

En base al informe de cobertura se puede concluir que la cobertura en main.py fue del 92% y en test_main.py fue del 100%, dando como resultado total que la cobertura global es del 96%

Coverage report: 96%

Files Functions Classes

coverage.py v7.5.3, created at 2024-06-17 15:28 -0500

File ▲	statements	missing	excluded	coverage
main.py	12	1	0	92%
test_main.py	15	0	0	100%
Total	27	1	0	96%

Coverage for main.py: 92%

12 statements 11 run 1 missing 0 excluded

« prev ^ index » next coverage.py v7.5.3, created at 2024-06-17 15:28 -0500

```
1 def suma(a, b):
2     return a + b
3
4 def resta(a, b):
5     return a - b
6
7 def multiplicacion(a, b):
8     return a * b
9
10 def division(a, b):
11     if b != 0:
12         return a / b
13     else:
14         raise ValueError("No se puede dividir entre cero")
15
16 # Ejemplo de uso
17 resultado_suma = suma(5, 3)
18 print(f"Suma: {resultado_suma}")
```

Coverage for **test_main.py**: 100%

15 statements

15 run

0 missing

0 excluded

[« prev](#) [^ index](#) [» next](#) coverage.py v7.5.3, created at 2024-06-17 15:28 -0500

```
1 import unittest
2 from main import suma, resta, multiplicacion, division
3
4 class TestOperacionesBasicas(unittest.TestCase):
5     def test_suma(self):
6         self.assertEqual(suma(5, -3), 2)
7         self.assertEqual(suma(25, 2), 27)
8
9     def test_resta(self):
10        self.assertEqual(resta(5, -3), 8)
11        self.assertEqual(resta(9, 8), 1)
12
13    def test_multiplicacion(self):
14        self.assertEqual(multiplicacion(5, -3), -15)
15        self.assertEqual(multiplicacion(-2, 20), -40)
16
17    def test_division(self):
18        self.assertAlmostEqual(division(10, -2), -5)
19        self.assertEqual(division(60, 2), 30)
20
```