

EVALUACIÓN DE SOFTWARE

Tarea UT01

Rebeca Vargas Macías
2024/2025

ÍNDICE

1. INTRODUCCIÓN.....	0
2. SCRIPT CADENA PALÍNDROMA.....	0
3. PROGRAMA DE TESTING	2
4. VERIFICACIÓN DE SOFTWARE	3

1. INTRODUCCIÓN

- **Apartado 1: Crea un script que compruebe si una cadena introducida por el teclado es palíndroma.**

Para realizar esta tarea es obligatorio que uses el lenguaje Python 3, por su facilidad y por el gran acceso a librerías de evaluación de software que posee. Si aún no has programado en Python su inmersión te será sencilla y amigable.

- Llamaremos a este script charfun.py
- Deberías de crear un programa con su estructura completa
- Dentro de este programa crearemos una función llamada *esPalindromo* que determinará si una cadena pasada como parámetro es palíndromo o no.
- El programa pedirá al usuario una frase y el programa determinará si es palíndromo o no.
-

Apartado 2: Creación del programa de Testing

Crearemos un programa principal donde definiremos una clase llamada Test donde probaremos nuestro software.

- Importaremos la librería de texto de software en este caso *unittest*.
- Crearemos nuestra clase (del tipo *unittest.TestCase*)
- Dentro de esta clase definiremos una función (podemos llamarla como consideremos pero es recomendable un nombre ilustrativo)
- Dentro de la función reflejaremos el tipo de testeo que vamos a realizar. En este caso nuestro objetivo es verificar si "Anita lava la tina" es palíndromo.
- No olvides incluir todas aquellas comprobaciones que creas necesarias para testear que la función *esPalindromo* funciona correctamente. No des por sentado que el programador puede hacer un "buen uso" de ella. Prepárala para "lo peor".
- **Apartado 3: Verificación de software y pregunta final**

- Ejecutaremos el programa final y verificaremos si realmente nuestro programa charfun.py se comporta como esperamos.
- Si el programa que comprueba el código detecta un error nos reflejará que dato está esperando y que ha recibido
- **Pregunta final** ¿Qué tipo de prueba hemos realizado?

2. SCRIPT CADENA PALÍNDROMA

A continuación, vamos a crear un script con el lenguaje Python, que nos indica cuando una cadena es palíndroma, es decir, que se lea igual de izquierda a derecha, que de derecha a izquierda.

Como razonamiento previo, entendemos que para que se recorra la cadena en ambas direcciones, no debe haber signos de puntuación. De este modo, la cadena se leerá como una línea continua, pudiendo “unir” las sílabas, aunque esté en otra palabra.

Por otro lado, todos los caracteres deberán estar en minúscula o mayúscula. Esto lo hacemos para que el palíndromo se verifique sin problemas y el script no sea sensible a este tipo de caracteres y no reconozca la palabra como palíndroma. En este caso elegiremos minúscula ya que la mayoría de los caracteres irán ya de este modo (es el formato normal de una oración).

Por último, pedimos que se compare la frase inicial, con la que se ha dado la vuelta.

Como resumen, los pasos a seguir serían:

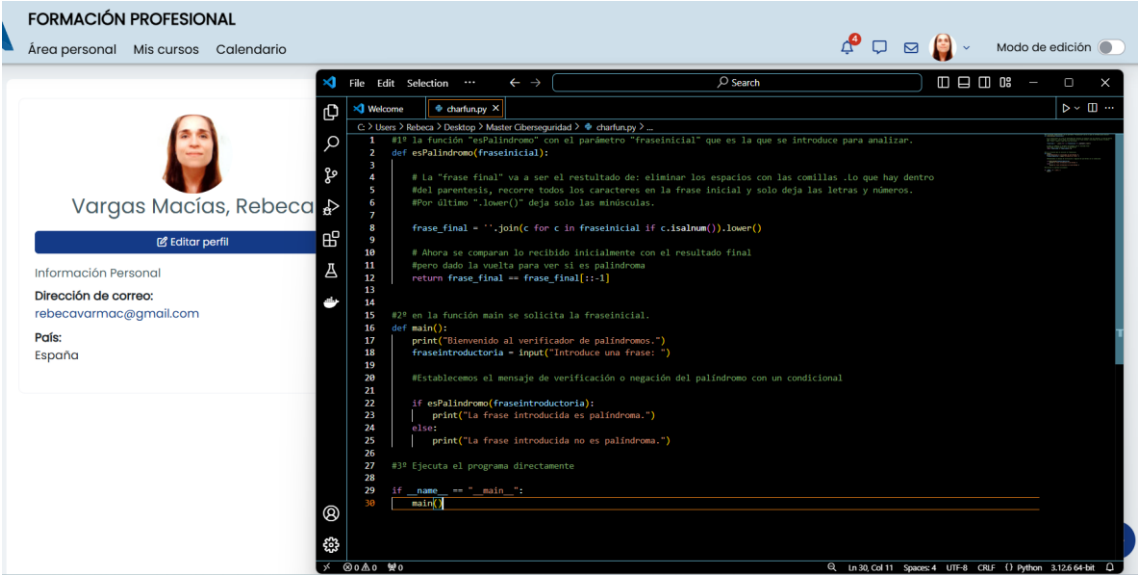
1º) Creación de la función esPalindromo, donde recogeremos lo expresado anteriormente.

2º) Definimos una función principal (main) donde se solicite la frase a analizar y la condición de si es palíndroma o no. Se llamará a esPalindromo

3º) Ejecutamos el script

Una vez analizados estos requerimientos, abrimos Visual Studio Code, donde podremos crear el código, ya que reconoce muchos tipos de lenguaje y luego lo podemos ejecutar en él mismo.

Podemos elegir otros editores de texto como Notepad++ que también los reconoce y luego lanzarlo en el terminal con Phyton, pero lo vamos a realizar con Visual Studio Code.



```
1 #1º la función "esPalindromo" con el parámetro "fraseinicial" que es la que se introduce para analizar.
2 def esPalindromo(fraseinicial):
3
4     # La "frase final" va a ser el resultado de: eliminar los espacios con las comillas .to que hay dentro
5     #del parentesis, recorre todos los caracteres en la frase inicial y solo deja las letras y números.
6     #Por último ".lower()" deja solo las minúsculas.
7
8     frase_final = ''.join(c for c in fraseinicial if c.isalnum()).lower()
9
10    # Ahora se comparan lo recibido inicialmente con el resultado final
11    #pero dado la vuelta para ver si es palíndroma
12    return frase_final == frase_final[::-1]
13
14
15 #2º en la función main se solicita la fraseinicial.
16 def main():
17     print("Bienvenido al verificador de palíndromos.")
18     fraseintrodutoria = input("Introduce una frase: ")
19
20     #Establecemos el mensaje de verificación o negación del palíndromo con un condicional
21
22     if esPalindromo(fraseintrodutoria):
23         print("La frase introducida es palíndroma.")
24     else:
25         print("La frase introducida no es palíndroma.")
26
27 #3º Ejecuta el programa directamente
28 if __name__ == "__main__":
29     main()
```

En los comentarios se explica con más detalle cada línea de código.

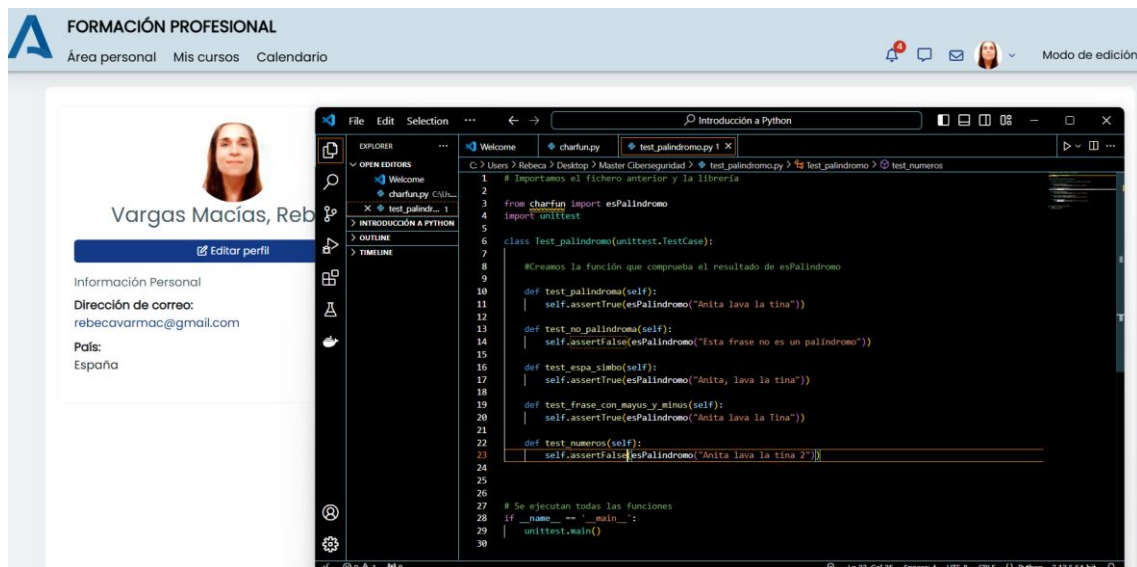
3. PROGRAMA DE TESTING

Importamos la librería unittest y creamos la clase Test de tipo unittest.TestCase donde definiremos nuestras funciones de comprobación, que ejecuta esPalindromo en cada una.

Como se indica que debemos incluir todas las comprobaciones posibles, vamos a definir 5 funciones que comprueban todo lo explicado en el punto 2:

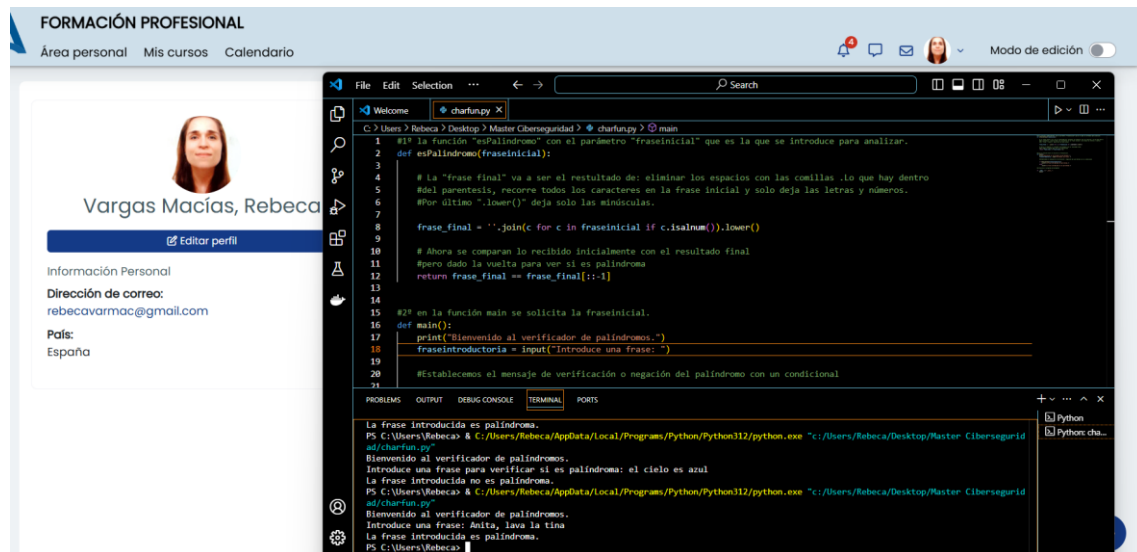
1. Afirmativa de que lo es (función: test_palindroma)
2. Negativa de que no lo es (función: test_no_palindroma)
3. Inclusión de espacios y símbolos (función: test_espa_simbo)
4. Mayúsculas y minúsculas (función: test_mayus_minus)
5. Números (función: test_numero)

Después de definirlas las comprobaremos con assertTrue o assertFalse y comprobaremos si se cumplen.



4. VERIFICACIÓN DE SOFTWARE

Ejecutamos charfun.py y probamos el script con esta frase: “Anita, lava la tina” (palíndroma) y con: “El cielo es azul” (no palíndroma). Los resultados son los esperados.



```
1 #! la función "esPalindromo" con el parámetro "fraseinicial" que es la que se introduce para analizar.
2 def esPalindromo(fraseinicial):
3
4     # La "frase final" va a ser el resultado de: eliminar los espacios con las comillas .lo que hay dentro
5     #del parentesis, recorre todos los caracteres en la frase inicial y solo deja las letras y números.
6     #Por último ".lower()" deja solo las minúsculas.
7
8     frase_final = ''.join(c for c in fraseinicial if c.isalnum()).lower()
9
10    # Ahora se comparan lo recibido inicialmente con el resultado final
11    #pero dado la vuelta para ver si es palindroma
12    return frase_final == frase_final[::-1]
13
14
15 #! en la función main se solicita la fraseinicial.
16 def main():
17     print("Bienvenido al verificador de palíndromos.")
18     fraseintrodutoria = input("Introduce una frase: ")
19
20     #Establecemos el mensaje de verificación o negación del palíndromo con un condicional
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

La frase introducida es palíndroma.

PS C:\Users\Rebeca> & C:\Users\Rebeca\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Rebeca/Desktop/Master Ciberseguridad/ad/charfun.py"

Bienvenido al verificador de palíndromos.

Introduce una frase para verificar si es palíndroma: el cielo es azul

La frase introducida no es palíndroma.

PS C:\Users\Rebeca> & C:\Users\Rebeca\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Rebeca/Desktop/Master Ciberseguridad/ad/charfun.py"

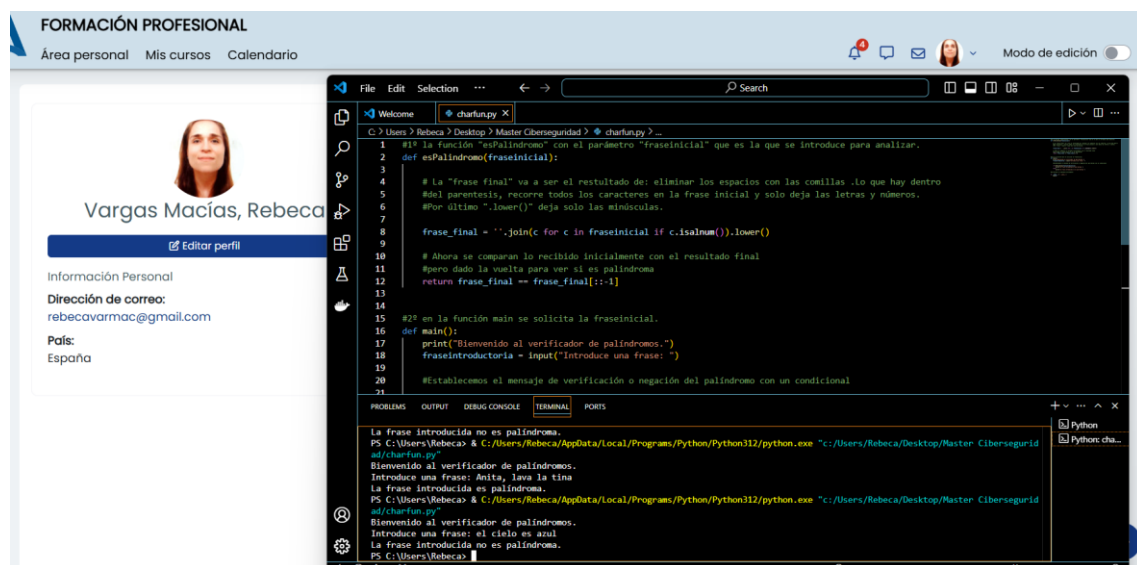
Bienvenido al verificador de palíndromos.

Introduce una frase: Anita, lava la tina

La frase introducida es palíndroma.

PS C:\Users\Rebeca>

La primera es palíndroma.



```
1 #! la función "esPalindromo" con el parámetro "fraseinicial" que es la que se introduce para analizar.
2 def esPalindromo(fraseinicial):
3
4     # La "frase final" va a ser el resultado de: eliminar los espacios con las comillas .lo que hay dentro
5     #del parentesis, recorre todos los caracteres en la frase inicial y solo deja las letras y números.
6     #Por último ".lower()" deja solo las minúsculas.
7
8     frase_final = ''.join(c for c in fraseinicial if c.isalnum()).lower()
9
10    # Ahora se comparan lo recibido inicialmente con el resultado final
11    #pero dado la vuelta para ver si es palindroma
12    return frase_final == frase_final[::-1]
13
14
15 #! en la función main se solicita la fraseinicial.
16 def main():
17     print("Bienvenido al verificador de palíndromos.")
18     fraseintrodutoria = input("Introduce una frase: ")
19
20     #Establecemos el mensaje de verificación o negación del palíndromo con un condicional
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

La frase introducida no es palíndroma.

PS C:\Users\Rebeca> & C:\Users\Rebeca\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Rebeca/Desktop/Master Ciberseguridad/ad/charfun.py"

Bienvenido al verificador de palíndromos.

Introduce una frase: Anita, lava la tina

La frase introducida es palíndroma.

PS C:\Users\Rebeca> & C:\Users\Rebeca\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Rebeca/Desktop/Master Ciberseguridad/ad/charfun.py"

Bienvenido al verificador de palíndromos.

Introduce una frase: el cielo es azul

La frase introducida no es palíndroma.

PS C:\Users\Rebeca>

La segunda no lo es.

Si usamos el test para comprobar los resultados, también es el correcto:



Vargas Macías, Rebekah

[Editar perfil](#)

Información Personal

Dirección de correo:
rebecavarmac@gmail.com

País:
España

Información Personal

rebecavarmac@gmail.com

País:
España

País:

España

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the file structure, with 'test_palindromo.py' selected under the 'test' folder. The main editor area shows the code for 'test_palindromo.py', which includes a class 'TestPalindromo' with two methods: 'test_frase' and 'test_numeros'. The bottom status bar indicates the file is open in the 'Python' interpreter.

```

19 def test_frase_con_mayus_y_minus(self):
20     | self.assertTrue(esPalindromo("Anita lava la TINA"))
21
22 def test_numeros(self):
23     | self.assertFalse(esPalindromo("Anita lava la tina 2"))
24
25
26
27 # Se ejecutan todas las funciones
28 if __name__ == '__main__':

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Run 5 tests in 0.001s

FAILED (failures=1)

PS C:\Users\Rebeca\Desktop\Introducción a Python> C:\Users\Rebeca\AppData\Local\Programs\Python\Python312\python.exe "c:\Users\Rebeca\Desktop\Master Ciberseguridad\test_palindromo.py"

.....

Run 5 tests in 0.000s

OK

PS C:\Users\Rebeca\Desktop\Introducción a Python>

