Python

Firmas Digitales

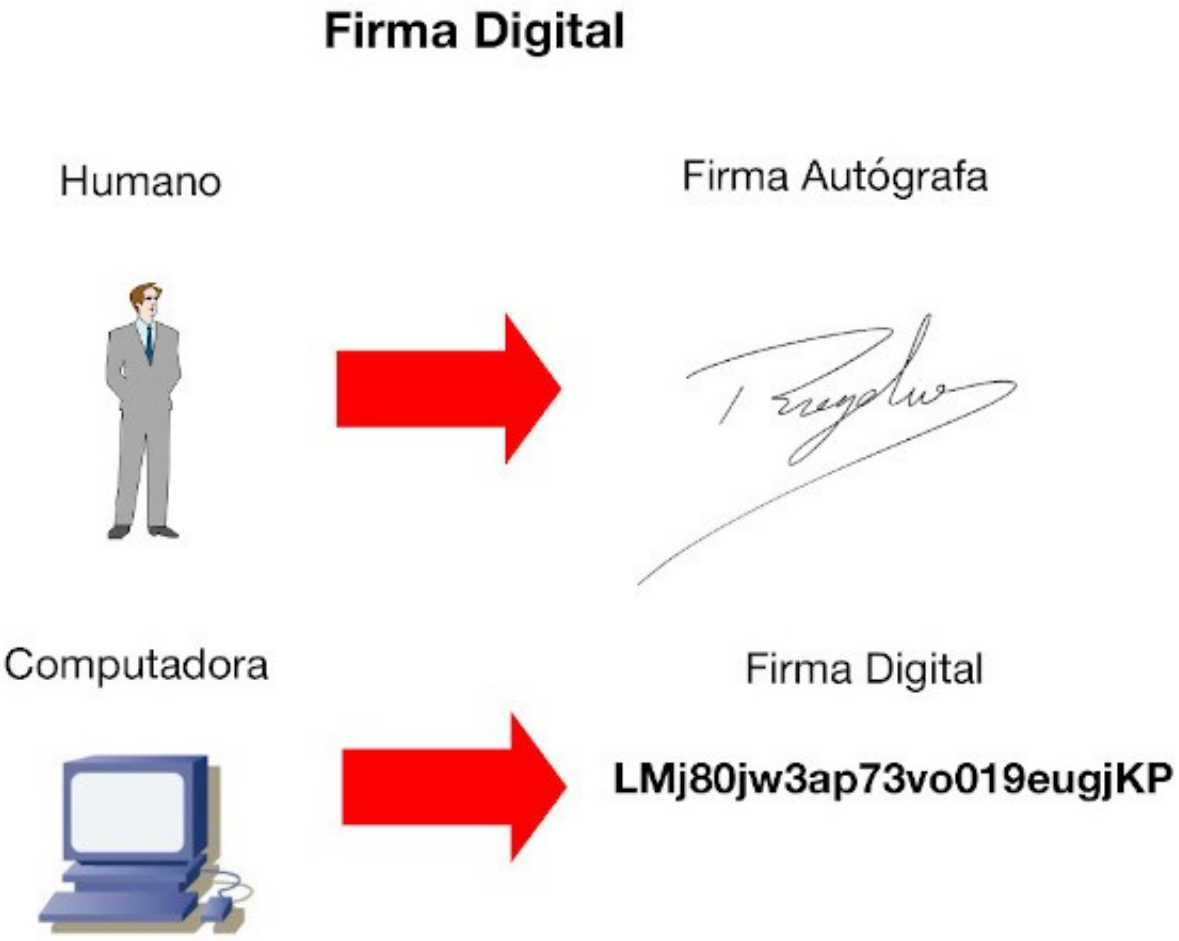
Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia

Firmas Digitales

Firmas

- **Firmas autógrafas y digitales:**



Métodos para generar Firmas Digitales

Firmas Digitales

- **Existen varias técnicas para generación de Firmas Digitales:**
  - **MD5**
  - **SHA1**
  - **SHA2**

Generación de Firmas Digitales con Python

Números Aleatorios

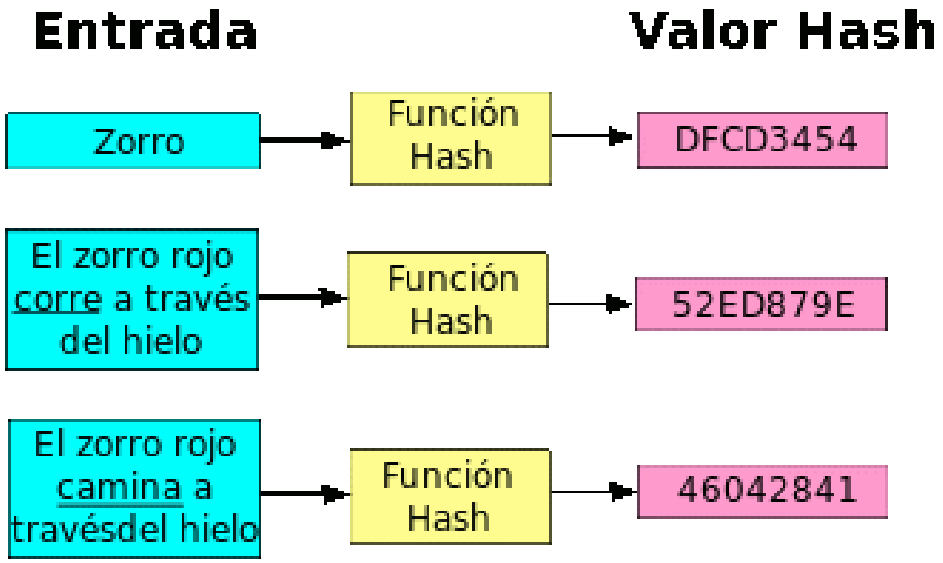
- **Se utilizarán números aleatorios para generar cadenas, por lo cual requerimos utilizar la biblioteca “random”:**

```
import random
```

Funciones HASH

- **Se utilizarán funciones HASH para generar las firmas.**

- **Una función "hash" tiene como entrada un conjunto de elementos, que suelen ser cadenas, y los convierte en un rango de salida finito, normalmente cadenas de longitud fija.**



"Función hash", Wikipedia, https://es.wikipedia.org/wiki/Funci%C3%B3n_hash, noviembre 2020

Funciones Hash

- **Se utilizarán funciones “hash”, por lo cual requerimos utilizar la biblioteca “hashlib”:**

```
import hashlib
```

Universo de caracteres

- **Se definirá un conjunto de caracteres posibles para generar una cadena aleatoria.**

- **En este ejemplo sólo se incluyen letras minúsculas y números, pero para mayor seguridad se recomienda agregar mayúsculas y caracteres especiales.**

```
caracteres = "abcdefghijklmnopqrstuvwxyz0123456789"
```

Cadena Aleatoria

- **A partir de nuestro universo de caracteres, generamos una funcion que nos genera una cadena aleatoria con una longitud de 10. Esta cadena aleatoria se utilizará como entrada de nuestra función "hash" para generar la firma:**

```
def generar_cadena():
    cadena = ""
    for ciclo in range(0,10):
        cadena = cadena + random.choice(caracteres)
    return cadena

cadena = generar_cadena()
```

- **Para este ejemplo se generó:**

```
Cadena aleatoria generada: 4xbp406eb
```

Firma con MD5

- **Usamos MD5 para generar la “Firma Digital” y le damos de como entrada nuestra cadena aleatoria que generamos anteriormente:**

```
firma_md5 = (hashlib.md5(cadena.encode())).hexdigest()
```

- **Mandamos a imprimir:**

```
print("\nMD5: " + firma_md5)
```

- **Ya impreso en pantalla nos genera lo siguiente:**

```
MD5: 3d53086661937c2f1d9c9578b8b21ea3
```

Firma con SHA

- **También podemos usar SHA (en sus diferentes versiones) para generar la "Firma Digital", y al igual que el ejemplo anterior, le damos de como entrada nuestra cadena aleatoria que generamos anteriormente (se incluye la de MD5 anterior):**

```
firma_md5 = (hashlib.md5(cadena.encode())).hexdigest()
firma_sha1 = (hashlib.sha1(cadena.encode())).hexdigest()
firma_sha224 = (hashlib.sha224(cadena.encode())).hexdigest()
firma_sha256 = (hashlib.sha256(cadena.encode())).hexdigest()
firma_sha384 = (hashlib.sha384(cadena.encode())).hexdigest()
firma_sha512 = (hashlib.sha512(cadena.encode())).hexdigest()
```

Firma con SHA

- **Mandamos a imprimr las cadenas generadas:**

```
print("\nMD5: " + firma_md5)
print("SHA1: " + firma_sha1)
print("SHA224: " + firma_sha224)
print("SHA256: " + firma_sha256)
print("SHA384: " + firma_sha384)
print("SHA512: " + firma_sha512)
```

Firma con SHA

- **La salida en pantalla es la siguiente:**

```
MD5: 40a4f4b41df1aa3d89e1feba05e12222
SHA1: e068d82794320739db4b918fe577792fb4a5f0b1
SHA224: 3baed3154bff34e2b5c75ac15ac7df23b2970b0fcb29c5c444d85986
SHA256: 9032f786796a7cc7f1c5becbcad47a46641c2535c8b14822cad71ade7daa161b
SHA384: bdf72af141257120561530aef862808e9e6faff4b9de2462656e7cc9d200c3491e89ecd56ce0d3efb76383c6e7cb8694
SHA512: 63fe6bf7233af261da14748a5b2d7e3bcdc7b5c7a3c270899c3b301bde2c9860abcd86e0c3eab4e5751c64311ca78dce1668831a0fb4dc933935e816b54bf450
```

- **Como se observa, hay diferencias en la longitud de las firmas generadas, para mayor seguridad es mejor las de mayor longitud.**

```python
# Biblioteca para manejo de números aleatorios
import random

# Biblioteca para usar funciones "hash"
import hashlib

# Conjunto de caracteres posibles para generar una cadena aleatoria
caracteres = "abcdefghijklmnopqrstuvwxyz0123456789"

# Función para generar una cadena aleatoria de longitud 10
def generar_cadena():
    cadena = ""
    for ciclo in range(0,10):
        cadena = cadena + random.choice(caracteres)
    return cadena

# Llama a la función anterior para tener una cadena aleatoria
cadena = generar_cadena()

# Generar las Firmas Digitales usando diferentes funciones
firma_md5 = (hashlib.md5(cadena.encode())).hexdigest()
firma_sha1 = (hashlib.sha1(cadena.encode())).hexdigest()
firma_sha224 = (hashlib.sha224(cadena.encode())).hexdigest()
firma_sha256 = (hashlib.sha256(cadena.encode())).hexdigest()
firma_sha384 = (hashlib.sha384(cadena.encode())).hexdigest()
firma_sha512 = (hashlib.sha512(cadena.encode())).hexdigest()

# Impresión en pantalla de todas las firmas generadas
print("\nGeneración de Firmas Digitales")
print("\nPosibles caracteres que llevará la firma: " + caracteres)
print("Cadena aleatoria generada de una longitud de 10: " + cadena)
print("\nMD5: " + firma_md5)
print("SHA1: " + firma_sha1)
print("SHA224: " + firma_sha224)
print("SHA256: " + firma_sha256)
print("SHA384: " + firma_sha384)
print("SHA512: " + firma_sha512)
```

Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia

rogelio.fe@morelia.tecnm.mx

rogeplus@gmail.com

xumarhu.net

@rogeplus

https://www.youtube.com/channel/UC0on88n3LwTKxJb8T09sGjg

rogelioferreiraescutia