



Facultad de Ciencias  
de la Computación

**BENEMERITA UNIVERSIDAD  
AUTONOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA  
COMPUTACION**

**MODELO DE REDES**

**PROF. IVAN OLMOS PINDA**

**ALUMNO: IGNACIO DÍAZ ROMERO**

**MATRICULA:201734055**

**2DO PARCIAL**

## Introducción:

Este proyecto simula conexión entre un cliente y un servidor, mediante un inicio de sesión que se realiza en el cliente, se manda el usuario y la contraseña que el servidor recibe y valida, primero si el usuario se encuentra dentro del registro de usuarios, y segundo, mediante una mezcla y un algoritmo MD5 valida si es la contraseña correcta, de esta manera permite o no el ingreso como ese usuario.

## Descripción de Programa:

El proyecto consiste en varias clases:

- Mezclador
- MD5
- Client
- Server
- User
- OPERACIONES

Este parcial fue desarrollado en Java, desde una máquina virtual de la distribución Debian de Linux, en editor de texto Geany.

Mediante la ejecución de Server.java y Client.java, estos llaman a las demás funciones respectivamente dentro de su main. Para ejecutar este programa se debe de abrir dos terminales. Tienes que compilar tanto Client.java como Server.java. De la siguiente manera:

- `javac Client.java`
- `javac Server.java`

Luego en una terminal debes de ejecutar el Server y en otra el Cliente de la siguiente manera:

- `java Client`
- `java Server`

## Clases

- **Mezclador:**

Se encarga de mezclar dos cadenas, A y B, dividiendo la longitud de A entre la longitud de B y este numero obtenido es la frecuencia con lo que se cambiaran los caracteres de A por el siguiente en B hasta que se llegue a la longitud de A. De igual forma esto pasara siempre y cuando no se haya llegado a la longitud de B. Retorna el texto ya mezclado.

Cuenta con un solo método que es:

**mezcla(String A, String B);**

- **MD5:**

Se encarga de generar el MD5 correspondiente de la cadena que se le pasa como atributo un arreglo de bytes para que respectivamente se realicen varias operaciones para que como resultado nos retorne el MD5 ya como String.

Cuenta con 3 métodos que son:

- **calculaMD5(byte[] message):** se obtiene el mensaje al que se le aplicara el MD5
- **toString():** Cambia los bytes recibidos por un String
- **M5D():** Retorna el MD5 generado del arreglo de bytes obtenido como String

- **Client**

Se encarga de hacer la del cliente que se conecta con un servidor, tiene un menú para registrar el usuario en el servidor y así de esta manera poder iniciar sesión después. Cuando se quiere iniciar sesión se comunica respectivamente con el servidor para que ambos hagan la mezcla y después el MD5 de los textos mezclados, para que así de esta manera puedan validar si la contraseña es correcta y permita el acceso al usuario. Cuenta con 7 métodos que son:

- **main():** Se encarga de entablar la conexión con el servidor, mediante la dirección “127.0.0.1” que es la del host local, con el puerto “1236”(puerto elegido al azar). También aquí se reciben los mensajes del servidor y los procesa mediante las demás funciones.
- **login():** Se encarga de obtener el nombre de usuario y contraseña para el inicio de sesión del usuario y retorna un User para que en el main se le asigne a alguna variable.
- **registro():** Se encarga de obtener el nombre de usuario y contraseña para el registro de algún usuario y retorna un User para que en el main se le asigne a alguna variable. (Se vio de una manera que pueda escalar el programa, ya que en el futuro tal vez al usuario se le pueda agregar algún atributo como “Fecha de nacimiento”, “Edad”, etc. De esta manera se modificaría este método y el login sería de la misma manera)
- **menú():** Se encarga de imprimir en pantalla el menú del cliente.
- **encriptar():** Se encarga de encriptar mediante la biblioteca Base64 de java.util.Base64
- **desencriptar():** Se encarga de desencriptar mediante la biblioteca Base 64 de java.util.Base64
- **clearScreen():** Se encarga de limpiar la pantalla de la consola.

- **Server**

Se encarga de conectar con los clientes que se comunican con el, de la misma manera que este se comunica con ellos. Recibe y manda mensajes. Asi como generar texto aleatorio y mandárselo al cliente, para que de esta manera ambos mezclen este texto con la contraseña y luego realicen el MD5, cada uno por su lado. Luego recibe el MD5 del cliente y lo compara, de esta manera permite acceso al cliente o no, tambien tiene la opción de que se registren usuarios y los guarde en su ArrayList de tipo Usuarios.

Cuenta con 1 método que es:

- **Main():** Se encarga de crear el socket del servidor con el puerto 1236, así como atender la solicitud del socket del cliente. Dentro de esta función se llama a la clase OPERACIONES() y se realizan todas las operaciones correspondientes.

- **User**

Es la clase en la cual se guardan el nombre y contraseña de un usuario, en el Server se tiene un ArrayList de este tipo para guardar a todos sus usuarios. Tiene como atributos name(nombre) y PW>Password).

Cuenta con 2 métodos que son sus respectivos getter de sus atributos: **getName();** y **getPW();**

- **OPERACIONES**

Esta clase hereda de la clase Thread=Hilo, y se encarga de todas las operaciones que tiene que hacer el servidor, que es procesar las solicitudes del cliente y mediante un BufferedReader y un PrintWriter, recibe y manda, respectivamente mensajes al cliente. En esta clase es en donde se llama a Mezclador y a MD5 para mezclar el texto aleatorio generado y después generar el MD5 para validar que la contraseña del usuario es correcta

Cuenta con su respectivo constructor y con 7 métodos que son:

- **run():** Se encargar de procesar las solicitudes de los clientes, mediante un switch y sus diferentes casos, si el usuario desea Logearse y mandar mensajes al server o si desea registrarse.
- **encriptar():** Se encarga de encriptar la cadena recibida mediante la biblioteca Base 64 de java.util.Base64. Retorna la cadena ya encriptada
- **desencriptar():** Se encarga de desencriptar la cadena recibida mediante la biblioteca Base 64 de java.util.Base64. Retorna la cadena ya desencriptada
- **findUser():** Se encarga de encontrar el usuario en el ArrayList de Users, recibe el nombre del usuario a buscar y retorna la posición en el ArrayList.
- **randomText():** Se encarga de generar un texto aleatorio. Retorna este texto aleatorio.
- **clearScreen():** Se encarga de limpiar la consola
- **printUsers():** Se encarga de imprimir los usuarios que están en el ArrayList de Users

## **Conclusión**

Mediante la realización de este programa se entendió lo valioso que es la seguridad de la información que se maneja por medio de la capa de enlace de datos. Realmente el algoritmo de encriptación, así como el MD5 mejoran la seguridad, ya que si no sería demasiado sencillo robar información valiosa y sería un gran problema social y económico.