**Atividade: Implementação de Algoritmos Criptográficos em Python**

**Linguagem utilizada:** Python  
**Bibliotecas utilizadas:** cryptography, rsa, hashlib

**1. Criptografia com Chaves Simétricas (AES)**

from cryptography.fernet import Fernet

# Gerar chave

chave = Fernet.generate\_key()

fernet = Fernet(chave)

# Mensagem original

mensagem = "Segredo muito importante"

mensagem\_cripto = fernet.encrypt(mensagem.encode())

mensagem\_descripto = fernet.decrypt(mensagem\_cripto)

print("Mensagem original:", mensagem)

print("Mensagem criptografada:", mensagem\_cripto)

print("Mensagem descriptografada:", mensagem\_descripto.decode())

**2. Criptografia com Chaves Assimétricas (RSA)**

import rsa

# Gerar par de chaves

(public\_key, private\_key) = rsa.newkeys(512)

# Mensagem

mensagem = "Mensagem confidencial"

mensagem\_cripto = rsa.encrypt(mensagem.encode(), public\_key)

mensagem\_descripto = rsa.decrypt(mensagem\_cripto, private\_key)

print("Mensagem original:", mensagem)

print("Mensagem criptografada:", mensagem\_cripto)

print("Mensagem descriptografada:", mensagem\_descripto.decode())

**3. Função Hash (SHA-256)**

import hashlib

mensagem = "Minha senha super segura"

# Converter string para bytes e gerar hash

mensagem\_hash = hashlib.sha256(mensagem.encode()).hexdigest()

print("Mensagem original:", mensagem)

print("Hash gerado (SHA-256):", mensagem\_hash)

**Requisitos**

Para executar os códigos acima, é necessário instalar as bibliotecas externas:

pip install cryptography rsa