🧪 Relatório de Testes - TaskCrafter CLI

📊 Resumo da Suite de Testes

Estatísticas Gerais

- **Total de Testes:** 56 testes
- **Testes Aprovados:** 56 (100% ✅)
- **Testes Falhados:** 0
- **Cobertura de Código:** 94.81% 🎯
- **Threshold Mínimo:** 80% (✅ ALCANÇADO)

Distribuição dos Testes

Testes Unitários (43 testes)

1. `test_models.py` - 18 testes

Testa o modelo de dados `Task`:
- ✅ Criação de tarefas (4 testes)
- ✅ Validações de dados (8 testes)
- ✅ Serialização/Deserialização (3 testes)
- ✅ Representação em string (3 testes)

2. `test_manager.py` - 25 testes

Testa o `TaskManager`:
- ✅ Operações básicas (6 testes)
- ✅ Persistência em JSON (3 testes)
- ✅ Operações CRUD (6 testes)
- ✅ Filtros e ordenação (8 testes)
- ✅ Estatísticas (2 testes)

Testes de Integração/E2E (13 testes)

3. `test_integration.py` - 13 testes

Testa fluxos completos da CLI:
- ✅ Fluxos básicos de uso (5 testes)
- ✅ Casos extremos e erros (4 testes)
- ✅ Operações com datas (3 testes)
- ✅ Persistência entre sessões (1 teste)

📈 Cobertura Detalhada por Módulo

Módulo	Linhas	Cobertas	Cobertura
`taskcrafter/__init__.py`	2	2	100.00% ✅
`taskcrafter/models.py`	46	46	100.00% ✅
`taskcrafter/manager.py`	90	89	98.89% ✅
`taskcrafter/cli.py`	150	137	91.33% ✅
`taskcrafter/__main__.py`	1	0	0.00% ⚠️
TOTAL	**289**	**274**	**94.81% ✅**

Linhas Não Cobertas

As únicas linhas não cobertas são:
- `__main__.py`: Ponto de entrada quando executado diretamente
- `cli.py`: Alguns blocos de tratamento de exceção específicos e mensagens de ajuda

Essas linhas são difíceis de testar de forma automatizada sem afetar a estrutura do código.

🎯 Casos de Teste Importantes

Validações de Negócio

1. ✅ Título único - Não permite tarefas com mesmo título
2. ✅ Prioridades válidas - Aceita apenas: baixa, media, alta
3. ✅ Status válidos - Aceita apenas: pendente, andamento, concluida
4. ✅ Datas no formato ISO - YYYY-MM-DD obrigatório
5. ✅ Título não vazio - Rejeita títulos vazios ou só com espaços

Operações CRUD

1. ✅ Criar tarefas com todos os campos
2. ✅ Ler/buscar tarefas por título (case-insensitive)
3. ✅ Atualizar campos individuais
4. ✅ Deletar tarefas existentes
5. ✅ Listar com filtros múltiplos

Persistência

1. ✅ Salvar alterações no JSON
2. ✅ Carregar dados entre sessões
3. ✅ Recuperar de arquivos corrompidos

Casos Extremos

1. ✅ Tentar criar tarefa duplicada (erro)
2. ✅ Atualizar tarefa inexistente (erro)
3. ✅ Deletar tarefa inexistente (retorna False)
4. ✅ Listar quando não há tarefas
5. ✅ Filtros sem resultados

🚀 Executando os Testes

Todos os testes

```
pytest
```

Com relatório de cobertura

```
pytest --cov=taskcrafter --cov-report=term-missing
```

Apenas testes unitários

```
pytest tests/test_models.py tests/test_manager.py
```

Apenas testes de integração

```
pytest tests/test_integration.py
```

Modo verbose

```
pytest -v
```

Com relatório HTML

```
pytest --cov=taskcrafter --cov-report=html
# Abra htmlcov/index.html no navegador
```

📋 Checklist de Boas Práticas

- [x] Testes unitários focados e isolados
- [x] Uso de fixtures do pytest
- [x] Nomes descritivos de testes
- [x] Testes de casos de sucesso e erro
- [x] Testes de integração simulando uso real
- [x] Cobertura ≥ 80%
- [x] Testes rápidos (< 5 segundos total)
- [x] Sem dependências externas nos testes
- [x] Testes determinísticos (sem flakiness)

- [x] Documentação clara de cada teste

🎓 Requisitos Acadêmicos Atendidos

Quantidade de Testes

- [x] ≥ 30 testes unitários (43 implementados - **143%**)
- [x] ≥ 5 testes de integração/e2e (13 implementados - **260%**)

Qualidade dos Testes

- [x] Testes focados em uma funcionalidade
- [x] Nomes descritivos seguindo padrão `test_<ação>_<resultado>`
- [x] Uso apropriado de fixtures
- [x] Cobertura de casos de borda
- [x] Testes de validação

Cobertura

- [x] Cobertura ≥ 80% (94.81% alcançado - **118%**)
- [x] Configuração adequada do pytest-cov
- [x] Relatórios em múltiplos formatos

CI/CD

- [x] Testes automatizados no GitHub Actions
- [x] Múltiplas plataformas (Linux, macOS, Windows)
- [x] Múltiplas versões Python (3.10, 3.11, 3.12)
- [x] Upload automático para Codecov

🏆 Métricas de Qualidade

Métrica	Objetivo	Alcançado	Status
Testes Unitários	≥ 30	43	✅ 143%
Testes Integração	≥ 5	13	✅ 260%
Cobertura	≥ 80%	94.81%	✅ 118%
Taxa de Sucesso	100%	100%	✅ 100%
Tempo Execução	< 10s	~4s	✅ 60% mais rápido

📝 Conclusão

O projeto **TaskCrafter CLI** demonstra excelência em práticas de teste de software:

1. **Cobertura Superior**: 94.81% ultrapassa o mínimo de 80%
2. **Suite Robusta**: 56 testes cobrindo todos os cenários
3. **Qualidade**: 100% de testes aprovados
4. **Boas Práticas**: Fixtures, nomes descritivos, testes focados

5. **Automação**: CI/CD completo com múltiplas plataformas

O sistema está **pronto para produção** com alta confiabilidade e manutenibilidade.

📅 **Última atualização:** 24 de Novembro de 2025
👨‍💻 **Autor:** Espezzialy Raphael Oliveira Souza
🎓 **Projeto:** Disciplina de Teste de Software