

Test_prueba1:

```
C:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio>pytest tests/test_prueba1.py -v
===== test session starts =====
platform win32 -- Python 3.12.6, pytest-8.3.3, pluggy-1.5.0 -- C:\Users\Usuario\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio
collected 3 items

tests/test_prueba1.py::test_suma_params[0-0-0] PASSED [ 33%]
tests/test_prueba1.py::test_suma_params[2-1-2] PASSED [ 66%]
tests/test_prueba1.py::test_suma_params[1-2-2] PASSED [100%]

===== 3 passed in 0.02s =====

C:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio>|
```

```
C:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio>pytest tests/test_prueba1.py
===== test session starts =====
platform win32 -- Python 3.12.6, pytest-8.3.3, pluggy-1.5.0
rootdir: C:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio
collected 3 items

tests\test_prueba1.py FFF [100%]

===== FAILURES =====
----- test_suma_params[0-0-0] -----
input_x = 0, input_y = 0, expected = 0

    @pytest.mark.parametrize(
        "input_x, input_y, expected",
        [
            (0, 0, 0),
            (2, 1, 2),
            (1, 2, 2)
        ]
    )

    def test_suma_params(input_x, input_y, expected):
        assert comparacion_num(input_x, input_y) == expected
tests\test_prueba1.py:14:
-----
num1 = '0', num2 = 0

    def comparacion_num(num1, num2):
        num1 = str(num1)
        if num1 < num2:
E           TypeError: '<' not supported between instances of 'str' and 'int'

src\prueba1.py:4: TypeError
----- test_suma_params[2-1-2] -----
input_x = 2, input_y = 1, expected = 2
```

```
src\prueba1.py:4: TypeError
----- test_suma_params[2-1-2] -----
input_x = 2, input_y = 1, expected = 2

    @pytest.mark.parametrize(
        "input_x, input_y, expected",
        [
            (0, 0, 0),
            (2, 1, 2),
            (1, 2, 2)
        ]
    )

    def test_suma_params(input_x, input_y, expected):
        assert comparacion_num(input_x, input_y) == expected
tests\test_prueba1.py:14:
-----
num1 = '2', num2 = 1

    def comparacion_num(num1, num2):
        num1 = str(num1)
        if num1 < num2:
E           TypeError: '<' not supported between instances of 'str' and 'int'

src\prueba1.py:4: TypeError
----- test_suma_params[1-2-2] -----
input_x = 1, input_y = 2, expected = 2

    @pytest.mark.parametrize(
        "input_x, input_y, expected",
        [
            (0, 0, 0),
            (2, 1, 2),
            (1, 2, 2)
        ]
    )
```

```

----- test_suma_params[1-2-2] -----
input_x = 1, input_y = 2, expected = 2

@pytest.mark.parametrize(
    "input_x, input_y, expected",
    [
        (0, 0, 0),
        (2, 1, 2),
        (1, 2, 2)
    ]
)

def test_suma_params(input_x, input_y, expected):
    assert comparacion_num(input_x, input_y) == expected

tests\test_prueba1.py:14:
-----
num1 = '1', num2 = 2

def comparacion_num(num1, num2):
    num1 = str(num1)
    if num1 < num2:
>
E       TypeError: '<' not supported between instances of 'str' and 'int'

src\prueba1.py:4: TypeError
===== short test summary info =====
FAILED tests/test_prueba1.py::test_suma_params[0-0-0] - TypeError: '<' not supported between instances of 'str' and 'int'
FAILED tests/test_prueba1.py::test_suma_params[2-1-2] - TypeError: '<' not supported between instances of 'str' and 'int'
FAILED tests/test_prueba1.py::test_suma_params[1-2-2] - TypeError: '<' not supported between instances of 'str' and 'int'
===== 3 failed in 0.07s =====

C:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio>

```

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the test results and output panes at the bottom. The test results pane shows that all three test cases passed. The output pane shows the test session starting and finishing successfully.

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the test results and output panes at the bottom. The test results pane shows that all three test cases failed. The output pane shows the test session starting and finishing with 3 failures. The error messages in the output pane are:

[0-0-0] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio\src\prueba1.py:4: TypeError: '<' not supported between instances of 'str' and 'int'

[2-1-2] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio\src\prueba1.py:4: TypeError: '<' not supported between instances of 'str' and 'int'

[1-2-2] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio\src\prueba1.py:4: TypeError: '<' not supported between instances of 'str' and 'int'

Test_ej01

The screenshot shows the VS Code interface with the file explorer on the left displaying a project structure with files like `test_ej01.py`, `test_ej02.py`, `test_ej04.py`, and `test_prueba1.py`. The main editor shows the code for `test_ej01.py`, which imports `pytest` and `saludo` from `src.ej01_def`, and defines a test function `test_saludo()` that asserts `saludo('Alejandro') == "Hola, Alejandro."`. The bottom panel shows the `TEST RESULTS` tab with a message indicating that the test passed successfully. The `OUTPUT` tab shows the command used to run the tests: `python -m pytest -p 'vscode.pytest' --rootdir=c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej01.py::test_saludo`.

```
tests > test_ej01.py >
1 import pytest
2 from src.ej01_def import saludo
3
4 def test_saludo():
5     assert saludo('Alejandro') == "Hola, Alejandro."
```

Running pytest with args: ['-p', 'vscode.pytest', '--rootdir=c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej01.py::test_saludo']
===== test session starts =====
platform win32 -- Python 3.12.6, pytest-8.3.3, pluggy-1.5.0
rootdir: c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio
collected 1 item

tests\test_ej01.py . [100%]

===== 1 passed in 0.03s =====
Finished running tests!

The screenshot shows the VS Code interface with the file explorer on the left. The main editor shows the code for `test_ej01.py`. The bottom panel shows the `TEST RESULTS` tab with a message indicating that the test failed. The `OUTPUT` tab shows the command used to run the tests: `python -m pytest -p 'vscode.pytest' --rootdir=c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej01.py::test_saludo`. The error message is: `TypeError: saludo() takes 0 positional arguments but 1 was given`.

```
tests > test_ej01.py > test_saludo
1 import pytest
2 from src.ej01_def import saludo
3
4 def test_saludo():
5     assert saludo('Alejandro') == "Hola, Alejandro."
```

Running pytest with args: ['-p', 'vscode.pytest', '--rootdir=c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej01.py::test_saludo']
===== test session starts =====
platform win32 -- Python 3.12.6, pytest-8.3.3, pluggy-1.5.0
rootdir: c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio
collected 1 item

tests\test_ej01.py F [100%]

===== 1 failed, 0 passed in 0.03s =====
Finished running tests!

Test_ej02

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the test results panel at the bottom. The file explorer shows a project structure with a 'tests' directory containing 'test_ej01.py', 'test_ej02.py', and 'test_salario'. The editor shows the code for 'test_ej02.py', which imports 'pytest' and 'salario' from 'src.ej02_def', and defines a test function 'test_salario' that uses 'pytest.mark.parametrize' to test the 'salario' function with three sets of input values: (2, 3, 6), (2, 1, 2), and (2, 2, 4). The test results panel at the bottom shows the output of the test run, indicating that all three tests passed successfully. The output includes the test session start, the platform (win32), Python version (3.12.6), and the test results for the three parametrized test cases.

```
tests > test_ej02.py > test_salario
1 import pytest
2 from src.ej02_def import salario
3
4 @pytest.mark.parametrize(
5     "input_x, input_y, expected",
6     [
7         (2, 3, 6),
8         (2, 1, 2),
9         (2, 2, 4)
10    ]
11 )
12 def test_salario(input_x, input_y, expected):
13     assert salario(input_x, input_y) == expected
```

Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio

test session starts
platform win32 -- Python 3.12.6, pytest-8.3.3, pluggy-1.5.0
rootdir: c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio
collected 3 items

tests\test_ej02.py ... [100%]

===== 3 passed in 0.03s =====
Finished running tests!

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the test results panel at the bottom. The file explorer shows a project structure with a 'tests' directory containing 'test_ej01.py', 'test_ej02.py', and 'test_salario'. The editor shows the code for 'test_ej02.py', which imports 'pytest' and 'salario' from 'src.ej02_def', and defines a test function 'test_salario' that uses 'pytest.mark.parametrize' to test the 'salario' function with three sets of input values: (2, 3, 6), (2, 1, 2), and (2, 2, 4). The test results panel at the bottom shows the output of the test run, indicating that all three tests failed. The output includes the test session start, the platform (win32), Python version (3.12.6), and the test results for the three parametrized test cases, each failing with an 'AssertionError'.

```
src > ej02_def.py > salario
1 #Escribe un programa para pedirle al usuario las horas de trabajo y el precio por hora y calcule el importe total del servicio
2
3 def salario(horas, coste_horas):
4     horas = str(horas)
5     return horas * coste_horas
6
7 def main():
8     horas = int(input("Escriba las horas de trabajo: "))
9     coste_horas = float(input("Escriba el precio por hora: "))
10    print(f"El importe es: {salario(horas, coste_horas)}€.")
11
12 if __name__ == "__main__":
13     main()
```

Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio

test session starts
platform win32 -- Python 3.12.6, pytest-8.3.3, pluggy-1.5.0
rootdir: c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-Lmrocio
collected 3 items

tests\test_ej02.py:13: AssertionError
===== short test summary info =====
FAILED tests\test_ej02.py::test_salario[2-3-6] - AssertionError: assert '222' ...
FAILED tests\test_ej02.py::test_salario[2-1-2] - AssertionError: assert '2' == 2
FAILED tests\test_ej02.py::test_salario[2-2-4] - AssertionError: assert '22' ...
===== 3 failed in 0.08s =====
Finished running tests!

Test_ej04

The screenshot shows the VS Code interface with the file `test_ej04.py` open. The code defines a parametrized test `test_grados_celsius` with three test cases: `(105, 40.56)`, `(100, 37.78)`, and `(32, 0.0)`. The test function asserts that `grados_celsius(input_x) == expected`. The Test Results panel shows that all three tests passed. The Output panel displays the pytest execution details, including the platform (win32), Python version (3.12.6), and the number of collected items (3).

```
tests > test_ej04.py > test_grados_celsius
1 import pytest
2 from src.ej04_def2 import grados_celsius
3
4 @pytest.mark.parametrize(
5     "input_x, expected",
6     [
7         (105, 40.56),
8         (100, 37.78),
9         (32, 0.0)
10    ]
11 )
12 def test_grados_celsius(input_x, expected):
13     assert grados_celsius(input_x) == expected
```

Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej04.py

- ✓ [105-40.56]
- ✓ [100-37.78]
- ✓ [32-0.0]

> 39 older results

tests\test_ej04.py ... [100%]

===== 3 passed in 0.03s =====

Finished running tests!

The screenshot shows the VS Code interface with the file `test_ej04.py` open. The code is the same as in the previous screenshot. However, the Test Results panel shows that all three tests failed. The Output panel displays the pytest execution details, including the platform (win32), Python version (3.12.6), and the number of collected items (3). The error message indicates that the `assert` statement failed because the actual value was `None` instead of the expected value.

```
tests > test_ej04.py > test_grados_celsius
1 import pytest
2 from src.ej04_def2 import grados_celsius
3
4 @pytest.mark.parametrize(
5     "input_x, expected",
6     [
7         (105, 40.56),
8         (100, 37.78),
9         (32, 0.0)
10    ]
11 )
12 def test_grados_celsius(input_x, expected):
13     assert grados_celsius(input_x) == expected
```

Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej04.py

- ✗ [105-40.56] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej04.py:13: AssertionError
- ✗ [100-37.78] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej04.py:13: AssertionError
- ✗ [32-0.0] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmrocio\tests\test_ej04.py:13: AssertionError

> 56 older results

tests\test_ej04.py:13: AssertionError

===== short test summary info =====

FAILED tests\test_ej04.py:13: AssertionError - assert None == 40.56

FAILED tests\test_ej04.py:13: AssertionError - assert None == 37.78

FAILED tests\test_ej04.py:13: AssertionError - assert None == 0.0

===== 3 failed in 0.08s =====

Finished running tests!

Test_ej05

The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a test results panel on the right. The file explorer shows a project structure with a 'tests' directory containing 'test_ej05.py'. The code editor shows the following code:

```
1 import pytest
2 from src.ej05_def2 import calcula_precio
3
4 @pytest.mark.parametrize(
5     "input_x, input_y, expected",
6     [
7         (100, -1, (1.21, 121)),
8         (100, 15, (1.15, 115)),
9     ]
10 )
11 def test_grados_celsius(input_x, input_y, expected):
12     assert calcula_precio(input_x, input_y) == expected
```

The test results panel shows the following output:

```
Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmroci...
[100--1-expected0]
[100-15-expected1]
> 58 older results
```

The test results panel also shows the following summary:

```
tests\test_ej05.py .. [100%]
===== 2 passed in 0.03s =====
Finished running tests!
```

The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a test results panel on the right. The file explorer shows a project structure with a 'tests' directory containing 'test_ej05.py'. The code editor shows the following code:

```
1 import pytest
2 from src.ej05_def2 import calcula_precio
3
4 @pytest.mark.parametrize(
5     "input_x, input_y, expected",
6     [
7         (100, -1, (1.21, 121)),
8         (100, 15, (1.15, 115)),
9     ]
10 )
11 def test_grados_celsius(input_x, input_y, expected):
12     assert calcula_precio(input_x, input_y) == expected
```

The test results panel shows the following output:

```
Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmroci...
[100--1-expected0] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmroci...
[100-15-expected1] c:\Users\Usuario\Documents\ReposGit\dawb1-2425-ejercicios-u1-lmroci...
> 59 older results
```

The test results panel also shows the following summary:

```
tests\test_ej05.py:12: AssertionError
===== short test summary info =====
FAILED tests\test_ej05.py::test_grados_celsius[100--1-expected0] - assert Non...
FAILED tests\test_ej05.py::test_grados_celsius[100-15-expected1] - assert Non...
===== 2 failed in 0.09s =====
Finished running tests!
```

Test_ej11

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the test results panel on the right. The file explorer shows a project structure with a 'tests' folder containing several test files. The editor shows the code for 'test_ej11.py', which imports 'pytest' and 'suma_final' from 'src.ej11_def'. It uses '@pytest.mark.parametrize' to define test cases for 'test_grados_celsius' with inputs (10, 45) and (7, 21). The test function asserts that 'suma_final(input_x)' equals 'expected'. The test results panel shows that the tests passed successfully, with a summary of 2 passed tests in 0.02s.

```
1 import pytest
2 from src.ej11_def import suma_final
3
4 @pytest.mark.parametrize(
5     "input_x, expected",
6     [
7         (10, 45),
8         (7, 21),
9     ]
10 )
11 def test_grados_celsius(input_x, expected):
12     assert suma_final(input_x) == expected
```

Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-...

2 passed in 0.02s

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the test results panel on the right. The file explorer shows the same project structure as the first screenshot. The editor shows the code for 'test_ej11.py', which is the same as in the first screenshot. The test results panel shows that the tests failed, with a summary of 2 failed tests in 0.08s. The error messages indicate that the assertions failed for input_x = 10 and input_x = 7.

```
1 import pytest
2 from src.ej11_def import suma_final
3
4 @pytest.mark.parametrize(
5     "input_x, expected",
6     [
7         (10, 45),
8         (7, 21),
9     ]
10 )
11 def test_grados_celsius(input_x, expected):
12     assert suma_final(input_x) == expected
13     assert None == 45
14     + where None = suma_final(10)
```

Running Tests for Workspace(s): c:\Users\Usuario\Documents\ReposGit\dawb1-2425-...

2 failed in 0.08s