

Aqui está uma lista de comandos úteis do **pytest** para testes em Python:

## ★ Comandos básicos:

- `pytest` → Executa todos os testes na pasta atual e subpastas.
  - `pytest arquivo_test.py` → Executa os testes dentro do arquivo especificado.
  - `pytest -k "nome_do_teste"` → Executa testes que contêm "nome\_do\_teste" no nome da função.
  - `pytest -m marca` → Executa apenas testes marcados com `@pytest.mark.marca`.
- 

## □ Opções para saída detalhada:

- `pytest -v` → Mostra saída detalhada (verbose).
  - `pytest -q` → Exibe uma saída mais compacta (quiet).
  - `pytest -s` → Mantém os prints dos testes visíveis.
  - `pytest --tb=short` → Exibe apenas um traceback reduzido em caso de erro.
  - `pytest --tb=long` → Exibe traceback completo.
  - `pytest --maxfail=3` → Interrompe após 3 falhas.
- 

## 🚀 Execução paralela (pytest-xdist):

- `pytest -n auto` → Roda testes em múltiplas CPUs automaticamente.
  - `pytest -n 4` → Roda testes em 4 processos paralelos.
- 

## 🎯 Executando testes específicos:

- `pytest teste_exemplo.py::test_funcao` → Executa apenas a função `test_funcao` dentro do arquivo `teste_exemplo.py`.
  - `pytest -m "marca1 and not marca2"` → Roda testes marcados com `marca1`, mas exclui os marcados com `marca2`.
- 

## ■ Testes em pastas específicas:

- `pytest tests/` → Executa todos os testes dentro da pasta `tests`.
  - `pytest tests/unit/` → Roda apenas os testes da pasta `unit`.
- 

## 📄 Geração de relatórios:

- `pytest --junitxml=report.xml` → Gera um relatório em formato JUnit XML.
  - `pytest --html=report.html` → Gera um relatório HTML (requer **pytest-html**).
-

## 🔗 Testes contínuos e falhas anteriores:

- `pytest --lf` → Executa apenas os testes que falharam na última execução.
  - `pytest --ff` → Executa primeiro os testes que falharam na última execução.
- 

## 🔍 Cobertura de código (pytest-cov):

- `pytest --cov=meu_projeto` → Mede a cobertura de código da pasta `meu_projeto`.
  - `pytest --cov-report=html` → Gera um relatório da cobertura em HTML.
- 

## 🔧 Depuração de testes:

- `pytest --pdb` → Abre o depurador (PDB) quando um teste falha.
- `pytest --disable-warnings` → Oculta avisos.