

Generador de contraseñas seguras

Fecha de Creación: 18 de Mayo del 2025

Objetivo

Crear un juego para que el usuario aprenda a crear contraseñas seguras, ya sea generándolas automáticamente o escribiéndolas manualmente, y reciba retroalimentación inmediata sobre su nivel de seguridad. A través de intentos limitados, el programa promueve la conciencia sobre buenas prácticas de ciberseguridad de forma interactiva y educativa.

Librerías

- **random** la uso para que el sistema elija letras, números y símbolos al azar.
- **string** me da acceso a conjuntos como todas las letras, los dígitos y los signos especiales.

🚦 Variables globales

```
NIVELES = ("Débil", "Media", "Fuerte")
historial = [ ]
MAX_INTENTOS = 3
usuario_actual = ""
```

- Se creo una tupla con los tres niveles posibles de seguridad.
- En **historial** es una lista donde voy a guardar cada contraseña con sus datos.
- **MAX_INTENTOS** es el límite de intentos que tendrá el usuario al escribir contraseñas manualmente.
- Y **usuario_actual** guarda el nombre de la persona que está jugando o usando el programa.

🚦 Función para ingresar el nombre del usuario

```
def obtener_nombre_usuario():
    global usuario_actual
    usuario_actual = input("Por favor, ingrese su nombre de usuario: ").strip().capitalize()
    print(f"\n¡Hola, {usuario_actual}! Comencemos...\n")
```

- Esta función le pide al usuario que escriba su nombre.
- Lo guarda en una variable global y lo capitaliza para que se vea bonito.
- Luego lo uso para mostrarlo en pantalla y también para registrar a quién pertenece cada contraseña.

🚦 Función para mostrar el menú

```
def mostrar_menu():  
    print("...")
```

Esta función simplemente muestra el menú principal. Le permite al usuario elegir qué quiere hacer: ver reglas, generar, escribir, ver historial o salir.

🚦 Función para mostrar las reglas

```
def mostrar_reglas():  
    print("...")
```

Aquí explico qué es el juego: el objetivo es hacer contraseñas seguras. Se aclara que mientras más fuerte la contraseña, más puntos gana. Y que, si el usuario escribe una manualmente, solo puede hacerlo 3 veces por intento.

🚦 Generar una contraseña aleatoria

```
def generar_contraseña(longitud=12):  
    caracteres = string.ascii_letters + string.digits + string.punctuation  
    return "".join(random.choice(caracteres) for _ in range(longitud))
```

Esta función crea una contraseña nueva de 12 caracteres al azar. Mezcla letras, números y símbolos, y selecciona uno por uno usando un bucle con `random.choice`.

🚦 Evaluar la seguridad de una contraseña

```
def evaluar_contraseña(contraseña):  
    puntaje = 0  
    if len(contraseña) >= 8: puntaje += 2  
    if any(c.islower() for c in contraseña): puntaje += 2  
    if any(c.isupper() for c in contraseña): puntaje += 2  
    if any(c.isdigit() for c in contraseña): puntaje += 2  
    if any(c in string.punctuation for c in contraseña): puntaje += 2  
    return puntaje
```

Esta parte evalúa qué tan buena es la contraseña. Por cada cosa positiva suma 2 puntos:

- si tiene al menos 8 caracteres,
- si tiene minúsculas,
- mayúsculas,
- números,
- y símbolos.

El puntaje máximo posible es 10.

Determinar el nivel de seguridad

```
def determinar_nivel(puntaje):  
    if puntaje <= 4: return NIVELES[0]  
    elif puntaje <= 7: return NIVELES[1]  
    else: return NIVELES[2]
```

Según el puntaje anterior, esta función clasifica la contraseña como débil, media o fuerte. Lo hace usando la tupla **NIVELES**.

Procesar y guardar los datos de una contraseña

```
def procesar_contraseña(contraseña, generada=False):  
    ...
```

Esta función es una de las más importantes porque:

- Evalúa la contraseña.
- Determina el nivel de seguridad.
- Muestra los resultados en pantalla.
- Y guarda todo en el historial como un diccionario, incluyendo el nombre del usuario.

Mostrar el historial completo

```
def mostrar_historial():
```

Aquí muestro todas las contraseñas evaluadas, junto con el nombre del usuario, el puntaje y el nivel.

Lo saco de la lista **historial** que fui llenando cada vez que alguien genera o escribe una contraseña.

Bucle principal del programa

```
while True:  
    mostrar_menu()  
    ...
```

Esta es la parte que mantiene el programa funcionando. El menú se repite hasta que el usuario elija la opción 5 para salir.

Dentro del ciclo, el programa reacciona según la opción seleccionada:

- Si elige 2, se genera una contraseña automática.
- Si elige 3, puede escribir hasta 3 contraseñas manuales.
- Y en la opción 4, ve el historial de todo lo que se ha hecho.

Si se pasan los 3 intentos, vuelve al menú y puede volver a intentar.

Link del video

[VIDEO DE PRESENTACION DEL JUEGO DE CONTRASEÑAS.mp4](#)