UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1

CATEDRÁTICO: ING. WILLIAM ESTUARDO ESCOBAR ARGUETA

TUTOR ACADÉMICO: JOSUÉ RODOLFO MORALES CASTILLO

MANUAL TÉCNICO

NOMBRE ELEAZAR NEFTALÍ COLOP COLOP

CARNÉ: 3198935960914

SECCIÓN: A

GUATEMALA, 13 DE JUNIO DEL 2,024

ÍNDICE

INTRODUCCIÓN

El presente manual tiene como finalidad proporcionar una guía detallada para la implementación, uso y desarrollo futuro del sistema de inicio de sesión y gestión de usuarios de la plataforma de mensajería CipherChat. Se abordan los objetivos del sistema, los requisitos técnicos, la descripción de la solución y la lógica del programa.

OBJETIVOS

1. GENERAL

1.1. Este manual tiene como objetivo principal describir la funcionalidad y uso del sistema de inicio de sesión y gestión de usuarios en la plataforma CipherChat, proporcionando una referencia tanto para usuarios finales como para desarrolladores.

2. ESPECÍFICOS

2.1. Objetivo 1: Proveer instrucciones claras para la instalación y configuración del sistema, asegurando que los usuarios y desarrolladores puedan trabajar con él de manera efectiva

2.2. Objetivo 2: Explicar detalladamente la lógica del programa y la estructura del código, facilitando la comprensión y el mantenimiento del sistema por parte de los desarrolladores.

ALCANCES DEL SISTEMA

El objetivo de este manual es detallar explícitamente cómo utilizar y mantener el sistema de inicio de sesión y gestión de usuarios en CipherChat. Esto incluye la configuración inicial, la autenticación de usuarios, el registro de nuevos usuarios, y la gestión de perfiles, así como las capacidades especiales para

administradores y moderadores.

ESPECIFICACIÓN TÉCNICA

- **REQUISITOS DE HARDWARE**
  - Computadora con al menos 4GB de RAM
  - Procesador Intel Core i3 o equivalente
  - Al menos 500MB de espacio libre en disco
- **REQUISITOS DE SOFTWARE**

  Para trabajar con la aplicación y seguir un desarrollo futuro, el programador necesita:
  - Sistema operativo Windows, macOS o Linux
  - JDK (Java Development Kit) 8 o superior en mi caso jdk 21
  - IDE para desarrollo en Java (Eclipse, IntelliJ IDEA, NetBeans, etc.) en nuestro caso Netbeans.
  - Librerías de Java Swing para la interfaz gráfica
  - Biblioteca JTattoo para la apariencia de la interfaz gráfica

DESCRIPCIÓN DE LA SOLUCIÓN

- El desarrollo del sistema se basó en la necesidad de proporcionar una interfaz intuitiva y segura para la gestión de usuarios en CipherChat. Se analizaron los requisitos del enunciado para diseñar un sistema que permita la autenticación de usuarios, el registro de nuevos usuarios y la gestión de perfiles, asegurando además que los administradores y moderadores puedan gestionar la plataforma de manera efectiva.

LÓGICA DEL PROGRAMA

LOGIN

Librerías Usadas:

```
import java.awt.Color;
import java.awt.Font;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
```

Breve descripción de cada librería:

- java.awt.*: Utilizada para el manejo de componentes gráficos y eventos de la interfaz.
- javax.swing.*: Utilizada para la creación y manipulación de componentes de la interfaz gráfica.

Variables Globales de la Clase `LOGIN`

```
private JTextField usernameField;
private JPasswordField passwordField;
private JButton loginButton;
private JButton registerButton;
private JCheckBox cb1;
private UsuarioList usuarios;
```

Estas variables globales se utilizan para definir y manipular los componentes de la interfaz de inicio de sesión y mantener una referencia a la lista de usuarios.

Función Main

La función main inicializa el sistema, establece la apariencia de la interfaz y agrega usuarios predeterminados.

```
/**
 *
 * @author 3198935960914 - Eleazar Colop
 */
public class MAIN {

    static UsuarioList usuarios = new UsuarioList();

    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(className: "com.jtattoo.plaf.mcwin.McWinLookAndFeel");
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException | UnsupportedLookAndFee
            Logger.getLogger(name: LOGIN.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
        }

        // Agregar usuarios predeterminados
        usuarios.agregarUsuario(new Usuario(codigo: "1", contrasena: "123", nombre: "Fernanda", apellido: "Alvarez", genero:
        LOGIN login = new LOGIN(usuarios);
    }
}
```

Esta función se encarga de configurar la apariencia del sistema y agregar un usuario predeterminado antes de iniciar la ventana de login.

Procedimientos, Métodos y Funciones Utilizadas

Función initComponents

```
private void initComponents() {
```

Esta función configura y agrega los componentes de la interfaz gráfica, incluyendo etiquetas, campos de texto, botones y casillas de verificación.

Función actionPerformed

```
@Override
public void actionPerformed(ActionEvent ae) {
```

Esta función maneja los eventos generados por las acciones del usuario, como el clic en los botones de inicio de sesión y registro, y la selección de la casilla de verificación.

Función focusGained

```
@Override
public void focusGained(FocusEvent e) {
```

Esta función maneja el evento cuando el campo de contraseña gana el foco, permitiendo borrar el texto predeterminado y ocultar la contraseña.

Función focusLost

```
@Override
public void focusLost(FocusEvent e) {
```

Esta función maneja el evento cuando el campo de contraseña pierde el foco, mostrando el texto predeterminado y revelando la contraseña si está vacía.

Usuario

Librerías Usadas

import java.util.ArrayList;

Breve descripción:

- java.util.ArrayList: Utilizada para la gestión de listas dinámicas de usuarios.

Variables Globales de la Clase Usuario

```
private String codigo;
private String contrasena;
private String nombre;
private String apellido;
private String genero;
private int edad;
private String telefono;
private int mensajes_enviados;
private String[] mensajesReportados;
private int penalizaciones;
private boolean bloqueadoSistema;
private String[] contactos;
private int numContactos;
```

Estas variables representan las propiedades de un usuario, incluyendo sus credenciales, información personal, estado de cuenta y contactos.

Función Main

No aplicable en esta clase.

Procedimientos, Métodos y Funciones Utilizadas

Constructor Usuario

```
public Usuario(String codigo, String contrasena, String nombre, String apellido, String genero, int edad, String telefono, int mensajes_enviados) {
   // Inicialización de un usuario
   ...
}
```

Este constructor inicializa las propiedades de un usuario con los valores proporcionados.

UsuarioList

Librerías Usadas

import java.util.ArrayList;

Breve descripción:

- java.util.ArrayList: Utilizada para la gestión de listas dinámicas de usuarios.

Variables Globales de la Clase UsuarioList

private ArrayList<Usuario> usuarios;

Esta variable es una lista que contiene todos los usuarios registrados en el sistema.

Función Main

No aplicable en esta clase.

Procedimientos, Métodos y Funciones Utilizadas

Método agregarUsuario

```
public void agregarUsuario(Usuario usuario) {
   // Agregar un usuario a la lista
   ...
}
```

Este método agrega un usuario a la lista de usuarios.