Buscado

## Métodos









# Register

## **Atributos**

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

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

### 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 esté 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.