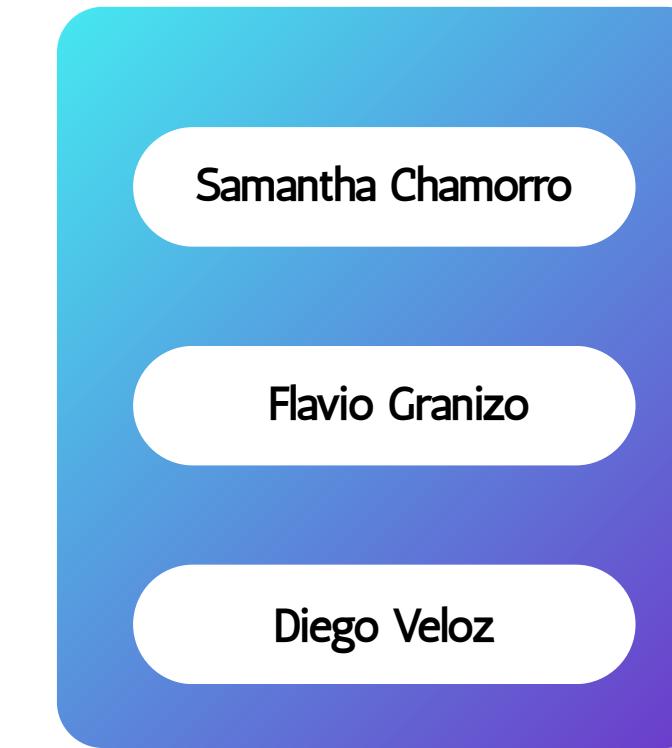


Generador Seguro de contraseñas



Strong Password Generator

qué hace nuestro Generador ?

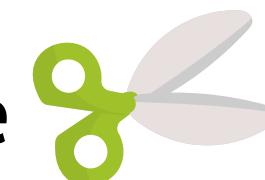
Crea contraseñas personalizadas basadas en una palabra clave del usuario,, con 20 caracteres exactos. Diviendo la palabra en 3 partes y combinándola con letras aleatorias y números entre ellas.

Longitud Fija



Siempre 20 caracteres

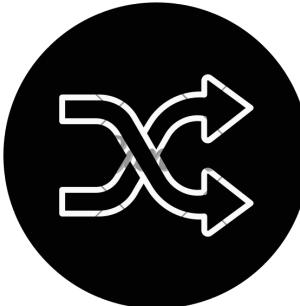
División Inteligente



La palabra se divide en 3 partes iguales.



Ruido Aleatorio



Añade caracteres random entre secciones.

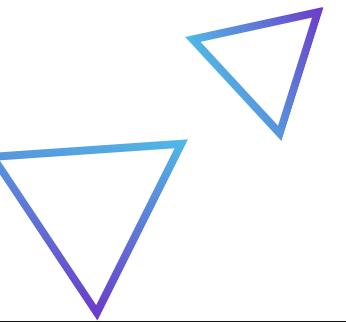


Alta seguridad



Incluye letras, números y símbolos.

Como funciona?



```
1 import random
2 import string
3 caracteres_mix = string.ascii_letters + string.digits + string.punctuation
4 palabra_cl = input('Ingrese una palabra para su clave: ').upper()
5
6 palabra_cl = palabra_cl[:9]
7
8 longitud = len(palabra_cl)
9 corte1 = longitud // 3
10 corte2 = 2 * (longitud // 3)
11
12 parte1 = palabra_cl[:corte1]
13 parte2 = palabra_cl[corte1:corte2]
14 parte3 = palabra_cl[corte2:]
15
16 secciones_palabra = [parte1, parte2, parte3]
17
18 partes_finales = []
19
20 espacio_sobrante = 20 - len(palabra_cl) - 2
21
22 for parte in secciones_palabra:
23
24     cantidad_random = espacio_sobrante // 3
25
26     ruido = "".join(random.choices(caracteres_mix, k=cantidad_random))
27
28     bloque = ruido + parte
29
30     partes_finales.append(bloque)
31
32
33
34 partes_finales.append(bloque)
35
36
```



1.. Entrada de Palabra Clave y Límite

El usuario ingresa una palabra (máximo 9 caracteres) que se convierte a MAYÚSCULAS. → `palabra_cl = palabra_cl[:9]`

2. División en 3 partes

La palabra se divide en tres secciones iguales usando división entera

Ejemplo: "SEGURIDAD" → "SEG" + "URI" + "DAD"

3. Generación de Ruido Aleatorio

Añade caracteres aleatorios (letras, números y símbolos) antes de cada parte.

`caracteres_mix = ascii_letters + digits + punctuation`

4. Ensamblaje Final

Une las 3 secciones con guiones (-) y ajusta a exactamente 20 caracteres

Flujo del Algoritmo

1.. Captura y Limitacion → `palabra_cl = input().upper()[:9]`

Máximo 9 caracteres para optimizar espacio

2. Calculo de Cortes → `corte1 = longitud // 3
corte2 = 2 * (longitud // 3)`

División entera para 3 partes iguales.

3. Division en secciones → `parte1 = palabra[:corte1]
parte2 = palabra[corte1:corte2]
parte3 = palabra[corte2:]`

4. Generacion de ruido → `espacio_sobrante = 20 - len(palabra_cl) - 2
ruido = random.choices(caracteres_mix, k=cantidad)`

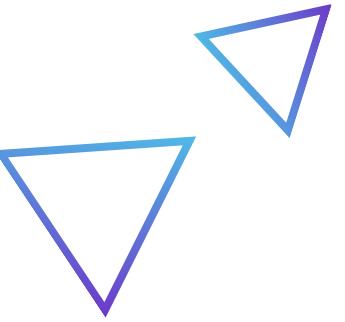
Calcula espacio disponible y lo distribuye

5. Ensamblaje final → `clave_segura = "-".join(partes_finales)[:20]`

Une con guiones y ajusta a 20 caracteres exactos.



Lenguaje y estructuras



Lenguaje utilizado: PYTHON - 3

- **Biblioteca: random**
- random.choice(): Genera múltiples caracteres aleatorios.

Python



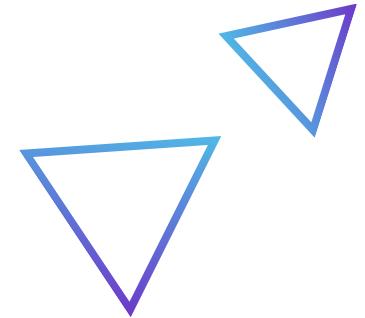
Random String

- **Biblioteca: string**
 - ascii_letters: Todas las letras (a-z, A-Z)
 - digits : 0 - 9
 - punctuation: !@#\$%& * ...

- **Estructuras utilizadas:**

- Slicing: Palabra [:9]
- División entera : longitud // 3
- join() : Unión de strings

Calculo del Espacio para Ruido



Fórmula del Espacio Disponible:

```
espacio_sobrante = 20 -  
len(palabra) - 2
```

20: Longitud total deseada

len(palabra): Caracteres de tu palabra

2: Espacio para los 2 guiones separadores (-)

Ejemplos de Cálculo:

Palabra: "SOL" (3 caracteres)
espacio_sobrante = 20 - 3 - 2 = 15

caracteres

Por sección: $15 \div 3 = 5$ caracteres aleatorios

Palabra: "SEGURIDAD" (9 caracteres)
espacio_sobrante = 20 - 9 - 2 = 9

caracteres

Por sección: $9 \div 3 = 3$ caracteres aleatorios

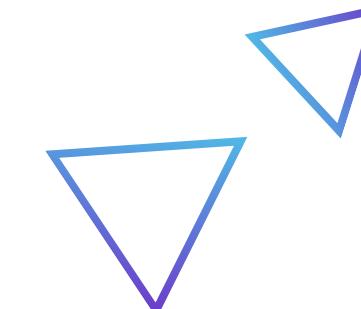
Demostracion Practica.

Entrada:

SEGURIDAD

9 caracteres → Se mantiene completa

Cada combinación es única y aleatoria, pero mantiene tu palabra clave para ayudarte a recordarla.



División en 3 partes:

Parte 1

SEG

Parte 2

URI

Parte 3

DAD

Añadir Ruido Aleatorio:

Ruido + Parte
1

aB3SEG

Ruido + Parte
2

x9!URI

Ruido + Parte
3

K7@DAD

Resultado Final (20 caracteres):

aB3SEG-x9!URI-K7@DAD

✓ Longitud exacta: 20 caracteres

Longitud Estandarizada

Todas las contraseñas tienen exactamente 20 caracteres..

Distribucion Inteligente

El ruido se distribuye equitativamente en 3 secciones

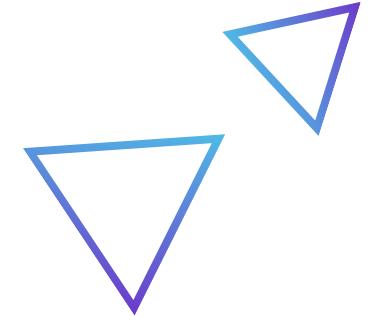
Alta Complejidad

Incluye 94 tipos diferentes de caracteres posibles

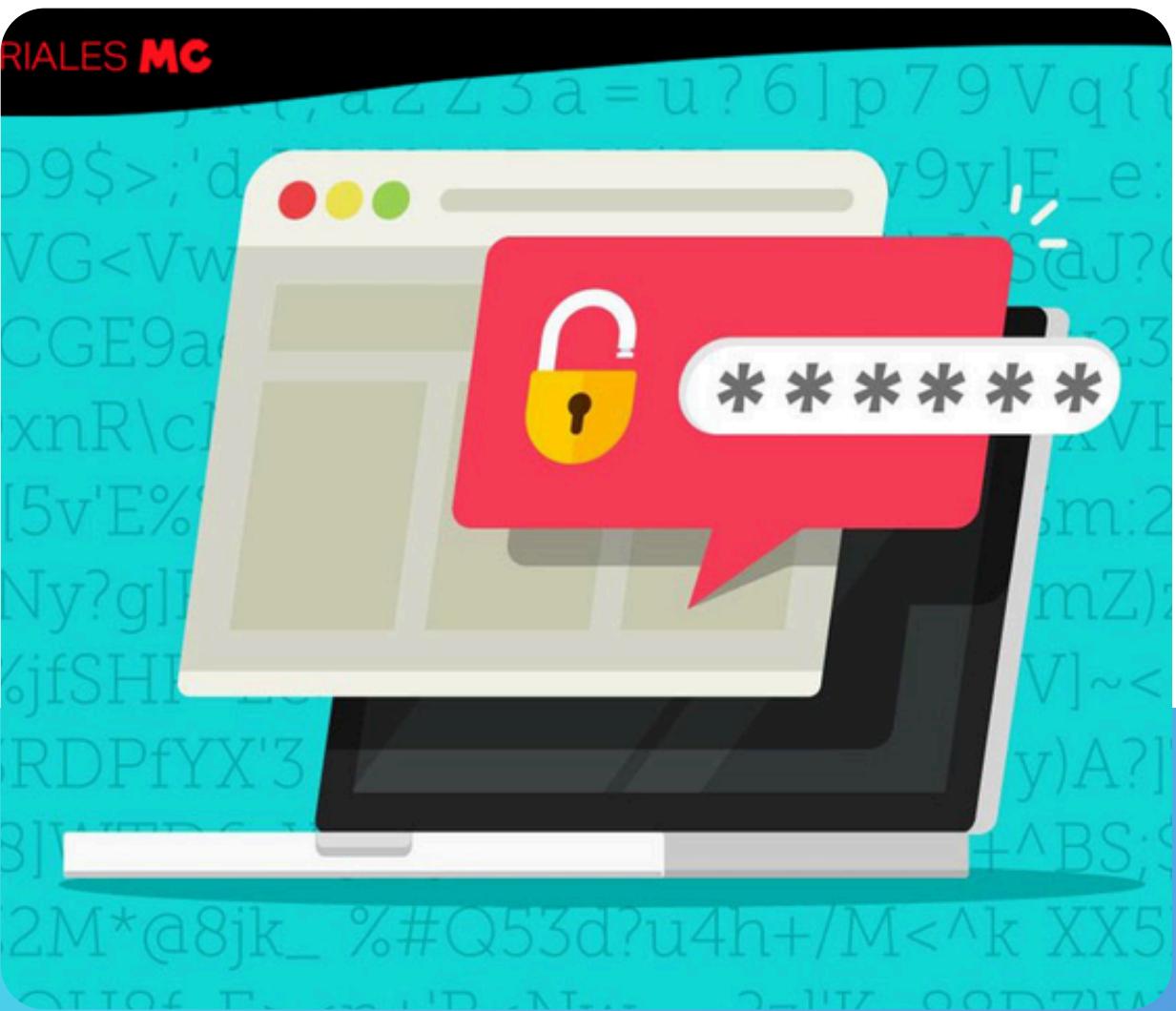
Memorabilidad

Tu palabra clave está presente para ayudarte a recordar

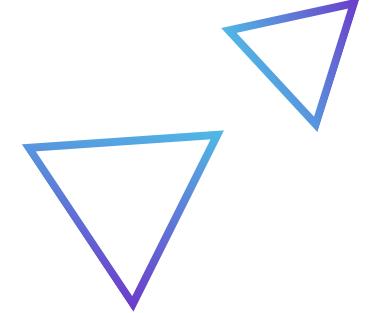
Logros del Proyecto



Conclusiones



- El algoritmo genera contraseñas de longitud fija (20 caracteres) con alta seguridad.
- Implementación exitosa de división de strings y cálculo de espacios.
- La combinación de palabra memorable + ruido aleatorio ofrece seguridad y usabilidad.



thank

you!

