Universidade São Judas Tadeu
Sistemas Computacionais e Segurança

Codificação de Algoritmos de Criptografia

João Vitor Chioatto Serafim – 825133189

Prof ° Robson Calvetti

São Paulo – SP
2025

1. **Código** **Função** **Hash**

```
import java.security.MessageDigest;
import java.nio.charset.StandardCharsets;
import java.util.Base64;

public class HashDemo {
   public static void main(String[] args) {
      try {
         String texto = "Teste função Hash ";

                  MessageDigest digest = MessageDigest.getInstance("SHA-256");
         byte[] hashBytes = digest.digest(texto.getBytes(StandardCharsets.UTF_8));

               String hashBase64 = Base64.getEncoder().encodeToString(hashBytes);

         System.out.println("  Função Hash (SHA-256) ");

        System.out.println("Texto normal: " + texto);

         System.out.println("Hash (Base64): " + hashBase64);
           } catch (Exception e) {
         e.printStackTrace();

      }

   }
}
```

2. Criptografia Simétrica - AES

```
import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;
```

```
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.security.SecureRandom;
import java.util.Base64;
import java.nio.charset.StandardCharsets;

public class AESCrypto {
    public static void main(String[] args) {
        try {
            String mensagem = "Teste Criptografia Simétrica ";
  KeyGenerator keyGen = KeyGenerator.getInstance("AES");
            keyGen.init(128);
            SecretKey chave = keyGen.generateKey();

            byte[] iv = new byte[16];
            new SecureRandom().nextBytes(iv);
            IvParameterSpec ivSpec = new IvParameterSpec(iv);

            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, chave, ivSpec);

 byte[] textoCifrado = cipher.doFinal(mensagem.getBytes(StandardCharsets.UTF_8));

            String base64Cipher = Base64.getEncoder().encodeToString(textoCifrado);
            String base64Key = Base64.getEncoder().encodeToString(chave.getEncoded());
            String base64Iv = Base64.getEncoder().encodeToString(iv);

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

```
        decipher.init(Cipher.DECRYPT_MODE, new SecretKeySpec(chave.getEncoded(),
"AES"), ivSpec);
        String textoOriginal = new String(decipher.doFinal(textoCifrado),
StandardCharsets.UTF_8);


        System.out.println("=== Criptografia Simétrica (AES) ===");
        System.out.println("Texto original: " + mensagem);
        System.out.println("Texto criptografado (Base64): " + base64Cipher);
        System.out.println("Chave (Base64): " + base64Key);
        System.out.println("IV (Base64): " + base64Iv);
        System.out.println("Texto descriptografado: " + textoOriginal);


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

3. Criptografia Assimétrica - RSA

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

```
public class RSACrypto {
    public static void main(String[] args) {
        try {
            String mensagem = "Teste Criptografia Assimetrica! ";

            KeyPairGenerator keyGen = KeyPairGenerator.getInstance("RSA");
            keyGen.initialize(2048);
            KeyPair parChaves = keyGen.generateKeyPair();
            PublicKey chavePublica = parChaves.getPublic();
            PrivateKey chavePrivada = parChaves.getPrivate();

            Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
            cipher.init(Cipher.ENCRYPT_MODE, chavePublica);

byte[] textoCifrado = cipher.doFinal(mensagem.getBytes(StandardCharsets.UTF_8));
            String base64Cipher = Base64.getEncoder().encodeToString(textoCifrado);

                cipher.init(Cipher.DECRYPT_MODE, chavePrivada);
            byte[] textoDecifrado = cipher.doFinal(Base64.getDecoder().decode(base64Cipher));
            String textoOriginal = new String(textoDecifrado, StandardCharsets.UTF_8);

            System.out.println("=== Criptografia Assimétrica (RSA) ===");
            System.out.println("Texto original: " + mensagem);
            System.out.println("Texto criptografado (Base64): " + base64Cipher);
            System.out.println("Texto descriptografado: " + textoOriginal);

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

}