SESIÓN 27 · MÓDULO 2

Probar y depurar

Leer un traceback de abajo arriba, escribir pruebas unitarias y buscar el porqué de un error con método.

Antonio Chanes Espigares · CC BY-NC-ND 4.0

Hoy

0:00	Corrección de la tarea 26.
0:15	Situación: una avería en cadena: el coche no arranca porque la batería no carga porque la correa está rota.
0:25	**Predice** y visualizador: el traceback enseña la pila de llamadas al revés.
0:45	Construcción: `assert`, tracebacks, `unittest`, depuración por hipótesis.
1:20	Ejercicios del capítulo.
1:45	Preguntas tipo examen, tarea 27 y hito 2.

Para empezar

Una avería en cadena: el coche no arranca porque la batería no carga porque la correa está rota.

Predice

Escribe en papel qué crees que pasará. Después, lo ejecutamos.

```python
def media(lista):
    return sum(lista) / len(lista)

def informe(datos):
    return f"Media: {media(datos):.1f}"

print(informe([4, 6]))
print(informe([]))
```

Predice

Escribe en papel qué crees que pasará. Después, lo ejecutamos.

```python
def media(lista):
    return sum(lista) / len(lista)

def informe(datos):
    return f"Media: {media(datos):.1f}"

print(informe([4, 6]))
print(informe([]))
```

LO QUE SALE

`Media: 5.0` y después `ZeroDivisionError`

El traceback acaba en la línea de `media`, la división; la causa está en quien la llamó con una lista vacía.

`assert`: una comprobación en una línea

Probar, depurar y trabajar en equipo · 1

```python
def media(notas):
    assert len(notas) > 0, "la lista está vacía"
    return sum(notas) / len(notas)

print(media([6, 8]))
print(media([]))
```

Pruebas unitarias con `unittest`

Probar, depurar y trabajar en equipo · 3

```
import unittest

def media(notas):
    if len(notas) == 0:
        raise ValueError("la lista está vacía")
    return sum(notas) / len(notas)

class PruebasMedia(unittest.TestCase):
    def test_caso_tipico(self):
        self.assertEqual(media([6, 8]), 7)

    def test_un_solo_valor(self):            # caso límite
        self.assertEqual(media([5]), 5)
…
```

Manos a la obra

Abre el cuaderno de la sesión en Colab, con los ejemplos de hoy y estos ejercicios:

1. Clasifica el error
2. Lee el traceback
3. Diseña los casos
4. Una prueba con unittest
5. Revisa a la IA



Cuaderno en Colab

Tarea 27 · Pruebas para la calificación

1. Toma `calificacion(nota)` de la tarea 13 y cámbiala para que **lance** `ValueError` si la nota no está entre 0 y 10, en…
2. Escribe una clase `PruebaCalificacion(unittest.TestCase)` con tres pruebas: los bordes (5, 9, 10), justo por debajo…
3. En Colab, ejecútalas con `unittest.main(argv=["ignorado"], exit=False)`.
4. Rompe a propósito la función (cambia un `<` por `<=`) y mira qué prueba falla.

COMPROBACIÓN Las tres pruebas pasan con la función correcta; con `nota <= 5` falla la de los bordes.

El proyecto



Cuaderno del proyecto

Hito 2 del proyecto. Clases, excepciones y pruebas del proyecto.

Antes de irte

Las **preguntas tipo examen** del capítulo, en directo: se corrigen al momento.



Preguntas del capítulo

LA PRÓXIMA SESIÓN

Sesión 28 · Trabajar en equipo y con IA

Programación en Python · Sesión 27 · Probar y depurar