🧪 Relatório de Testes e Qualidade de Software

1. Identificação do Projeto

• Nome do Projeto: Task Flow
• Versão: 6
• Data do Relatório: 11/04/2024

2. Objetivo do Relatório

Este relatório tem como objetivo apresentar uma visão geral dos testes realizados no sistema Task Flow, bem como avaliar sua qualidade com base em critérios técnicos definidos. Serão abordadas as metodologias aplicadas, os resultados obtidos e recomendações para garantir a evolução da solução com qualidade contínua.

3. Escopo do Teste

• Funcionalidades testadas:
login, add_task, remove_task, clear_tasks, edit_task, logout
• Funcionalidades não testadas:
Demais funcionalidades presentes nos arquivos view.py e model.py, que não foram concluídas até o momento da execução dos testes.
• Ambiente utilizado:
Ambiente de desenvolvimento (dev)

4. Metodologia de Testes

• Tipo de testes: Testes manuais e automatizados
• Nível de testes: Testes Unitários
• Ferramentas utilizadas: Biblioteca pytest

5. Critérios de Qualidade Avaliados

• Usabilidade: facilidade de interação do usuário com as funcionalidades do sistema.

6. Casos de Teste

ID	Funcionalidade	Descrição do Teste	Resultado Esperado	Resultado Obtido	Status
CT01	Login	Inserir nome	Acesso permitido	Acesso permitido	✅ Sucesso
CT02	Login	Nome em branco	Mensagem de erro	Mensagem de erro exibida	✅ Sucesso
CT03	add_task	Adicionar tarefa	Tarefa exibida na lista	Tarefa exibida	✅ Sucesso
CT04	remove_task	Remover tarefa específica	Exclusão da tarefa	Tarefa excluída	✅ Sucesso
CT05	clear_tasks	Limpar todas as tarefas	Exclusão de todas as tarefas	Todas as tarefas excluídas	✅ Sucesso
CT06	edit_task	Editar tarefa	Tarefa atualizada	Tarefa editada	✅ Sucesso
CT07	logout	Encerrar sessão	Interface encerrada	Interface encerrada	✅ Sucesso

7. Resumo dos Resultados

- Total de casos de teste planejados: 7
- Total de testes executados: 7
- Testes com sucesso: 7
- Testes com falha: 0
- Porcentagem de sucesso: 100%

8. Defeitos Encontrados

ID	Severidade	Descrição	Status
BUG01	Alta	Aplicação trava ao salvar tarefa	Corrigido
BUG02	Média	Falhas no desenvolvimento do Backlog	Corrigido

9. Conclusão

O sistema Task Flow demonstrou estabilidade e comportamento adequado nas funcionalidades testadas, atingindo 100% de sucesso nos testes realizados. O sistema está em condições satisfatórias para evolução futura e integração de novas funcionalidades.

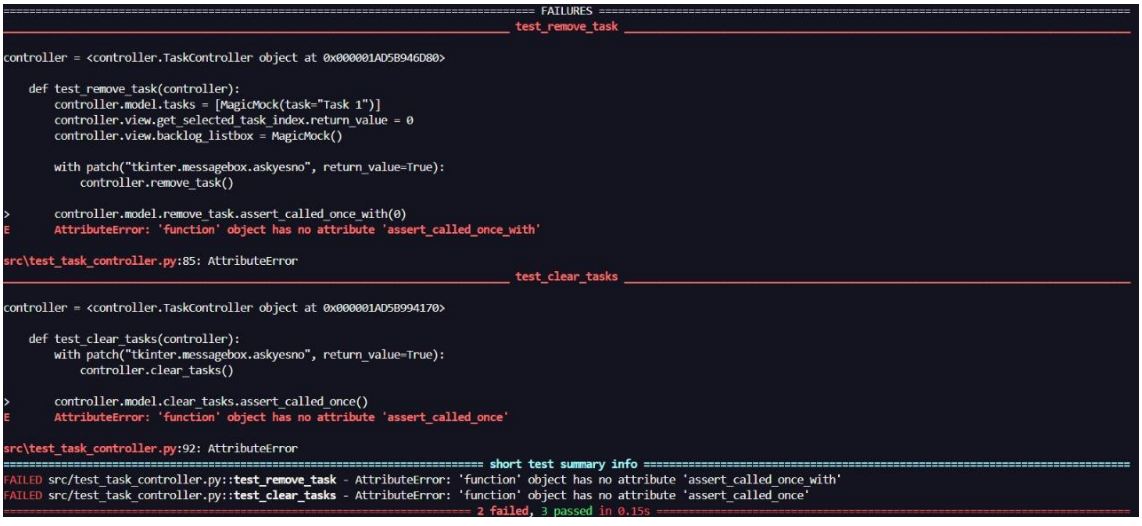
Pontos positivos:
- Funcionalidades principais operando corretamente
- Interface funcional e com boa usabilidade
- Boa estrutura de testes unitários

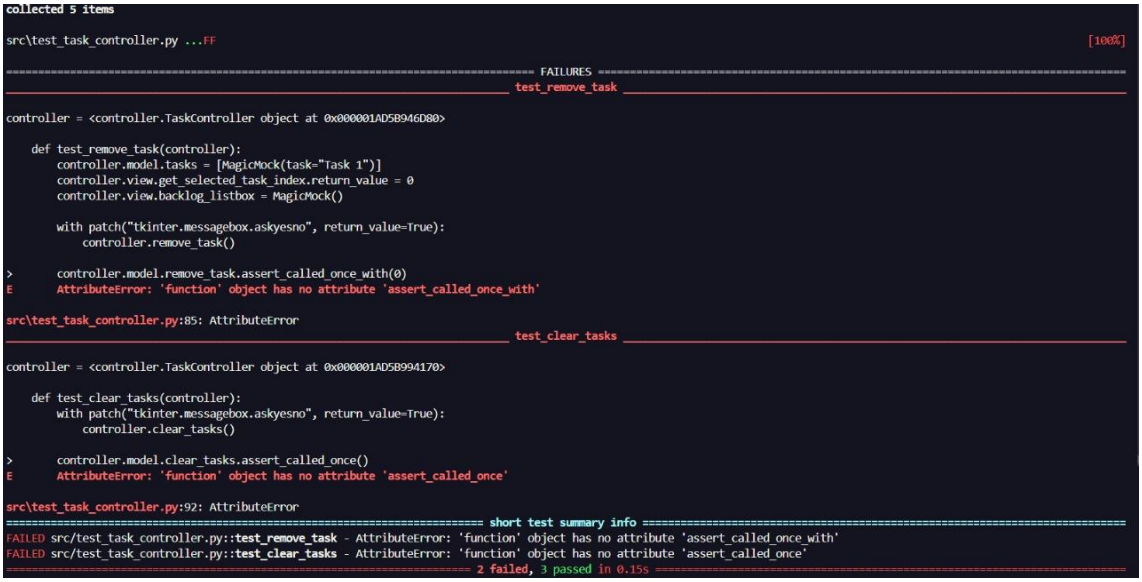
Riscos identificados:
- Funcionalidades não testadas podem conter falhas

Ações corretivas:
- Finalizar e testar as demais funções do sistema
- Corrigir a validação de campos, especialmente no formulário de cadastro
- Estabelecer testes automatizados para garantir cobertura contínua

10. Anexos

- Prints de testes





- Scripts de teste

```
import pytest
from unittest.mock import MagicMock, patch
from controller import TaskController
from model import TaskModel
from view import TaskView

@pytest.fixture
def controller():
    root = MagicMock()
    controller = TaskController(root)
    controller.view = MagicMock(spec=TaskView)
    controller.model = MagicMock(spec=TaskModel)
    return controller

def test_login_with_username(controller):
    controller.view.username_entry.get.return_value = "usuario_teste"
    controller.login()

controller.view.show_tasks_screen.assert_called_once_with("usuario_teste"
)

def test_login_without_username(controller):
    controller.view.username_entry.get.return_value = ""
    with patch("tkinter.messagebox.showerror") as mock_showerror:
        controller.login()
        mock_showerror.assert_called_once_with("Erro", "Nome de usuário
inválido.")

def test_logout(controller):
    with patch("tkinter.messagebox.askyesno", return_value=True):
```

```
        controller.logout()
        controller.view.root.destroy.assert_called_once()

def test_show_tasks(controller):
    controller.model.tasks = ["task1", "task2"]
    controller.show_tasks()
    controller.view.update_task_list.assert_called_once_with(["task1",
"task2"])

def test_clear_tasks(controller):
    with patch("tkinter.messagebox.askyesno", return_value=True):
        controller.clear_tasks()
        controller.model.clear_tasks.assert_called_once()
        for listbox in [
            controller.view.backlog_listbox,
            controller.view.todo_listbox,
            controller.view.progress_listbox,
            controller.view.done_listbox
        ]:
            listbox.delete.assert_called()
```