Rebeca Vargas Macías

2024/2025

EVALUACIÓN DE SOFTWARE

Tarea UT01

ÍNDICE

[**1.** **INTRODUCCIÓN** 0](#_Toc183729479)

[**2.** **SCRIPT CADENA PALÍNDROMA** 0](#_Toc183729480)

[**3.** **PROGRAMA DE TESTING** 2](#_Toc183729481)

[**4.** **VERIFICACIÓN DE SOFTWARE** 3](#_Toc183729482)

# **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 funciona 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 palíndroma 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.TextCase)
* 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 palíndroma.
* 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?

## **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:

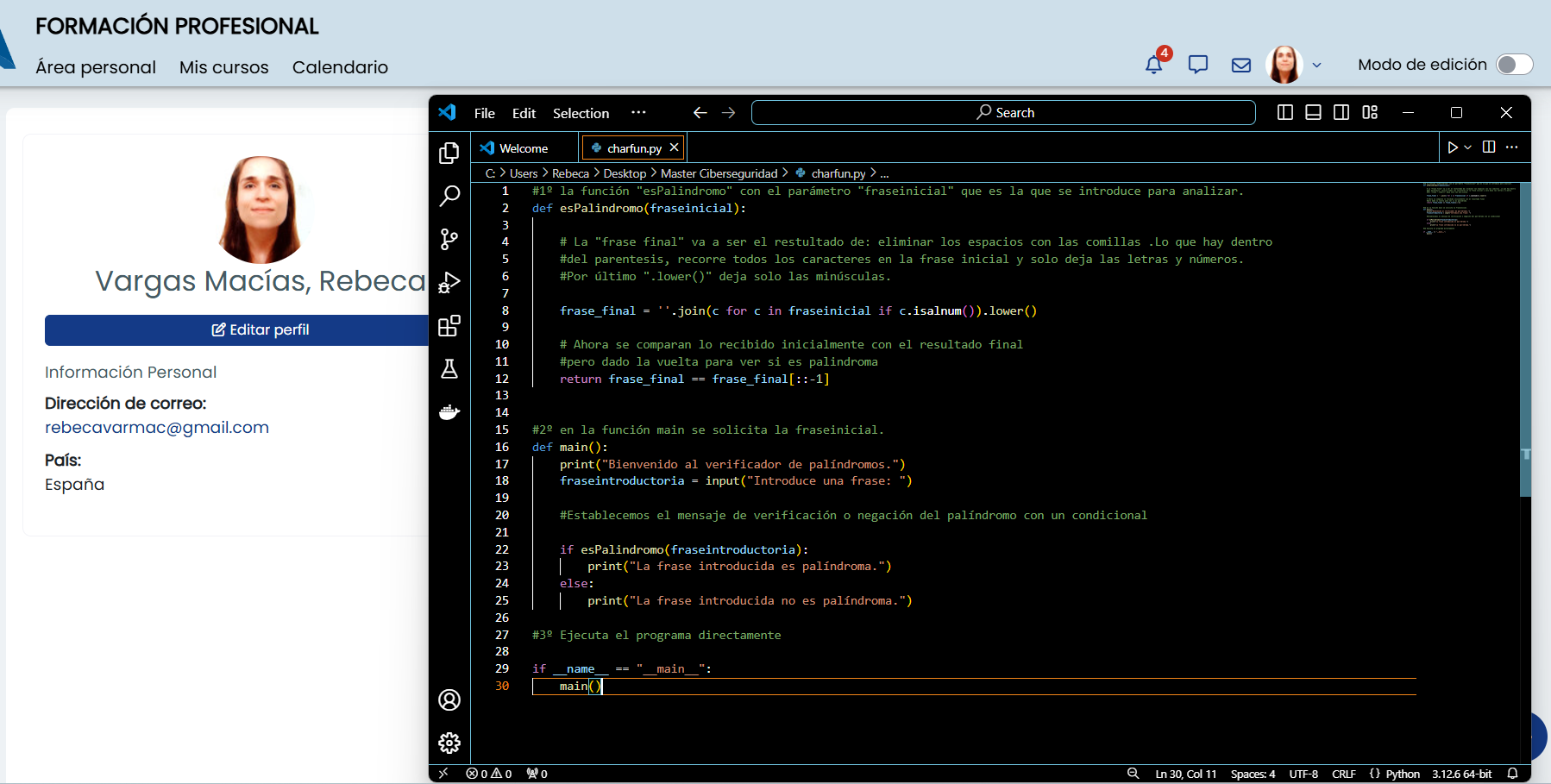
1º) Creación de la función esPalindromo, donde recogeremos lo expresado anteriormente.

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.



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

## **PROGRAMA DE TESTING**

Importamos la librería unittest y creamos la clase Test de tipo unittest.TextCase 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.

Texto

Descripción generada automáticamente

## **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.

*Texto

Descripción generada automáticamente*

La primera es palíndroma.

Texto

Descripción generada automáticamente

La segunda no lo es.

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

Captura de pantalla de computadora

Descripción generada automáticamente