1 Criptografía con Java

Javascript dispone de paquetes para trabajar con herramientas de criptografía. Con estos paquetes se pueden trabajar con varios motores para realizar los cifrados, llamados proveedores.

Para ello se usará el módulo crypto ya incluido en NodeJS. Para importarlo se usará:

```
const crypto = require('node:crypto')
```

2 Funciones HASH

Java puede aplicar diversos algoritmos hash a un mensaje. El método getHashes() devuelve una lista con todos los algoritmos soportados.

Con el siguiente código se puede obtener un listado de los algoritmos soportados:

```
const crypto = require('node:crypto')

console.log(crypto.getHashes());

crypto.getHashes().forEach((hash) => {console.log('Algoritmo: ' + hash)})
```

Los algoritmos soportados son:

```
Algoritmo: RSA-MD5
Algoritmo: RSA-RIPEMD160
Algoritmo: RSA-SHA1
Algoritmo: RSA-SHA1-2
Algoritmo: RSA-SHA224
Algoritmo: RSA-SHA256
Algoritmo: RSA-SHA3-224
Algoritmo: RSA-SHA3-256
Algoritmo: RSA-SHA3-384
Algoritmo: RSA-SHA3-512
Algoritmo: RSA-SHA384
Algoritmo: RSA-SHA512
Algoritmo: RSA-SHA512/224
Algoritmo: RSA-SHA512/256
Algoritmo: RSA-SM3
Algoritmo: blake2b512
Algoritmo: blake2s256
Algoritmo: id-rsassa-pkcs1-v1_5-with-sha3-224
Algoritmo: id-rsassa-pkcs1-v1_5-with-sha3-256
Algoritmo: id-rsassa-pkcs1-v1_5-with-sha3-384
Algoritmo: id-rsassa-pkcs1-v1_5-with-sha3-512
Algoritmo: md5
Algoritmo: md5-sha1
Algoritmo: md5WithRSAEncryption
Algoritmo: ripemd
Algoritmo: ripemd160
Algoritmo: ripemd160WithRSA
Algoritmo: rmd160
Algoritmo: sha1
Algoritmo: sha1WithRSAEncryption
Algoritmo: sha224
Algoritmo: sha224WithRSAEncryption
Algoritmo: sha256
Algoritmo: sha256WithRSAEncryption
Algoritmo: sha3-224
Algoritmo: sha3-256
Algoritmo: sha3-384
Algoritmo: sha3-512
```

```
Algoritmo: sha384
Algoritmo: sha384WithRSAEncryption
Algoritmo: sha512
Algoritmo: sha512-224
Algoritmo: sha512-224WithRSAEncryption
Algoritmo: sha512-256
Algoritmo: sha512-256WithRSAEncryption
Algoritmo: sha512WithRSAEncryption
Algoritmo: shake128
Algoritmo: shake256
Algoritmo: sm3
Algoritmo: sm3WithRSAEncryption
Algoritmo: ssl3-md5
Algoritmo: ssl3-sha1
```

Sabiendo los algoritmos se puede usar el método createHash() para obtener un objeto que calcule el resumen (hash) del mensaje:

```
const crypto = require('node:crypto')

const hashFunc = crypto.createHash('md5')
hashFunc.update('Hola mundo')
// Se puede añadir más información para calcular el MD5
hashFunc.update('Adios mundo')
// El MD5 en hexadecimal:
console.log(hashFunc.digest('hex'))
// También se puede en base64, pero no se pueden hacer dos
// llamadas a la función digest, pues se genera un error.
//console.log(hashFunc.digest('base64'))
```

Como se puede ver de la salida del programa, el resumen del mensaje es una cadena de bytes ilegible, por lo que es normal transformarlos a Base 64 o hexadecimal para almacenarlos o enviarlos por la red.

Ejercicio: Nos pasan el md5 de un mensaje en Base 64 siendo:

TOe2L3H40IAR5im38ZsANg==

Se ha tenido un problema y no se sabe a cuál de los siguientes textos corresponde:

- En un lugar de la mancha
- De cuyo nombre no quiero acordarme

Escriba un programa que calcule el MD5 de los dos textos y los compare con el MD5 del que se dispone. El programa debe indicar cuál de los dos textos es el correcto. Nota: Se deben respetar mayúsculas y minúsculas