**✅ Objetivo de um Teste de Integração**

Enquanto um **teste unitário** verifica uma classe isoladamente (por exemplo, só a Calculadora), o **teste de integração** garante que **duas ou mais classes trabalham corretamente juntas**.

**No caso da Calculadora**

Temos essas duas classes:

1. **Calculadora** → faz as operações matemáticas.
2. **CalculadoraService** → usa a Calculadora para compor uma lógica de negócio.

public int calcularExpressaoSimples(int a, int b) {

int soma = calculadora.somar(a, b);

int multiplicacao = calculadora.multiplicar(a, b);

return soma + multiplicacao;

}

**Teste de Integração (exemplo real)**

@Test

void testExpressaoSimples() {

Calculadora calculadora = new Calculadora();

CalculadoraService service = new CalculadoraService(calculadora);

int resultado = service.calcularExpressaoSimples(2, 3);

// (2 + 3) + (2 \* 3) = 5 + 6 = 11

assertEquals(11, resultado);

}

**O que está sendo testado?**

* A **comunicação entre as classes** Calculadora e CalculadoraService.
* Se a **lógica integrada** entre somar e multiplicar gera o resultado esperado.
* Se existe algum **problema no encadeamento das operações**.

**Diferença para teste unitário:**

| **Tipo de Teste** | **O que testa?** | **Exemplo** |
| --- | --- | --- |
| **Unitário** | Apenas uma classe isolada | Testar só somar(a, b) |
| **Integração** | Interação entre duas ou mais | CalculadoraService usando Calculadora |

**Conclusão**

Esse teste de integração é **simples**, mas é o passo inicial para os alunos entenderem como componentes se conectam. Com o tempo, você pode escalar para:

* Serviços REST
* Conexão com banco de dados
* Integração com APIs externas