

Universidad Rafael Landívar

Facultad de Ingeniería

Ingeniería Informática y Sistemas

Manejo e Implementación de Archivos

Ing. David Luna

## **PROYECTO MEIA (FASE I)**

Erick Fabián Alvarez Franco      Carnet: 1070117

Erick Eduardo Contreras Gómez Carnet: 1009017

Guatemala de la Asunción 01 de octubre de 2018

## **REQUERIMIENTOS SOFTWARE**

- Para las aplicaciones generadas se debe tener un mínimo de 32MB de RAM, se recomienda que se tengan 48MB o más.
- El procesador debe de ser mínimo un Pentium 133 para compilar/ejecutar las aplicaciones generadas.

## **REQUERIMIENTOS HARDWARE**

- Java Development Kit - Compilador y Máquina Virtual
  - Para la compilación y ejecución de los programas es necesario tener el JDK 1.1.8 de Sun.
  - No es necesario tener todos los development kits, aunque es recomendable.

## **FUNCIONALIDADES**

### **Funciones Requeridas**

#### **Acceso al sistema**

Para la autenticación del sistema es necesario contar con un usuario y una contraseña que se deben de ir a comparar contra el archivo de usuarios donde se guardan todos los registros.

Existen dos roles, el rol administrador por default se debe asignar a un primer usuario en el sistema y por defecto a los demás es necesario asignar el rol de usuario.

La organización del archivo de Usuarios deberá de ser de tipo secuencial teniendo como llave principal el usuario.

#### **Validación de nivel de contraseña**

Validando por medio de puntuaciones se confirma que tan segura es la contraseña registrada validando que si es muy insegura no permite el registro al usuario.

#### **Path de fotografía**

Se deja al usuario elegir la ruta donde se encuentra almacenada su fotografía para el perfil.

## **Estatus**

Permite validar si el usuario se encuentra activo o no, si está activo se puede dar de baja el mismo o el mismo administrador lo puede hacer.

## **Mantenimiento Usuarios**

### **Usuario**

Cambiar password, correo, fecha de nacimiento, teléfono, fotografía.

### **Administrador**

Lo mismo que el usuario

Ingreso de nuevos usuarios

Buscar el usuario y poder modificar sus datos

Dar de baja a un usuario

## **Opción Respaldo de la Información**

Permite al usuario realizar una copia de todos los datos almacenados en la carpeta MEIA a una memoria secundaria y deja escoger la ruta y lo almacena en el directorio MEIA\_Backup.

La bitácora de las transacciones realizadas con el backup se almacena en su archivo y llevan una organización de apilo.

## **Funcionalidad Extra**

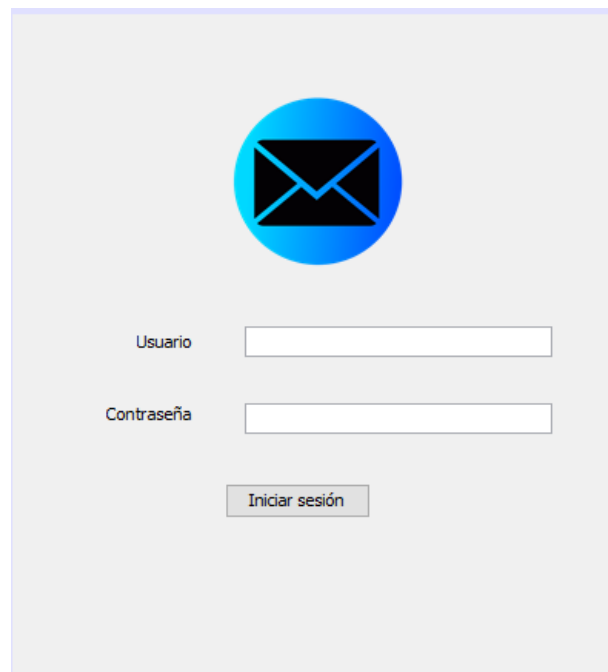
Cuando busca al usuario el administrador tienen la capacidad de modificar sus valores registrados por si en algún momento se le es solicitado.

## CLASES Y METODOS

```
public class ClaseGeneral {  
  
    public static boolean yaLogeado = false; //Booleano para validar si ya esta logeado algun usuario  
    public static boolean esAdmin = false; //Booleano para validar si el usuario logeado/a logear es admin  
    //Usuario logeado/a logear/a registrar para saber quien modifica los archivos  
    public static String usuarioActual = "Erick";  
    public static String usuariobuscado = "";  
    public static String usuario = "";  
    public static String rol = "";  
    public static String rutaFotografia = "";  
    //Booleano para validar que existan usuarios, y si no hay el primer registro se hace el admin  
    public static boolean hayUsuarios = false;  
    public static String[] datosUsuarioBuscado = null; //String para la busqueda de usuarios  
    public static boolean busqueda = false; //Booleano para validar que se esta modificando un usuario diferente al logeado  
    //Variables de backup  
    public static String rutaDestino = "";  
}
```

Clase que permite usar esas variables en toda la aplicación para su uso, valida rutas, valida campos para saber que acciones son las que tiene que realizar.

### Ventana de Login

The image shows a login window with a light gray background. At the top center is a blue circular icon containing a white envelope symbol. Below the icon, there are two input fields. The first is labeled 'Usuario' and the second is labeled 'Contraseña'. Both labels are positioned to the left of their respective input boxes. At the bottom center of the window is a button labeled 'Iniciar sesión'.

Ventana de Login que permite el acceso a la aplicación, lee el archivo usuario mediante los datos ingresados y los compara.

```

public static String Desencriptar(String textoEncriptado) {

    String secretKey = "MEIA"; //llave para desencriptar datos
    String base64EncryptedString = "";

    try {
        byte[] message = Base64.decodeBase64(textoEncriptado.getBytes("utf-8"));
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8"));
        byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
        SecretKey key = new SecretKeySpec(keyBytes, "DESede");

        Cipher decipher = Cipher.getInstance("DESede");
        decipher.init(Cipher.DECRYPT_MODE, key);

        byte[] plainText = decipher.doFinal(message);

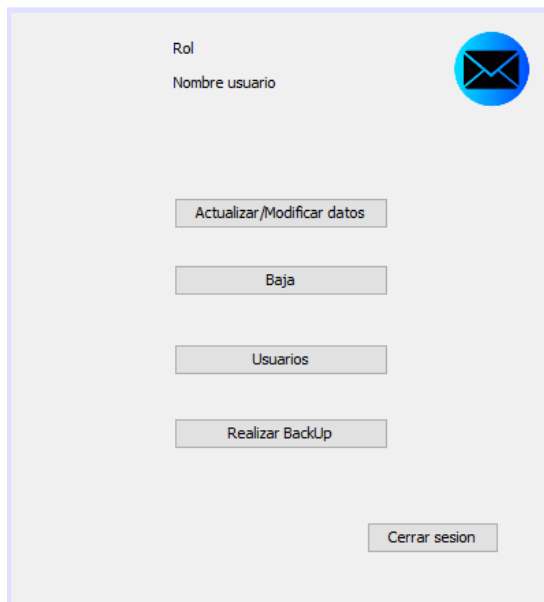
        base64EncryptedString = new String(plainText, "UTF-8");

    } catch (Exception ex) {
    }
    return base64EncryptedString;
}

```

Método de Desencriptar usando el DESede y valores de desencriptar con MD5, usando las librerías Cipher, este algoritmo obtiene los bytes de la contraseña a codificar y lo pasa únicamente para que sean utf-8. Para que estos valores puedan leerse o estipular un estándar usa la base64 y con ese logra pasar cada uno de los bytes a ese y así lograr que todos lleven como que un mismo formato y puedan comprimirse y descomprimirse de forma igual para todos.

## Ventana de mantenimiento



En esta parte de la aplicación el usuario tanto el admin como el usuario normal pueden realizar todas las acciones, actualizar los datos (correo, teléfono, contraseña) darse de baja automáticamente y el administrador puede buscar el usuario y darle de baja, de igual forma el tiene el acceso al registro para poder ingresar nuevos usuarios a la aplicación. El administrador también cuenta con la opción de backup que realiza una copia de todos los archivos que se encuentran en la carpeta MEIA y los pasa a la ruta que el usuario indica para destino

y guardarlo en dispositivos secundarios.

```

public void actualizarDescriptor2(String descriptor) {
    Archivo archivo = new Archivo();
    String[] split = archivo.leerArchivo("desc_" + descriptor);
    Date fecha = new Date();

    if (split[2].equals("usuario_creacion:")) {
        ClaseGeneral.usuarioActual = jUsuario.getText();
        split[2] = "usuario_creacion:" + ClaseGeneral.usuarioActual;
    }
    split[3] = "fecha_modificacion:" + fecha.toString();
    split[4] = "usuario_modificacion:" + ClaseGeneral.usuarioActual;
    //calcula el total de usuarios en el archivo original
    contarUsuarios(descriptor);
    split[5] = "#_registros:" + total;
    split[6] = "registro_activos:" + activos;
    split[7] = "registro_inactivos:" + inactivos;

    String error = "";
    archivo.limpiarArchivo("desc_" + descriptor);
    for (int i = 0; i < split.length; i++) {
        if (split[i] != null) {
            archivo.escribirArchivo("desc_" + descriptor, split[i], error);
        }
    }
}

```

Método de actualizarDescriptor, este método se encarga de la actualización de la información del descriptor ya que cada usuario puede realizar mas de algo que afecte ese archivo, en este método se puede realizar el conteo de registros disponibles tanto activos o inactivos en el archivo original usuarios.

```

public void actualizarDatos() {
    Archivo archivo = new Archivo();
    String[] datosUsuario = null, split = null, splitaux = null, cambiar = null;

    int posicion = 0;

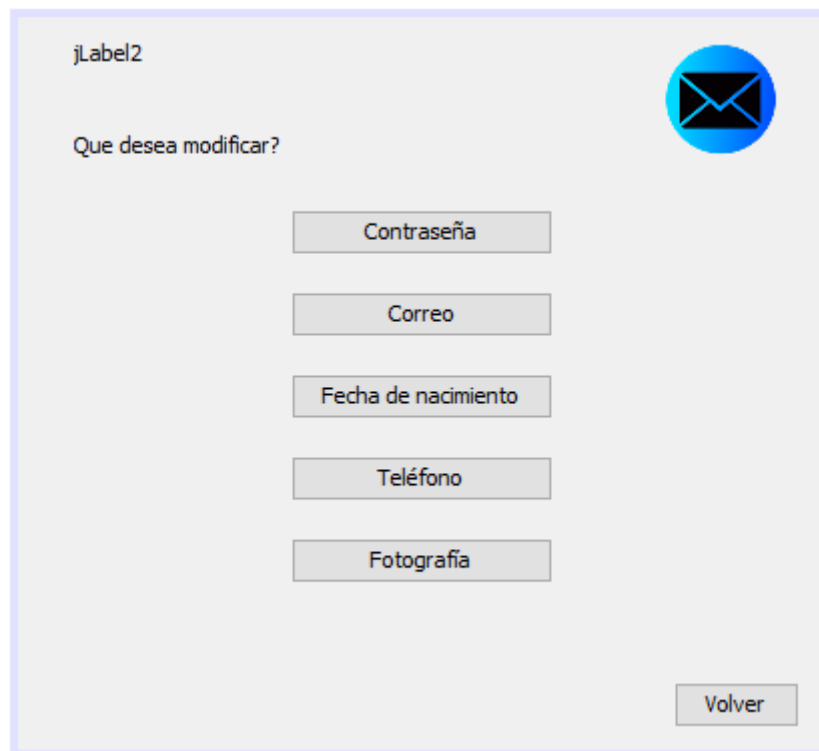
    split = archivo.leerArchivo("usuario");
    splitaux = split;
    for (int i = 0; i < split.length; i++) {
        if (split[i] != null) {
            datosUsuario = split[i].split("\\|");
            if (datosUsuario[0].equals(ClaseGeneral.datosUsuarioBuscado[0])) {
                posicion = i;
                splitaux[posicion] = "";
                cambiar = datosUsuario;
            } else {
                splitaux[i] = split[i];
            }
        }
    }

    cambiar[9] = "0";
}

```

Método de actualizarDatos, en este método se lee los archivos tanto de bitácora como de usuarios para la corrección de su información registrada, cuando se encuentra solo se obtiene la posición en la cual se encuentra y se cambia, en este caso es el método que cambia el estatus del usuario, le da de baja, por lo que pone 0.

## Ventana de cambios de la información



Solo se encarga de solicitar al usuario sus respectivos cambios que elija y quiera realizar y se guardan de inmediato.

```
int opc = JOptionPane.showConfirmDialog(null, "Desea modificar su número de teléfono?");
if (opc == 0) {
    String cambio = JOptionPane.showInputDialog(rootPane, "Ingrese el nuevo número");
    modificarDatosBuscado("tel", cambio);
    JOptionPane.showMessageDialog(null, "cambio exitoso");
}
```

Método de pedir el cambio respectivo de alguno de los registros, busca la información del usuario logeado, luego dependiendo del botón seleccionado se abre el cuadro de dialogo y pide el ingreso de lo solicitado. Se busca el registro y se guarda con lo nuevo.

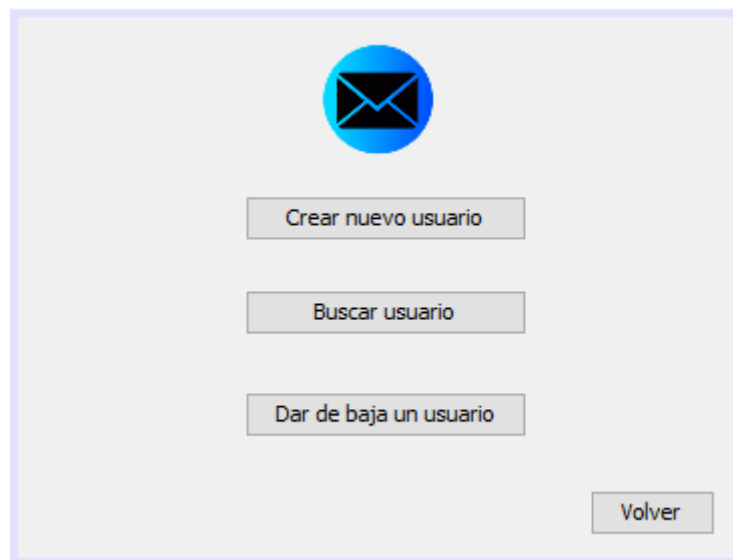
```

int opc = JOptionPane.showConfirmDialog(null, "Desea modificar la foto?");
if (opc == 0) {
    JFileChooser dialogo = new JFileChooser();
    FileNameExtensionFilter filtro = new FileNameExtensionFilter("Foto de perfil", "jpg");
    File ficheroImagen;
    String rutaArchivo = "";
    dialogo.setFileFilter(filtro);
    int valor = dialogo.showOpenDialog(this);
    if (valor == JFileChooser.APPROVE_OPTION) {
        ficheroImagen = dialogo.getSelectedFile();
        rutaArchivo = ficheroImagen.getPath();
        ClaseGeneral.rutaFotografia = rutaArchivo;
    }
    modificarDatosBuscado("foto", rutaArchivo);
    JOptionPane.showMessageDialog(null, "cambio exitoso");
}

```

Metodo en el cual se utilizan las librerías de FileChooser porque en este se captura la ruta de la imagen de perfil que el usuario quiera agregar, filtra todos los archivos por medio de la extensión jpg luego únicamente abre el cuadro y se elige la imagen.

### Ventana de Usuarios (Solo admin)



Ventana en la cual el administrador puede crear un usuario nuevo redirigiendo a la ventana de registro, luego viene la opción de buscar usuario, si lo encuentra puede mostrar el mensaje de si desea modificar su información, por ultimo tiene la opción de dar de baja a un usuario con el simple hecho de pasar el usuario lo encuentra y se da de baja.