**UNIVERSIDADE SÃO JUDAS TADEU**

**SISTEMAS DE INFORMAÇÃO**

**LARISSA OLIVEIRA DOS SANTOS**

**UC SISTEMAS COMPUTACIONAIS E SEGURANÇA**

**São Paulo**

**2025**

**ATIVIDADE SOBRE CRIPTOGRAFIA**

* Criptografia com Chaves Simétricas - em Java

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.IvParameterSpec;

import java.util.Base64;

public class AESExample {

public static void main(String[] args) throws Exception {

String data = "Criptografia Simétrica";

// Gerar uma chave secreta AES

KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");

keyGenerator.init(256); // Chave de 256 bits

SecretKey secretKey = keyGenerator.generateKey();

// Gerar um vetor de inicialização (IV)

byte[] iv = new byte[16]; // IV de 16 bytes para AES

IvParameterSpec ivSpec = new IvParameterSpec(iv);

// Criptografar dados

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

cipher.init(Cipher.ENCRYPT\_MODE, secretKey, ivSpec);

byte[] encryptedData = cipher.doFinal(data.getBytes());

// Descriptografar dados

cipher.init(Cipher.DECRYPT\_MODE, secretKey, ivSpec);

byte[] decryptedData = cipher.doFinal(encryptedData);

System.out.println("Texto original: " + data);

System.out.println("Texto criptografado (Base64): " + Base64.getEncoder().encodeToString(encryptedData));

System.out.println("Texto descriptografado: " + new String(decryptedData));

}

}

* Criptografia com Chaves Assimétricas- em Java

import javax.crypto.Cipher;

import java.security.\*;

public class RSAExample {

public static void main(String[] args) throws Exception {

// Gerar par de chaves RSA

KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");

keyPairGenerator.initialize(2048);

KeyPair keyPair = keyPairGenerator.generateKeyPair();

PublicKey publicKey = keyPair.getPublic();

PrivateKey privateKey = keyPair.getPrivate();

// Mensagem a ser criptografada

String data = "Criptografia Assimétrica";

// Criptografar a mensagem com a chave pública

Cipher cipher = Cipher.getInstance("RSA");

cipher.init(Cipher.ENCRYPT\_MODE, publicKey);

byte[] encryptedData = cipher.doFinal(data.getBytes());

// Descriptografar a mensagem com a chave privada

cipher.init(Cipher.DECRYPT\_MODE, privateKey);

byte[] decryptedData = cipher.doFinal(encryptedData);

System.out.println("Texto original: " + data);

System.out.println("Texto criptografado: " + new String(encryptedData));

System.out.println("Texto descriptografado: " + new String(decryptedData));

}

}

* Função Hash- em Java

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

public class HashExample {

public static void main(String[] args) throws NoSuchAlgorithmException {

String data = "Função Hash SHA-256";

// Criar o MessageDigest para SHA-256

MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");

// Gerar o hash

byte[] hashBytes = messageDigest.digest(data.getBytes());

// Converter para formato hexadecimal

StringBuilder hexString = new StringBuilder();

for (byte b : hashBytes) {

hexString.append(String.format("%02x", b));

}

System.out.println("Texto original: " + data);

System.out.println("Hash SHA-256: " + hexString.toString());

}

}