

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



José Fernando Salguero García

CARNÉ: 202300708

SECCIÓN: B

GUATEMALA, 13 DE JUNIO DEL 2,024

# ÍNDICE

<b>ÍNDICE</b>	<b>1</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>OBJETIVOS</b>	<b>1</b>
1. GENERAL	1
2. ESPECÍFICOS	1
<b>ALCANCES DEL SISTEMA</b>	<b>1</b>
<b>ESPECIFICACIÓN TÉCNICA</b>	<b>1</b>
• REQUISITOS DE HARDWARE	1
• REQUISITOS DE SOFTWARE	1
<b>DESCRIPCIÓN DE LA SOLUCIÓN</b>	<b>2</b>
<b>LÓGICA DEL PROGRAMA</b>	<b>2</b>
❖ NOMBRE DE LA CLASE	
Captura de las librerías usadas	2
➤ Librerías	2
➤ Variables Globales de la clase _(El nombre de su clase actual)	3
➤ Función Main	3
➤ Métodos y Funciones utilizadas	3

## **INTRODUCCIÓN**

Este manual técnico está diseñado para proporcionar una visión general detallada del código fuente y la estructura de la aplicación de chat encriptado. Su objetivo es ayudar a los desarrolladores a entender cómo está construido el sistema, describiendo las clases, métodos y librerías utilizadas, así como proporcionar un diagrama visual de la arquitectura del sistema. Este manual es esencial para aquellos que buscan mantener, actualizar o expandir la funcionalidad de la aplicación.

## **OBJETIVOS**

### **1. GENERAL**

2. 1.1. Proveer una guía comprensiva que detalle la estructura y funcionamiento del código fuente de la aplicación de chat encriptado, facilitando su mantenimiento y desarrollo futuro.

### **3. ESPECÍFICOS**

- 2.1. Describir de manera clara y concisa las clases y métodos utilizados en el código, incluyendo sus responsabilidades y relaciones.
- 2.2. Presentar un diagrama visual de la arquitectura del sistema utilizando el lenguaje de modelado Mermaid, que permita a los desarrolladores comprender rápidamente la estructura y flujo del programa.

## **ALCANCES DEL SISTEMA**

El objetivo de este manual es proporcionar una referencia técnica detallada que permita a los desarrolladores entender completamente la arquitectura y funcionamiento interno del sistema de chat encriptado. Esto incluye una descripción detallada de cada clase y método, así como un diagrama visual que represente las interacciones entre las diferentes partes del sistema. Además, este

manual busca facilitar el proceso de mantenimiento y expansión del sistema al proporcionar una comprensión clara de su diseño y lógica de implementación.

## **ESPECIFICACIÓN TÉCNICA**

- **REQUISITOS DE HARDWARE**

- Intel Core Duo

- 4 GB de ram

- **REQUISITOS DE SOFTWARE**

- Windows xp

## **DESCRIPCIÓN DE LA SOLUCIÓN**

El programa es una aplicación de chat que permite a los usuarios comunicarse mediante mensajes encriptados utilizando matrices de cifrado. La aplicación incluye funcionalidades como la carga de matrices de cifrado, el envío y recepción de mensajes, y la capacidad de reportar mensajes a un moderador para su revisión.

## LÓGICA DEL PROGRAMA

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>jcommon</groupId>
    <artifactId>jcommon</artifactId>
    <version>0.9.5</version>
  </dependency>
  <dependency>
    <groupId>jfree</groupId>
    <artifactId>jfreechart</artifactId>
    <version>1.0.13</version>
  </dependency>
  <dependency>
    <groupId>com.jtattoo</groupId>
    <artifactId>JTattoo</artifactId>
    <version>1.6.13</version>
  </dependency>
</dependencies>
```

### ☐ Login

- **Propósito:** Manejar el inicio de sesión de los usuarios.
- **Métodos:**
  - Login(): Constructor que inicializa la interfaz de inicio de sesión.
  - setVisible(boolean b): Muestra u oculta la ventana de inicio de sesión.

## □ **Usuario**

- **Propósito:** Interfaz principal para los usuarios normales.
- **Métodos:**
  - Usuario(Persona persona, Login selfLogin): Constructor que inicializa la interfaz del usuario.
  - setVisible(boolean b): Muestra u oculta la ventana del usuario.

## □ **Mod**

- **Propósito:** Interfaz para los moderadores que permite revisar reportes.
- **Métodos:**
  - Mod(Persona persona, Login login): Constructor que inicializa la interfaz del moderador.
  - setVisible(boolean b): Muestra u oculta la ventana del moderador.
  - agregarReporte(Reporte reporte): Método para agregar un reporte a la lista de reportes.

## □ **Contactos**

- **Propósito:** Maneja la visualización y gestión de contactos del usuario.
- **Métodos:**
  - Contactos(int codigoPersonaLoggeada, Mod mod): Constructor que inicializa la lista de contactos.

## □ **Chat**

- **Propósito:** Interfaz de chat que permite a los usuarios comunicarse.
- **Métodos:**
  - Chat(String[] datosContacto, int codigoPersonaLoggeada, Mod mod): Constructor que inicializa la ventana de chat.
  - checkMatricesLoaded(): Verifica si las matrices A y B han sido cargadas.
  - cargarMatrizA(String ruta, int filas, int columnas): Carga la matriz A desde un archivo.
  - cargarMatrizB(String ruta, int filas): Carga la matriz B desde un archivo.

## □ **Chat2**

- **Propósito:** Panel interno de la clase Chat para la visualización y envío de mensajes.
- **Métodos:**
  - Chat2(String datosContacto, int codigoPersonaLoggeada, Mod mod): Constructor que inicializa el panel de chat.
  - setMatrizA(int[][] matrizA): Establece la matriz A.

- setMatrizB(int[][] matrizB): Establece la matriz B.
- setMatricesLoaded(boolean matricesLoaded): Establece si las matrices han sido cargadas.
- sendMessage(int[][] matrizA, int[][] matrizB): Envía un mensaje encriptado.

## 7. Reporte

- **Propósito:** Clase para representar un reporte de mensaje.
- **Métodos:**
  - Reporte(int codigoLoggeado, int codigoContacto, String mensaje, int[][] matrizM2, boolean estado): Constructor que inicializa un reporte.
  - Métodos getters para acceder a los campos del reporte.

## Librerías Usadas

- **javax.swing.\*:** Para la creación de interfaces gráficas de usuario.
- **java.awt.\*:** Para el manejo de eventos y componentes gráficos.
- **java.io.\*:** Para la lectura de archivos.
- **java.util.\*:** Para el manejo de colecciones y utilidades.





