INFORME CASO 2

Juan Esteban Rodríguez Ospino – 202011171

Gabriel Esteban Beltran Lopez - 201921903

INFRAESTRUCTURA TECNOLÓGICA

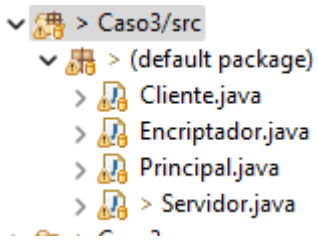
UNIVERSIDAD DE LOS ANDES

2022

Tabla de contenido

1) **Organización de archivos**

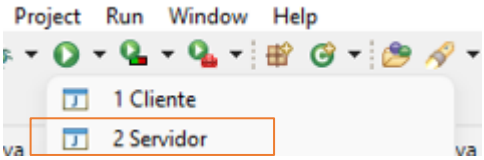
El programa se divide en cuatro clases que estarán en el paquete default: Cliente, Encriptador, Principal y servidor.



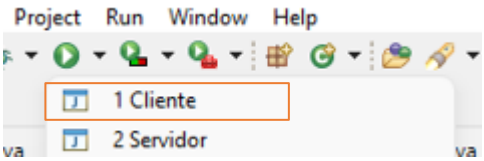
Cliente y Servidor se comunicarán mediante Sockets, y las clases Encriptador y Principal nos permitirán aplicar métodos para cifrar la comunicación entre estos.

2) **Instrucciones para correr**

1) **Correr la clase Servidor: Al correr esta clase se desplegará una interfaz gráfica dónde llegarán los mensajes del cliente.**



2) **Correr cliente: Una vez se esté ejecutando el servidor, deberemos ejecutar el cliente, en este se te preguntará si quieres que se ejecute iterativamente o secuencialmente.**



3) Descripción esquema de llaves

En la clase Encriptador generamos los códigos correspondientes para poder generar llaves simétricas y asimétricas, publicas y privadas para poder manipular las cadenas de caracteres que se envían entre Servidor y Cliente.

```
public KeyPair generateAsymmetricKeyPair() {
    KeyPairGenerator keyGen = null;
    KeyPair keyPair = null;
    try {
        keyGen = KeyPairGenerator.getInstance(ASYMMETRIC_ALGORITHM);
        keyGen.initialize(1024);
        keyPair = keyGen.genKeyPair();
        PrivateKey privateKey = keyPair.getPrivate();
        PublicKey publicKey = keyPair.getPublic();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return keyPair;

}
public Key generateSymmetricKey() {
    KeyGenerator keyGen = null;
    try {
        keyGen = KeyGenerator.getInstance(SYMMETRIC_ALGORITHM);
        keyGen.init(256);
    } catch (NoSuchAlgorithmException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return keyGen.generateKey();
}
```