



**Facultad de Ingeniería.
Ingeniería de Sistemas.**

Asignatura:

Inteligencia Artificial – Electiva CP (2665)

Ejercicios Python

Presenta:

Johnier Alfonso Sánchez Cárdenas ID: 548559.

Docente:

Segundo Fidel Puerto Garavito

Colombia, Ciudad Bogotá D.C.

Octubre 16, de 2020.

Ejercicio N°1

Crear un módulo para validación de nombres de usuarios. Dicho módulo, deberá cumplir con los siguientes criterios de aceptación:

- El nombre de usuario debe contener un mínimo de 6 caracteres y un máximo de 12
- El nombre de usuario debe ser alfanumérico
- Nombre de usuario con menos de 6 caracteres, retorna el mensaje "El nombre de usuario debe contener al menos 6 caracteres"
- Nombre de usuario con más de 12 caracteres, retorna el mensaje "El nombre de usuario no puede contener más de 12 caracteres"
- Nombre de usuario con caracteres distintos a los alfanuméricos, retorna el mensaje "El nombre de usuario puede contener solo letras y números"
- Nombre de usuario válido, retorna True

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Oct 16 18:12:36 2020
4
5 @author: Johnier Sanchez
6 """
7
8 import EjercicioUno
9
10 def nombreUsuario(nombreIngresado):
11
12     # Se calcular la longitud del nombre de usuario.
13     long=len(nombreIngresado)
14     # Se calcula que la cadena contenga valores alfanuméricos.
15     x=nombreIngresado.isalnum()
16     # Se presenta mensaje cuando la cadena contiene valores no alfanuméricos.
17     if x== False:
18         print("El nombre de usuario puede contener solo letras y números")
19     # Se presenta mensaje cuando la cadena contiene menos de 6 caracteres.
20     if long < 6:
21         print("El nombre de usuario debe contener al menos 6 caracteres")
22     # Se presenta mensaje cuando la cadena contiene mas de 12 caracteres.
23     if long > 12:
24         print("El nombre de usuario no puede contener más de 12 caracteres")
25     # Se valida cantidad de caracteres.
26     if long >5 and long <13 and x ==True:
27         # Se retorna verdadero, si el tamaño es mayor a 5 y menor a 13.
28         return True
29
30 def main():
31
32     # Se inicializa la variable booleana.
33     ciclo=False
34     # Se crea bucle para la presentación de los mensajes.
35     while ciclo==False:
36         # Se imprime petición de nombre.
37         nombreIngresado=input("Ingrese nombre de usuario: ")
38         # Se realizan las validaciones especificadas.
39         if EjercicioUno.nombreUsuario(nombreIngresado) == True:
40             # Se muestra mensaje exitoso de creación.
41             print("Usuario creado exitosamente")
42             # Se termina ciclo while.
43             ciclo=True
44
45 main()
```

Ingrese nombre de usuario: johnier123
Usuario creado exitosamente

Enlace GitHub: <https://github.com/JohnierSanchez/Python.git>

Ejercicio N°2

Crear un módulo para validación de contraseñas. Dicho módulo, deberá cumplir con los siguientes criterios de aceptación:

- La contraseña debe contener un mínimo de 8 caracteres
- Una contraseña debe contener letras minúsculas, mayúsculas, números y al menos 1 carácter no alfanumérico
- La contraseña no puede contener espacios en blanco
- Contraseña válida, retorna True
- Contraseña no válida, retorna el mensaje "La contraseña elegida no es segura"

```
1 #-*- coding: utf-8 -*-
2 """
3 Created on Fri Oct 16 18:38:31 2020
4
5 @author: Johnier Sanchez
6 """
7 import EjercicioDos
8
9 def contraseñaUsuario(contraseñaIngresada):
10
11     # Se inicializa la variable booleana que controla el cumplimiento de todos los requisitos.
12     validador=False
13     # Se calcula la longitud de la contraseña.
14     long=len(contraseñaIngresada)
15     # Se inicializa la variable para identificar los espacios.
16     espacio=False
17     # Se inicializa la variable para identificar letras mayúsculas.
18     mayuscula=False
19     # Se inicializa la variable para identificar letras minúsculas.
20     minuscula=False
21     # Se inicializa la variable para identificar números.
22     numeros=False
23     # Se valida si hay caracteres alfanumérica y se retorna True.
24     y=contraseñaIngresada.isalnum()
25     # Se inicializa la variable para verificar que hayan mayuscula, minuscula, numeros y no alfanuméricos
26     aceptable=True
27
28     # Se genera ciclo for para recorrer caracter por caracter en la contraseña ingresada.
29     for carac in contraseñaIngresada:
30         # Se valida si el caracter es un espacio
31         if carac.isspace()==True:
32             #si encuentra un espacio se cambia el valor user
33             espacio = True
34         # Se valida si hay mayuscula
35         if carac.isupper()==True:
36             #acumulador o contador de mayusculas
37             mayuscula=True
38         # Se valida si hay minúsculas
39         if carac.islower()==True:
40             #acumulador o contador de minúsculas
41             minuscula=True
42         # Se valida si hay números
43         if carac.isdigit()==True:
44             #acumulador o contador de numeros
45             numeros = True |
46
47     # Se valida si hay espacios en blanco
48     if espacio == True:
49         print("La contraseña no puede contener espacios")
50     else:
51         #se cumple el primer requisito que no hayan espacios
52         validador = True
53
54     if long <8 and validador == True:
55         print("Mínimo 8 caracteres")
56         #cambia a False si no se cumple el requisito mínimo de caracteres
57         validador = False
58
59     if mayuscula == True and minuscula == True and numeros == True and y == False and validador == True:
```

```

60         #Cumple el requisito de tener mayuscula, minuscula, numeros y no alfanuméricos
61         validador = True
62     else:
63         #uno o mas requisitos de mayuscula, minuscula, numeros y no alfanuméricos no se cumple
64         acceptable=False
65
66     if validador == True and acceptable == False:
67         print("La contraseña elegida no es segura")
68
69     if validador == True and acceptable == True:
70         return True
71
72 def main():
73
74     # Se inicializa la varibale booleana.
75     ciclo=False
76     # Se crea blucle para la presentación de los mensajes.
77     while ciclo == False:
78         # Se imprime petición de Contraseña.
79         contraseñaIngresada=input("Ingrese su Contraseña: ")
80         # Se realizan las validaciones especificadas.
81         if EjercicioDos.contraseñaUsuario(contraseñaIngresada) == True:
82             # Se muestra mensaje exitoso de creación.
83             print("Contraseña creada exitosamente")
84             # Se termina ciclo while.
85             ciclo = True
86
87 main()

```

Enlace GitHub: <https://github.com/JohnierSanchez/Python.git>