



Instituto Tecnológico y de Estudios Superiores de Monterrey

## ***6.2 Ejercicio de programación 3 y pruebas de unidad***

**Daniel García Soni**

**A01796163**

Pruebas de software y aseguramiento de la calidad

Fecha: 16/02/2025

### Contenido

2.18 Explicar los fundamentos del desarrollo de pruebas unitarias .....	2
Características principales:.....	2
Buenas prácticas para escribir pruebas unitarias .....	2
First run flake8 and pylint: .....	3
Last flake8 and pylint run with no warnings: .....	3
Coverage :.....	3
Referencias.....	4

## 2.18 Explicar los fundamentos del desarrollo de pruebas unitarias

Las pruebas unitarias son una parte fundamental del desarrollo de software, ya que permiten verificar que cada módulo o función del código funcione correctamente de manera aislada. Su objetivo principal es asegurar que los componentes individuales de un programa cumplan con las expectativas definidas antes de integrarse con otros módulos.

Las pruebas unitarias son pequeños fragmentos de código diseñados para comprobar el correcto funcionamiento de funciones, métodos o clases dentro de un programa. Estas pruebas se ejecutan automáticamente y comparan los resultados obtenidos con los esperados.

### Características principales:

- **Independencia:** Cada prueba debe ejecutarse sin depender de otras pruebas o del entorno.
- **Automatización:** Se ejecutan automáticamente sin intervención del usuario.
- **Rapidez:** Permiten detectar errores rápidamente en la fase de desarrollo.
- **Reproducibilidad:** Los resultados deben ser consistentes cada vez que se ejecutan bajo las mismas condiciones.

### Buenas prácticas para escribir pruebas unitarias

- **Escribir pruebas simples y claras:** Cada prueba debe centrarse en una sola funcionalidad.
- **Usar nombres descriptivos:** Los nombres de las pruebas deben indicar claramente qué están verificando.
- **Cubrir casos normales y extremos:** Se deben probar valores típicos, valores límite y casos inesperados.
- **Mantener las pruebas independientes:** Cada prueba debe ejecutarse sin depender de otras.
- **Ejecutar las pruebas con frecuencia:** Se deben integrar en el flujo de desarrollo para detectar errores lo antes posible.

## First run flake8 and pylint:

```
C:\Users\Compu>cd C:\Users\Compu\Documents\reservationsystem

C:\Users\Compu\Documents\reservationsystem>flake8 main.py
main.py:62:80: E501 line too long (84 > 79 characters)
main.py:86:80: E501 line too long (83 > 79 characters)
main.py:178:80: E501 line too long (89 > 79 characters)
main.py:181:80: E501 line too long (92 > 79 characters)

C:\Users\Compu\Documents\reservationsystem>pylint main.py
***** Module main
main.py:1:0: C0114: Missing module docstring (missing-module-docstring)
main.py:7:0: C0116: Missing function or method docstring (missing-function-docstring)
main.py:10:9: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
main.py:17:0: C0116: Missing function or method docstring (missing-function-docstring)
main.py:18:9: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
main.py:23:0: C0115: Missing class docstring (missing-class-docstring)
main.py:30:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:36:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:42:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:47:0: C0115: Missing class docstring (missing-class-docstring)
main.py:54:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:60:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:66:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:71:0: C0115: Missing class docstring (missing-class-docstring)
main.py:78:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:84:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:90:4: C0116: Missing function or method docstring (missing-function-docstring)
main.py:95:0: C0115: Missing class docstring (missing-class-docstring)

-----
Your code has been rated at 8.71/10

C:\Users\Compu\Documents\reservationsystem>
```

## Last flake8 and pylint run with no warnings:

```
C:\Users\Compu\Documents\reservationsystem>flake8 main.py
main.py:110:80: E501 line too long (83 > 79 characters)

C:\Users\Compu\Documents\reservationsystem>flake8 main.py

C:\Users\Compu\Documents\reservationsystem>flake8 main.py

C:\Users\Compu\Documents\reservationsystem>pylint main.py

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

C:\Users\Compu\Documents\reservationsystem>
```

## Coverage :

```
-----
Ran 0 tests in 0.000s

OK
C:\Python311\Lib\site-packages\coverage\control.py:892: CoverageWarning: No data was collected. (no-data-collected)
  self._warn("No data was collected.", slug="no-data-collected")

C:\Users\Compu\Documents\reservationsystem>

C:\Users\Compu\Documents\reservationsystem>python -m coverage run main.py
.....
Ran 5 tests in 0.140s

OK

C:\Users\Compu\Documents\reservationsystem>

C:\Users\Compu\Documents\reservationsystem>python -m coverage report -m
Name      Stmts  Miss  Cover   Missing
-----
main.py    146      3    98%    9, 13-14
TOTAL      146      3    98%
```

## Referencias

- Luminousmen. (2019). *Python static analysis tools*. Recuperado el 16 de febrero de 2025, de <https://luminousmen.com/post/python-static-analysis-tools>
- PyCQA. (2023). *Flake8: Your Tool for Style Guide Enforcement*. Recuperado el 16 de febrero de 2025, de <https://flake8.pycqa.org/en/latest/>
- Python Software Foundation. (2023). *unittest — Unit testing framework*. Recuperado el 16 de febrero de 2025, de <https://docs.python.org/3/library/unittest.html>
- Pynative. (2021). *Python JSON programming*. Recuperado el 16 de febrero de 2025, de <https://pynative.com/python/json/>
- van Rossum, G., Warsaw, B., & Coghlan, N. (2023). *PEP 8 – Style Guide for Python Code*. Python Software Foundation. Recuperado el 16 de febrero de 2025, de <https://peps.python.org/pep-0008/>
- Python Software Foundation. (2023). *The Python tutorial*. Recuperado el 16 de febrero de 2025, de <https://docs.python.org/3/tutorial/index.html>