

UNIVERSIDADE SÃO JUDAS TADEU

SISTEMAS COMPUTACIONAIS E SEGURANÇA

ALUNO:

MICAEL WILLIAM - RA: 824213069

SÃO PAULO
2025

Demonstração de Criptografia Simétrica com AES em Java

```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

public class AESCrypto
{
    public static SecretKey gerarChave() throws Exception
    {
        KeyGenerator keyGen = KeyGenerator.getInstance("AES");
        keyGen.init(128);
        return keyGen.generateKey();
    }

    public static String criptografar(String texto, SecretKey chave) throws Exception
    {
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE, chave);
        byte[] textoCriptografado = cipher.doFinal(texto.getBytes());
        return Base64.getEncoder().encodeToString(textoCriptografado);
    }

    public static String descriptografar(String textoCriptografado, SecretKey chave) throws Exception
    {
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.DECRYPT_MODE, chave);
        byte[] bytesDecodificados = Base64.getDecoder().decode(textoCriptografado);
        byte[] textoOriginal = cipher.doFinal(bytesDecodificados);
        return new String(textoOriginal);
    }

    public static void main(String[] args)
    {
        try {
            SecretKey chave = gerarChave();

            String textoOriginal = "Texto Secreto!";
            System.out.println("Texto original: " + textoOriginal);

            String textoCriptografado = criptografar(textoOriginal, chave);
            System.out.println("Texto criptografado: " + textoCriptografado);

            String textoDescriptografado = descriptografar(textoCriptografado, chave);
            System.out.println("Texto descriptografado: " + textoDescriptografado);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Demonstração de Criptografia Assimétrica com RSA em Java

```
import java.security.*;
import javax.crypto.Cipher;
import java.util.Base64;

public class RSACrypto
{
    public static KeyPair gerarParDeChaves() throws Exception
    {
        KeyPairGenerator keyGen = KeyPairGenerator.getInstance("RSA");
        keyGen.initialize(2048);
        return keyGen.generateKeyPair();
    }

    public static String criptografar(String texto, PublicKey chavePublica) throws Exception
    {
        Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.ENCRYPT_MODE, chavePublica);
        byte[] textoCriptografado = cipher.doFinal(texto.getBytes());
        return Base64.getEncoder().encodeToString(textoCriptografado);
    }

    public static String descriptografar(String textoCriptografado, PrivateKey chavePrivada) throws Exception
    {
        Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.DECRYPT_MODE, chavePrivada);
        byte[] bytesDecodificados = Base64.getDecoder().decode(textoCriptografado);
        byte[] textoOriginal = cipher.doFinal(bytesDecodificados);
        return new String(textoOriginal);
    }

    public static void main(String[] args)
    {
        try {
            KeyPair parDeChaves = gerarParDeChaves();
            PublicKey chavePublica = parDeChaves.getPublic();
            PrivateKey chavePrivada = parDeChaves.getPrivate();

            String textoOriginal = "Mensagem confidencial!";
            System.out.println("Texto original: " + textoOriginal);

            String textoCriptografado = criptografar(textoOriginal, chavePublica);
            System.out.println("Texto criptografado: " + textoCriptografado);

            String textoDescriptografado = descriptografar(textoCriptografado, chavePrivada);
            System.out.println("Texto descriptografado: " + textoDescriptografado);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Demonstração da função Hash (SHA-256) em Java

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Base64;

public class Hash
{
    public static String gerarHashSHA256(String texto)
    {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hashBytes = digest.digest(texto.getBytes());
            return Base64.getEncoder().encodeToString(hashBytes);
        } catch (NoSuchAlgorithmException e) {
            throw new RuntimeException("Algoritmo SHA-256 não encontrado!", e);
        }
    }

    public static void main(String[] args)
    {
        String textoOriginal = "MinhaSenhaSuperSecreta";
        String hashGerado = gerarHashSHA256(textoOriginal);
        System.out.println("Texto original: " + textoOriginal);
        System.out.println("Hash SHA-256: " + hashGerado);
    }
}
```

Exibição do resultado do algoritmo de Criptografia Simétrica com AES

```
Texto original: Texto Secreto!  
Texto criptografado: gam5Cl1r3du0kDG1aJvP2Q==  
Texto descriptografado: Texto Secreto!
```

Exibição do resultado do algoritmo de Criptografia Assimétrica com RSA

```
Texto original: Mensagem confidencial!  
Texto criptografado:  
3tKsFLrs6nAih38jSube78i1TE8uWY0IBKKI2KSeeiVhnbJWYJEdknYgXSh4VpbzPFQwPPSMFuLLEtNhaQaa5Uqwbc2m+eMbWb0ZDRyUu  
yRiP8LajaEpo8ymgPDmjyK0B8MYWwKV8UVpasUcnUGT5Enk1rAA4UhnHERcFzCZHS6+8W3GNMdbWQCDV9mIzSt9dBKRpEoy8wQxkUMfxuZ  
+JgFuJQYNqysg+TVdLzTKgpoNS/FJquwAWvAVRqdyWMS6sqM9L1E3VFU/Ws+EKtH8h3cy6b6+wyEr+nXw/6Daw+3aru85+RkkwQqTp8j5H  
Q6px5gMAFBRSo95cfDVz0wgw==  
Texto descriptografado: Mensagem confidencial!
```

Exibição do resultado do algoritmo de função Hash (SHA-256)

```
Texto original: MinhaSenhaSuperSecreta  
Hash SHA-256: obDPHG9hxsmwOHm+Wcd2DeGaSTL4vF7P0TUWneTb70Y=
```