



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

ACG - Auditoría Calidad y Gestión de Sistemas  
2024/2025

Francesco Lorenzoni  
PCA25403GU

## **Practica 4**

*Pytest*



# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Trabajo</b>  | <b>5</b> |
| 1.1      | Ejercicios 1/2/3 - <code>sin_vocales</code> . . . . . | 5        |



# Chapter 1

# Trabajo

### 1.1 Ejercicios 1/2/3 - sin\_vocales

El primero error en la función dada `sin_vocales(s)` es que no se tiene cuenta de las mayúsculas y minúsculas. Por lo tanto, la función no elimina las vocales mayúsculas. Para solucionarlo, es suficiente añadir a la lista de vocales la versión mayúscula de cada vocal.

```
def test_sin_vocales():
    s = "el agua esta mojada"
    exp = "l g st mjd"
    assert sin_vocales(s) == exp

    ...

    s = "El AgUe eStA mOjAdA"
    exp = "l g St mjd"
    assert sin_vocales(s) == exp
```

El segundo error es que la función no tiene cuenta de los acentos. Por lo tanto, la función no elimina las vocales acentuadas. Para solucionarlo, es suficiente añadir a la lista de vocales la versión acentuada de cada vocal.

```
vocales = 'aeiouAEIOUáéíóúàèìòùäëïöüâê  
îôûÁÉÍÓÚÀÈÌÒÙÄËÏÖÜÂÊÎÔÛ
```

Esta solución no parece muy elegante, ya que la lista de vocales se vuelve muy larga. Buscando sobre el internet he visto que una solución más elegante sería usar una expresión regular para eliminar todas las vocales, acentuadas o no. Para ello, se puede usar el módulo `re` de Python, junto con `unicodedata`, como se muestra en el código ??.

```
import unicodedata
import re
```

```
def sin_vocales(s):  
    # Normaliza el texto separando los caracteres básicos  
    s_norm = unicodedata.normalize('NFD', s)  
    # Elimina todas las vocales base y los diacríticos  
    s_sin_vocales = re.sub(r'[aeiouAEIOU\u0300-\u036f]',  
        return s_sin_vocales
```