

doc_testes.html

Report generated on 02-Sep-2025 at 09:49:56 by [pytest-html](#) v4.1.1

Environment

Python	3.13.2
Platform	Windows-10-10.0.19045-SP0
Packages	<ul style="list-style-type: none">pytest: 8.4.1pluggy: 1.6.0
Plugins	<ul style="list-style-type: none">cov: 6.2.1html: 4.1.1metadata: 3.1.1

Summary

37 tests took 13 ms.

(Un)check the boxes to filter the results.

☒ 11 Failed, ☒ 26 Passed, ☒ 0 Skipped, ☒ 0 Expected failures, ☒ 0 Unexpected passes, ☒ 0 Errors, ☒ 0 Reruns

[Show all details](#) / [Hide all details](#)

Result	Test	Duration	Links
Failed	tests/test_diferenca_juros.py::test_numero_decimal	0 ms	
<div><pre>def test_numero_decimal(): > assert calcular_diferenca_juros(1500.50, 4.5, 3.2) == 34.57 E assert 10.89 == 34.57 E + where 10.89 = calcular_diferenca_juros(1500.5, 4.5, 3.2) tests\test_diferenca_juros.py:23: AssertionError</pre></div> <div>expand [+]</div>			
Failed	tests/test_juros_composto.py::test_calcular_juros_tudo_zero_erro	0 ms	
<div><pre>def test_calcular_juros_tudo_zero_erro(): > with pytest.raises(ValueError, match="não podem ser zero"): E Failed: DID NOT RAISE <class 'ValueError'> tests\test_juros_composto.py:45: Failed</pre></div> <div>expand [+]</div>			
Failed	tests/test_juros_composto.py::test_calcular_juros_capital_invalido_fail	0 ms	
<div><pre>def test_calcular_juros_capital_invalido_fail(): # Aqui a função levanta "O capital investido deve ser um número (int ou float)." # mas o match não bate, então o teste vai FALHAR de propósito. with pytest.raises(ValueError, match="Mensagem errada"): > calcular_juros_compostos("ola", 10, 5) tests\test_juros_composto.py:53: ----- ----- ----- capital_investido = 'ola', taxa_juros = 10, tempo = 5 def calcular_juros_compostos(capital_investido:float, taxa_juros:float, tempo:int) -> tuple: # Validação do capital investido</pre></div> <div>expand [+]</div>			
Failed	tests/test_juros_composto.py::test_fail_entrada_basica	0 ms	
<div><pre>def test_fail_entrada_basica(): juros, montante = calcular_juros_compostos(200, 10, 5) > assert round(juros, 2) == 999.99 E assert 122.1 == 999.99 E + where 122.1 = round(122.10200000000009, 2)</pre></div> <div>expand [+]</div>			

Result	Test	Duration	Links
Failed	tests/test_juros_composto.py:58: AssertionError		
Failed	tests/test_juros_composto.py::test_fail_taxa_invalida	0 ms	
<pre>def test_fail_taxa_invalida(): with pytest.raises(ValueError, match="Taxa inválida qualquer"): > calcular_juros_compostos(100, "oi", 5) tests\test_juros_composto.py:64: ----- ----- ----- capital_investido = 100, taxa_juros = 'oi', tempo = 5 def calcular_juros_compostos(capital_investido:float, taxa_juros:float, tempo:int) -> tuple: # Validação do capital investido if not isinstance(capital_investido, (int, float)): raise ValueError("O capital investido deve ser um número (int ou float).")</pre>			
Failed	tests/test_juros_composto.py::test_fail_capital_negativo	0 ms	
<pre>def test_fail_capital_negativo(): with pytest.raises(ValueError, match="Capital não pode ser menor que zero!!!"): > calcular_juros_compostos(-100, 10, 5) tests\test_juros_composto.py:70: ----- ----- ----- capital_investido = -100, taxa_juros = 10, tempo = 5 def calcular_juros_compostos(capital_investido:float, taxa_juros:float, tempo:int) -> tuple: # Validação do capital investido if not isinstance(capital_investido, (int, float)): raise ValueError("O capital investido deve ser um número (int ou float).")</pre>			
Failed	tests/test_juros_composto.py::test_fail_taxa_negativa	0 ms	
<pre>def test_fail_taxa_negativa(): with pytest.raises(ValueError, match="Taxa errada"): > calcular_juros_compostos(100, -10, 5) tests\test_juros_composto.py:75: ----- ----- ----- capital_investido = 100, taxa_juros = -10, tempo = 5 def calcular_juros_compostos(capital_investido:float, taxa_juros:float, tempo:int) -> tuple: # Validação do capital investido if not isinstance(capital_investido, (int, float)): raise ValueError("O capital investido deve ser um número (int ou float).")</pre>			
Failed	tests/test_juros_composto.py::test_fail_tempo_negativo	0 ms	
<pre>def test_fail_tempo_negativo(): with pytest.raises(ValueError, match="tempo negativo errado"): > calcular_juros_compostos(100, 10, -1) tests\test_juros_composto.py:80: ----- ----- ----- capital_investido = 100, taxa_juros = 10, tempo = -1 def calcular_juros_compostos(capital_investido:float, taxa_juros:float, tempo:int) -> tuple: # Validação do capital investido if not isinstance(capital_investido, (int, float)): raise ValueError("O capital investido deve ser um número (int ou float).")</pre>			
Failed	tests/test_juros_simples.py::test_juros_simples_levanta_erro_para_tipos_invalidos[capital_como_texto]	0 ms	

Result 	Test	Duration	Links
	<pre>principal = 'mil', taxa = 5.0, tempo = 2.0 @pytest.mark.error_handling @pytest.mark.parametrize("principal, taxa, tempo", [# Caso 1: Enviando um texto como capital ("mil", 5.0, 2.0), # Caso 2: Enviando uma lista como taxa (1000, [5.0], 2.0), # Caso 3: Enviando uma lista de string como tempo (1000, 5.0, ["dois anos"]),], ids=["capital_como_texto", "juros_como_lista", "tempo_como_lista_de_string"])</pre>		expand [+]
Failed	tests/test_juros_simples.py::test_juros_simples_levanta_erro_para_tipos_invalidos[juros_como_lista]	1 ms	
	<pre>principal = 1000, taxa = [5.0], tempo = 2.0 @pytest.mark.error_handling @pytest.mark.parametrize("principal, taxa, tempo", [# Caso 1: Enviando um texto como capital ("mil", 5.0, 2.0), # Caso 2: Enviando uma lista como taxa (1000, [5.0], 2.0), # Caso 3: Enviando uma lista de string como tempo (1000, 5.0, ["dois anos"]),], ids=["capital_como_texto", "juros_como_lista", "tempo_como_lista_de_string"])</pre>		expand [+]
Failed	tests/test_juros_simples.py::test_juros_simples_levanta_erro_para_tipos_invalidos[tempo_como_lista_de_string]	0 ms	
	<pre>principal = 1000, taxa = 5.0, tempo = ['dois anos'] @pytest.mark.error_handling @pytest.mark.parametrize("principal, taxa, tempo", [# Caso 1: Enviando um texto como capital ("mil", 5.0, 2.0), # Caso 2: Enviando uma lista como taxa (1000, [5.0], 2.0), # Caso 3: Enviando uma lista de string como tempo (1000, 5.0, ["dois anos"]),], ids=["capital_como_texto", "juros_como_lista", "tempo_como_lista_de_string"])</pre>		expand [+]
Passed	tests/test_diferenca_juros.py::test_valores_validos	0 ms	
Passed	tests/test_diferenca_juros.py::test_todos_zeros	0 ms	
Passed	tests/test_diferenca_juros.py::test_juros_e_tempo_zero	0 ms	
Passed	tests/test_diferenca_juros.py::test_tempo_zero	0 ms	
Passed	tests/test_diferenca_juros.py::test_taxa_zero	0 ms	
Passed	tests/test_diferenca_juros.py::test_entrada_string	0 ms	
Passed	tests/test_diferenca_juros.py::test_capital_negativo	0 ms	
Passed	tests/test_diferenca_juros.py::test_taxa_negativa	0 ms	
Passed	tests/test_diferenca_juros.py::test_tempo_negativo	0 ms	
Passed	tests/test_juros_composto.py::test_calcular_juros_entrada_invalida	0 ms	
Passed	tests/test_juros_composto.py::test_calcular_juros_entrada_basica	0 ms	
Passed	tests/test_juros_composto.py::test_calcular_juros_juros_invalido	0 ms	
Passed	tests/test_juros_composto.py::test_calcular_capital_numero_negativo	0 ms	
Passed	tests/test_juros_composto.py::test_calcular_juros_numero_negativo	0 ms	
Passed	tests/test_juros_composto.py::test_calcular_juros_tempo_string	0 ms	

Result ▲	Test	Duration	Links
Passed	tests/test_juros_composto.py::test_calcular_juros_tempo_negativo	0 ms	
Passed	tests/test_juros_simples.py::test_calculo_juros_simples_cenarios_validos[inteiros_padrao]	1 ms	
Passed	tests/test_juros_simples.py::test_calculo_juros_simples_cenarios_validos[longo_periodo]	1 ms	
Passed	tests/test_juros_simples.py::test_calculo_juros_simples_cenarios_validos[valores_decimais]	1 ms	
Passed	tests/test_juros_simples.py::test_calculo_juros_simples_cenarios_validos[taxa_alta]	1 ms	
Passed	tests/test_juros_simples.py::test_calculo_juros_simples_casos_limites[taxa_zero]	1 ms	
Passed	tests/test_juros_simples.py::test_calculo_juros_simples_casos_limites[tempo_zero]	1 ms	
Passed	tests/test_juros_simples.py::test_calculo_juros_simples_casos_limites[principal_zero]	0 ms	
Passed	tests/test_juros_simples.py::test_juros_simples_levanta_erro_para_valores_negativos[principal_negativo]	0 ms	
Passed	tests/test_juros_simples.py::test_juros_simples_levanta_erro_para_valores_negativos[taxa_negativa]	0 ms	
Passed	tests/test_juros_simples.py::test_juros_simples_levanta_erro_para_valores_negativos[tempo_negativo]	0 ms	