**UNIVERSIDADE SÃO JUDAS TADEU-MOOCA**

**SISTEMAS DE INFORMAÇÃO**

**Henrique Lima Cândido**

**UC SISTEMAS COMPUTACIONAIS E SEGURANÇA**

**São Paulo**

**2025**

**ATIVIDADE SOBRE CRIPTOGRAFIA**

**Em Python em aesfrom Crypto.Cipher import AES**

**“from Crypto.Util.Padding import pad, unpad**

**from Crypto.Random import get\_random\_bytes**

**# Gerar uma chave de 256 bits (32 bytes)**

**key = get\_random\_bytes(32)**

**# Gerar um vetor de inicialização (IV) de 16 bytes**

**iv = get\_random\_bytes(16)**

**# Mensagem a ser criptografada**

**data = b'Criptografia Simétrica'**

**# Criptografar a mensagem com o AES (modo CBC)**

**cipher = AES.new(key, AES.MODE\_CBC, iv)**

**ciphertext = cipher.encrypt(pad(data, AES.block\_size))**

**# Descriptografar a mensagem**

**decipher = AES.new(key, AES.MODE\_CBC, iv)**

**original\_data = unpad(decipher.decrypt(ciphertext), AES.block\_size)**

**# Exibir os resultados**

**print(f'Texto original: {data.decode("utf-8")}')**

**print(f'Texto criptografado (em bytes): {ciphertext}')**

**print(f'Texto descriptografado: {original\_data.decode("utf-8")}')”**

**Exemplo de código para Criptografia Assimétrica (RSA) em Python:**

**“from Crypto.PublicKey import RSA**

**from Crypto.Cipher import PKCS1\_OAEP**

**from Crypto.Random import get\_random\_bytes**

**# Gerar chaves pública e privada RSA**

**key = RSA.generate(2048)**

**private\_key = key**

**public\_key = key.publickey()**

**# Mensagem a ser criptografada**

**data = b'Criptografia Assimétrica'**

**# Criptografar a mensagem com a chave pública**

**cipher\_rsa = PKCS1\_OAEP.new(public\_key)**

**ciphertext = cipher\_rsa.encrypt(data)**

**# Descriptografar a mensagem com a chave privada**

**decipher\_rsa = PKCS1\_OAEP.new(private\_key)**

**decrypted\_message = decipher\_rsa.decrypt(ciphertext)**

**# Exibir os resultados**

**print(f'Mensagem original: {data.decode("utf-8")}')**

**print(f'Mensagem criptografada: {ciphertext}')**

**print(f'Mensagem descriptografada: {decrypted\_message.decode("utf-8")}')”**

**Função Hash   
import hashlib**

**“# Mensagem a ser hasheada**

**data = "Função Hash SHA-256".encode('utf-8')**

**# Gerar o hash SHA-256**

**hash\_object = hashlib.sha256()**

**hash\_object.update(data)**

**hashed\_data = hash\_object.hexdigest()**

**# Exibir os resultados**

**print(f'Texto original: {data.decode("utf-8")}')**

**print(f'Hash SHA-256: {hashed\_data}')”**

**Fonte:Chatgpt**