

Test ej01.

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a directory named 'dawb1-2425-ejercicios-u1-indadominguez'. The main editor displays the file 'test_ej01.py' with the following code:

```
1 import pytest
2 from src.ej01_def import dar_bienvenida
3
4 def test_dar_bienvenida():
5     assert dar_bienvenida("Indalecio") == ("Hola, Indalecio!")
6
```

The bottom panel shows the 'TEST RESULTS' tab with the following output:

```
st_ej01.py::test_dar_bienvenida]
===== test session starts =====
platform win32 -- Python 3.12.7, pytest-8.3.3, pluggy-1.5.0
rootdir: c:\Users\indal\OneDrive\Escritorio\TRABAJOS ALBERTI\Pruebas Programación\dawb1-2425-ejercicios-
collected 1 item

tests\test_ej01.py . [100%]

===== 1 passed in 0.02s =====
Finished running tests!
```

The right sidebar shows 'Running Tests for Workspace(s): c:\Users\indal\OneDrive\E...' with a list of tests including 'test_dar_bienvenida' and a total of 39 older results.

The screenshot shows the Visual Studio Code interface with the same file explorer. The main editor displays the file 'test_ej01.py' with the following code:

```
1 import pytest
2 from src.ej01_def import dar_bienvenida
3
4 def test_dar_bienvenida():
5     assert dar_bienvenida("Indalecio") == ("Hola, Indalecio!")
6
```

The bottom panel shows the 'TEST RESULTS' tab with the following output:

```
tests\test_ej01.py:5: AssertionError
===== short test summary info =====
FAILED tests\test_ej01.py::test_dar_bienvenida - AssertionError: assert 'Hola...
===== 1 failed in 0.08s =====
Finished running tests!
```

The right sidebar shows 'Running Tests for Workspace(s): c:\Users\indal\OneDrive\E...' with a list of tests including 'test_dar_bienvenida' and a total of 51 older results.

Test ej02.

The screenshot shows the Visual Studio Code interface with a Python project. The file explorer on the left shows a directory structure with files like `test_ej01.py`, `test_ej02.py`, and `test_ej03.py`. The main editor displays the code for `test_ej02.py`, which uses `pytest.mark.parametrize` to define three test cases for the `test_calcular_importe_total` function. The test cases are: `(4, 2, 8)`, `(1, 3, 3)`, and `(2, 2, 4)`. The test function asserts that the `calcular_importe_total` function returns the expected value. The terminal at the bottom shows the command `python -m pytest tests/test_ej02.py` being executed, and the output indicates that all three tests passed successfully. The status bar at the bottom shows the current file is `test_ej02.py` at line 1, column 1.

```
1 import pytest
2 from src.ej02_def import calcular_importe_total
3
4 @pytest.mark.parametrize(
5     "input_x, input_y, expected",
6     [
7         (4, 2, 8),
8         (1, 3, 3),
9         (2, 2, 4)
10    ]
11 )
12
13 def test_calcular_importe_total(input_x, input_y, expected):
14     assert calcular_importe_total(input_x, input_y) == expected
15
16
17
18
```

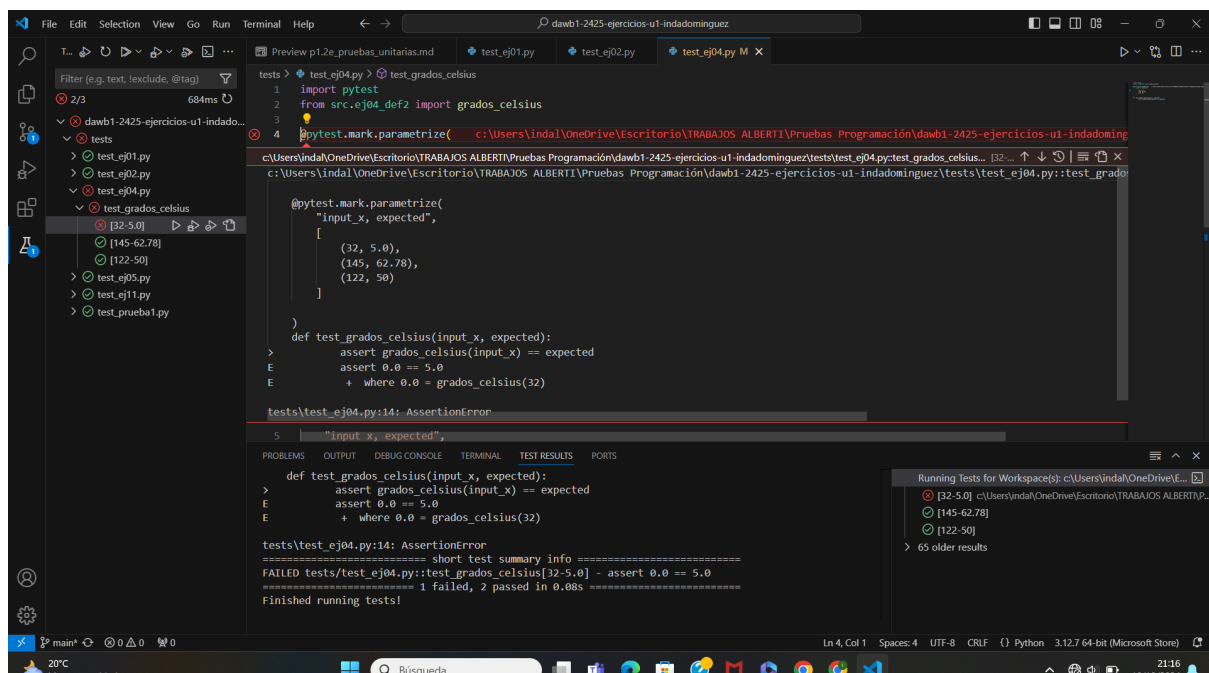
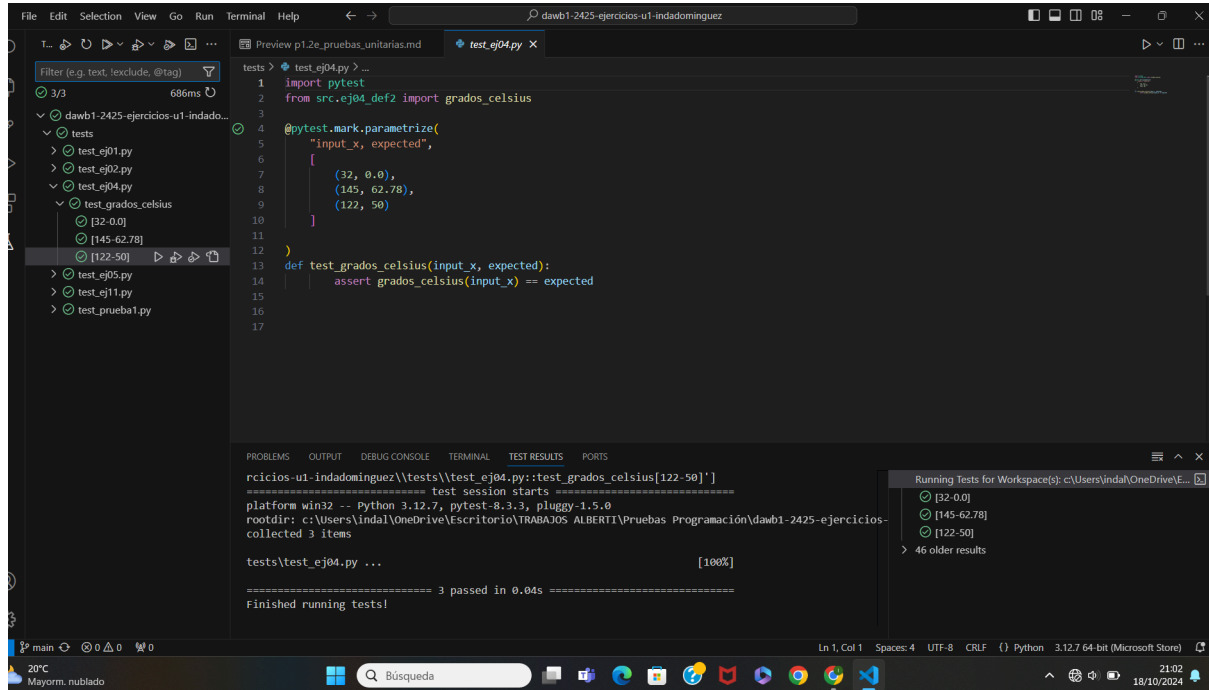
Running Tests for Workspace(s): c:\Users\indal\OneDrive\...
[4-2-8]
[1-3-3]
[2-2-4]
> 43 older results

The screenshot shows the Visual Studio Code interface with the same project as the first screenshot. The file explorer on the left shows the same directory structure. The main editor displays the code for `test_ej02.py`, which is identical to the first screenshot. The terminal at the bottom shows the command `python -m pytest tests/test_ej02.py` being executed, and the output indicates that the test failed. The error message is `AssertionError: 1 failed in 0.03s`. The status bar at the bottom shows the current file is `test_ej02.py` at line 7, column 17.

```
1 import pytest
2 from src.ej02_def import calcular_importe_total
3
4 @pytest.mark.parametrize(
5     "input_x, input_y, expected",
6     [
7         (4, 2, 7),
8         (1, 3, 3),
9         (2, 2, 4)
10    ]
11 )
12
13 def test_calcular_importe_total(input_x, input_y, expected):
14     assert calcular_importe_total(input_x, input_y) == expected
15
16
17
18
```

Running Tests for Workspace(s): c:\Users\indal\OneDrive\...
[2-2-4]
> 61 older results

Test ej04.



Test ej05.

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the terminal and test results panels at the bottom. The file explorer shows a project named 'dawb1-2425-ejercicios-u1-indadominguez' with a 'tests' folder containing several test files. The editor shows the code for 'test_ej05.py', which uses 'pytest' and 'parametrize' to test the 'calcula_precio' function. The terminal shows the command 'python -m pytest tests/test_ej05.py' and the output of the test run, which indicates that all tests passed.

```
1 import pytest
2 from src.ej05_def2 import calcula_precio
3
4 @pytest.mark.parametrize(
5     "input_x, input_y, expected",
6     [
7         (14, 25, "El precio final del articulo con IVA (25.00) es 17.50€."),
8         (10, 25, "El precio final del articulo con IVA (25.00) es 12.50€."),
9         (27, 10, "El precio final del articulo con IVA (10.00) es 29.70€."),
10    ]
11 )
12
13 def test_calcula_precio(input_x, input_y, expected):
14     assert calcula_precio(input_x, input_y) == expected
15
16
```

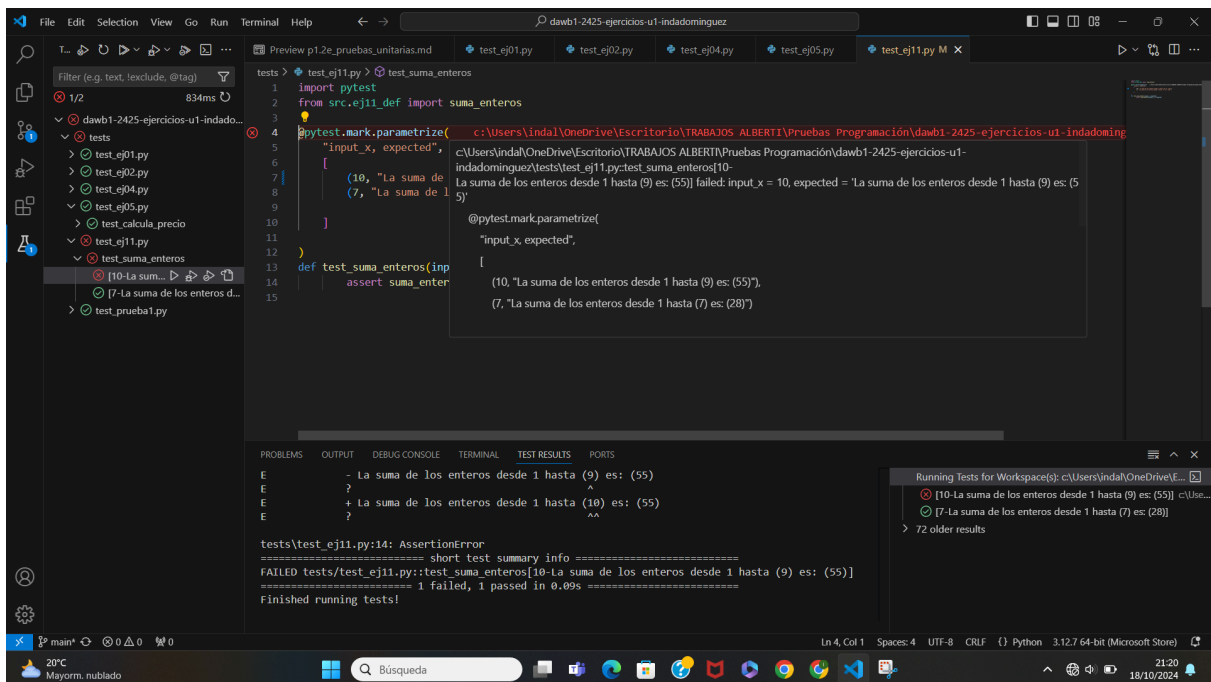
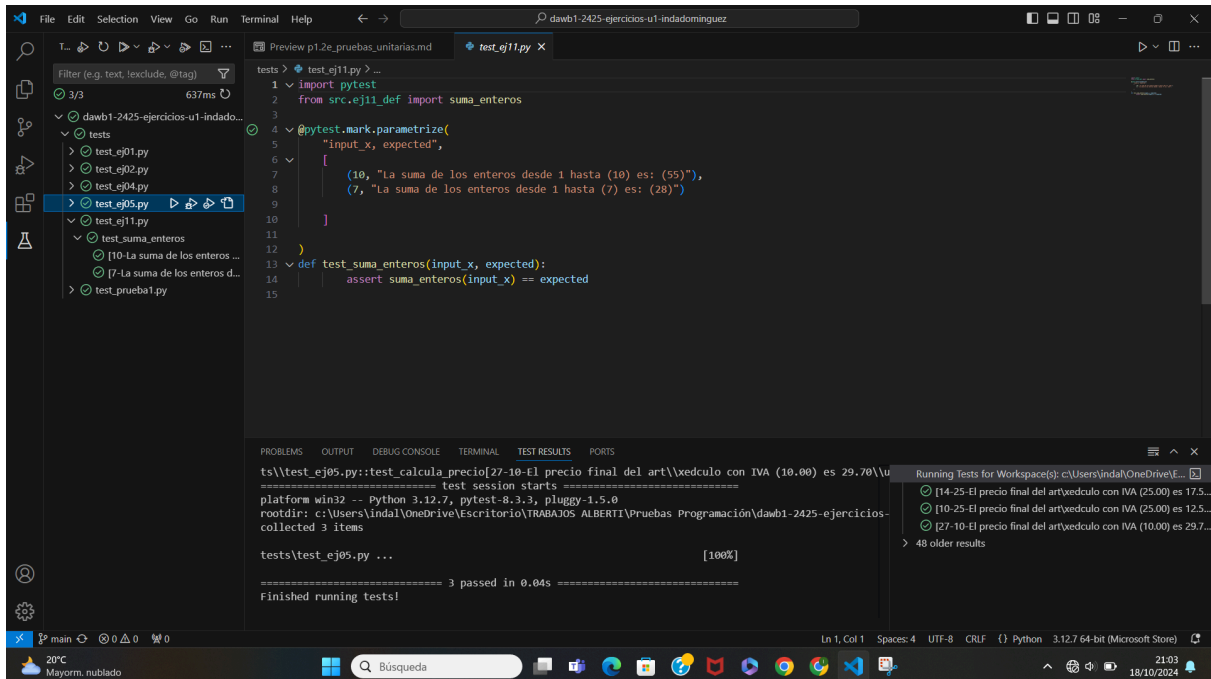
Running Tests for Workspace(s): c:\Users\indal\OneDrive\...
[14-25-El precio final del art\vedculo con IVA (25.00) es 17.5...
[10-25-El precio final del art\vedculo con IVA (25.00) es 12.5...
[27-10-El precio final del art\vedculo con IVA (10.00) es 29.7...
> 47 older results

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the terminal and test results panels at the bottom. The file explorer shows the same project as the first screenshot. The editor shows the code for 'test_ej05.py', which is the same as in the first screenshot. The terminal shows the command 'python -m pytest tests/test_ej05.py' and the output of the test run, which indicates that one test failed. The test results panel shows the details of the failed test, including the input values and the expected output.

```
1 import pytest
2 from src.ej05_def2 import calcula_precio
3
4 @pytest.mark.parametrize(
5     "input_x, input_y, expected",
6     [
7         (14, 25, "El precio final del articulo con IVA (22.00) es 17.50€."),
8         (10, 25, "El precio final del articulo con IVA (25.00) es 12.50€."),
9         (27, 10, "El precio final del articulo con IVA (10.00) es 29.70€."),
10    ]
11 )
12
13 def test_calcula_precio(input_x, input_y, expected):
14     assert calcula_precio(input_x, input_y) == expected
15
16
```

Running Tests for Workspace(s): c:\Users\indal\OneDrive\...
[14-25-El precio final del art\vedculo con IVA (22.00) es 17.5...
> 69 older results

Test ej11.



Test prueba1.

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure. The main editor shows the code for `test_prueba1.py`, which uses `pytest.mark.parametrize` to define three test cases for `test_num_mayor`. The bottom panel shows the 'TEST RESULTS' tab with a summary of the test run.

```
1 import pytest
2 from src.prueba1 import num_mayor
3
4
5 @pytest.mark.parametrize(
6     "input_x, input_y, expected",
7     [
8         (0, 0, 0),
9         (2, 1, 2),
10        (1, 2, 2)
11    ]
12 )
13
14 def test_num_mayor(input_x, input_y, expected):
15     assert num_mayor(input_x, input_y) == expected
```

TEST RESULTS

```
1-indadominguez\\tests\\test_prueba1.py::test_num_mayor[1-2-2]
===== test session starts =====
platform win32 -- Python 3.12.7, pytest-8.3.3, pluggy-1.5.0
rootdir: c:\Users\indal\OneDrive\Escritorio\TRABAJOS ALBERTI\Pruebas Programaci3n\dawb1-2425-ejercicios-
collected 3 items

tests\test_prueba1.py ... [100%]

===== 3 passed in 0.02s =====
Finished running tests!
```

Running Tests for Workspace(s): c:\Users\indal\OneDrive\...

- ✓ [0-0-0]
- ✓ [2-1-2]
- ✓ [1-2-2]

> 50 older results

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure. The main editor shows the code for `test_prueba1.py`, which uses `pytest.mark.parametrize` to define three test cases for `test_num_mayor`. The bottom panel shows the 'TEST RESULTS' tab with a summary of the test run.

```
1 import pytest
2 from src.prueba1 import num_mayor
3
4
5 @pytest.mark.parametrize(
6     "input_x, input_y, expected",
7     [
8         (1, 0, 0),
9         (2, 1, 2),
10        (1, 2, 2)
11    ]
12 )
13
14 def test_num_mayor(input_x, input_y, expected):
15     assert num_mayor(input_x, input_y) == expected
```

TEST RESULTS

```
> def test_num_mayor(input_x, input_y, expected):
>     assert num_mayor(input_x, input_y) == expected
E
+ where 1 = num_mayor(1, 0)

tests\test_prueba1.py:15: AssertionError
===== short test summary info =====
FAILED tests\test_prueba1.py::test_num_mayor[1-0-0] - assert 1 == 0
===== 1 failed, 2 passed in 0.11s =====
Finished running tests!
```

Running Tests for Workspace(s): c:\Users\indal\OneDrive\...

- ✗ [1-0-0]
- ✓ [2-1-2]
- ✓ [1-2-2]

> 76 older results