2022-02-23 - Implementación de one time pad

Cifrado y descifrado

Cadenas

```
def main():
    vector = "Amado Garcia was here"
    encrypted = encrypt(vector)
    decrypted = decrypt(encrypted[0], encrypted[1])

    print("Test Vector: " + vector)
    print("OTP: " + encrypted[0])
    print("Encrypted: " + encrypted[1])
    print("Decrypted: " + decrypted)


def encrypt(plaintext):
    """Encrypt plaintext value.
    Keyword arguments:
    plaintext -- the plaintext value to encrypt.
    """
    otp = "".join(random.sample(charset, len(charset)))
    result = ""

    for c in plaintext.upper():
        if c not in otp:
            continue
        else:
            result += otp[charset.find(c)]

    return otp, result


def decrypt(otp, secret):
    """Decrypt secret value.
    Keyword arguments:
    otp -- the one-time pad used when the secret value was encrypted.
    secret -- the value to be decrypted.
    """
    result = ""

    for c in secret.upper():
        if c not in otp:
            continue
        else:
            result += charset[otp.find(c)]

    return result


if __name__ == "__main__":
    main()
```

```
Test Vector: Amado Garcia was here
OTP: S5D8NQVOAY42I76XLK301CPFETJ9WBGMRHZU
Encrypted: SIS86VSKDASPS3ONKN
Decrypted: AMADOGARCIAWASHERE
```

```python
def main():
    vector = "Me Muero de sueenio"
    encrypted = encrypt(vector)
    decrypted = decrypt(encrypted[0], encrypted[1])

    print("Test Vector: " + vector)
    print("OTP: " + encrypted[0])
    print("Encrypted: " + encrypted[1])
    print("Decrypted: " + decrypted)


def encrypt(plaintext):
    """Encrypt plaintext value.
    Keyword arguments:
    plaintext -- the plaintext value to encrypt.
    """
    otp = "".join(random.sample(charset, len(charset)))
    result = ""

    for c in plaintext.upper():
        if c not in otp:
            continue
        else:
            result += otp[charset.find(c)]

    return otp, result


def decrypt(otp, secret):
    """Decrypt secret value.
    Keyword arguments:
    otp -- the one-time pad used when the secret value was encrypted.
    secret -- the value to be decrypted.
    """
    result = ""

    for c in secret.upper():
        if c not in otp:
            continue
        else:
            result += charset[otp.find(c)]

    return result


if __name__ == "__main__":
    main()
```

```
Test Vector: Me Muero de sueenio
OTP: OQ3UMAT6V1NI9W4YFH5G0RBSCLK2EPD7JZX8
Encrypted: 9M90MH4UM50MMWV4
Decrypted: MEMUERODESUEENIO
```

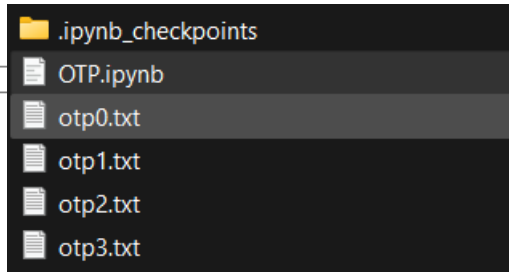
Archivos

```python
        else:
            print("Invalid option \n")

menu()
```

```
Option 1. Create OTP
Option 2. Encrypt message
Option 3. Decrypt message
Option 4. Exit program

Choose your number option:
```




```
How many OTP do you want 10
Maximum message length 50
Option 1. Create OTP
```

otp1.txt - Note

File Edit Vi

```
24
74
50
64
24
64
23
75
18
4
14
65
56
69
66
48
1
57
24
75
6
6
17
39
56
```

otp0.txt -

File Edit

```
23
41
70
76
79
73
29
71
11
73
10
15
6
67
53
6
73
38
16
38
68
60
19
54
13
```

otp2.txt - Notepad

File Edit View

```
59
49
43
51
64
10
5
19
44
73
58
44
69
18
8
53
65
3
13
12
76
68
10
74
65
```

```
Choose your number option: 2
Filename of the OTP you want: otp2.txt
Enter your message: ayudatengosuenio
Name of output encrypted file: ayuda.txt
Option 1. Create OTP
Option 2. Encrypt message
Option 3. Decrypt message
Option 4. Exit program

Choose your number option:
```

```
Choose your number option: 3
Filename of the OTP you: otp2.txt
Output file name (decrypted): ayuda.txt
The decrypted message is:ayudatengosuenio
Option 1. Create OTP
Option 2. Encrypt message
Option 3. Decrypt message
Option 4. Exit program
Choose your number option: 4
Bye
```