



Projeto A3

Gerenciador de Tarefas

Documentação de Desenvolvimento de Software



Disciplina

UC – Gestão e Qualidade de Software

Prof. Calvetti



Aluno

Bruno Rodrigues Reis

RA: 8222243147



Objetivos do Projeto

- ✓ Demonstrar aplicação prática de conceitos de engenharia de software
- ✓ Elaborar documentação completa de desenvolvimento
- ✓ Implementar sistema funcional com testes abrangentes
- ✓ Aplicar gestão de configuração e controle de qualidade



Metodologia e Planejamento

Abordagem sistemática baseada em engenharia de software



Abordagem Metodológica

- Processo iterativo e incremental
- Baseado em Pressman & Maxim
- Foco na qualidade e documentação



Cronograma Geral

| | |
|-------------------|------------------|
| Duração Total: | 4 semanas |
| Esforço Estimado: | 60 horas |
| Equipe: | Multidisciplinar |

Fases do Projeto



Análise & Design

Semana 1

Requisitos, arquitetura e especificações



Implementação

Semanas 2-3

Desenvolvimento backend e frontend



Testes & Docs

Semana 4

Validação e documentação final

Recursos Necessários



Humanos

Desenvolvedores, Analistas, Testadores



Tecnológicos

Flask, Git, Ferramentas de Teste



Infraestrutura

Ambientes de Dev, Teste e Produção



Tecnologias Escolhidas

Stack tecnológico otimizado para praticidade e eficiência



Flask Framework

Microframework Python para desenvolvimento web

Escolha Principal

Backend



Python + Flask

Framework web minimalista



SQLite

Banco de dados leve



Frontend



HTML5 + CSS3

Estrutura e estilização



JavaScript

Interatividade e APIs



Justificativa da Escolha do Flask



Praticidade

Microframework leve e de fácil aprendizado



Flexibilidade

Permite escolha de bibliotecas conforme necessidade



Comunidade

Ampla documentação e suporte ativo



Integração

Fácil integração com tecnologias frontend

Ferramentas de Apoio



Git



GitHub



Flask-CORS



Especificação de Requisitos

Definição clara das funcionalidades e características de qualidade



Requisitos Funcionais

RF001 Criação de Tarefas
Título obrigatório e descrição opcional

RF002 Visualização de Tarefas
Lista organizada e responsiva

RF003 Marcação de Conclusão
Alternar status concluído/pendente

RF004 Exclusão de Tarefas
Remoção com confirmação



Requisitos Não Funcionais

RNF001 Usabilidade
Interface intuitiva e responsiva

RNF002 Performance
Resposta em menos de 2 segundos

RNF003 Compatibilidade
Principais navegadores web

RNF004 Confiabilidade
Persistência segura de dados



Escopo do Sistema



Incluído no Escopo

- CRUD completo de tarefas
- Interface web responsiva
- Persistência em banco SQLite
- Validação de dados



Fora do Escopo

- Autenticação de usuários
- Compartilhamento de tarefas
- Notificações e lembretes
- Integração com calendários



Arquitetura do Sistema



Frontend
HTML/CSS/JS



Backend
Flask API



Database
SQLite

</> Implementação do Sistema

Sistema funcional demonstrando conceitos de engenharia de software



Interface do Sistema

Gerenciador de Tarefas

Organize suas atividades

Nova Tarefa

Título da tarefa

Adicionar

Implementar casos de teste

Concluir

Excluir



Arquitetura Implementada



Frontend

Interface responsiva com HTML5, CSS3 e JavaScript vanilla



Backend API

Flask com endpoints RESTful e CORS habilitado



Persistência

SQLite com SQLAlchemy ORM para modelagem de dados



Funcionalidades Implementadas



Criar

Novas tarefas com validação



Visualizar

Lista organizada e responsiva



Concluir

Marcar como feito/reabrir



Excluir

Remoção com confirmação



Destaques Técnicos

- ✓ API RESTful com Flask
- ✓ Validação frontend e backend
- ✓ Interface responsiva
- ✓ Persistência com SQLAlchemy



Status do Projeto

Desenvolvimento:

Concluído

Testes Funcionais:

Aprovado

Documentação:

Completa

Deploy:

Disponível

Planejamento de Testes

Estratégia abrangente para garantia da qualidade

Estratégia de Testes

- ✓ **Abordagem Sistemática**
Baseada em Gonçalves et al. e melhores práticas
- ✓ **Cobertura Completa**
Testes funcionais e não funcionais
- ✓ **Execução Estruturada**
Cronograma detalhado com marcos definidos

Métricas do Plano

12
Casos de Teste

25
Dias de Execução

7
Marcos Definidos

100%
Cobertura RF

Ferramentas de Teste

- **Testes Unitários:** `unittest`, `pytest` (Python)
- **Testes de Integração/Funcionais:** `Selenium`, `Cypress` (Web), `Postman` (API)
- **Testes de Performance:** `JMeter`, `Locust`
- **Gestão de Testes:** Planilhas Excel (para cronograma e casos simples)

Casos de Teste Principais

| | |
|-----------------------------------|----------|
| CT001 - Criação com título válido | Aprovado |
| CT003 - Validação de campos | Aprovado |
| CT004 - Marcação de conclusão | Aprovado |
| CT006 - Exclusão com confirmação | Aprovado |
| CT011 - Responsividade móvel | Aprovado |

Critérios de Aprovação

- ✓ **100% Funcionais**
Todos os casos funcionais devem passar
- ✓ **90% Interface**
Mínimo para testes de usabilidade
- ✓ **Zero Críticos**
Nenhum defeito crítico pendente



Cronograma Detalhado de Testes

Visão completa das atividades de teste e seus marcos

| ID | Atividade | Descrição | Responsável | Recursos | Data Início | Data Fim | Duração (dias) | Dependências | Marco | Status |
|------|---------------------------------|---|---------------------------|--|-------------|------------|----------------|---|-------------------------------|-----------|
| T001 | Planejamento de Testes | Definição da estratégia de testes e elaboração do plano de testes | Analista de Testes | Analista de Testes Senior | 15/06/2025 | 16/06/2025 | 2 | Conclusão dos requisitos | Plano de Testes Aprovado | Planejado |
| T002 | Criação de Casos de Teste | Elaboração dos casos de teste para todas as funcionalidades | Analista de Testes | Analista de Testes Junior, Analista de Testes Senior | 17/06/2025 | 19/06/2025 | 3 | T001 - Planejamento de Testes | Casos de Teste Criados | Planejado |
| T003 | Preparação do Ambiente de Teste | Configuração do ambiente de teste e dados de teste | DevOps | DevOps Engineer, Servidor de Teste | 17/06/2025 | 17/06/2025 | 1 | T001 - Planejamento de Testes | Ambiente de Teste Configurado | Planejado |
| T004 | Testes Unitários | Execução de testes unitários para validar componentes individuais | Desenvolvedor | Desenvolvedor Backend, Desenvolvedor Frontend | 18/06/2025 | 19/06/2025 | 2 | T003 - Preparação do Ambiente | Testes Unitários Concluídos | Planejado |
| T005 | Testes de Integração | Validação da integração entre componentes do sistema | Analista de Testes | Analista de Testes Senior, Desenvolvedor | 20/06/2025 | 22/06/2025 | 3 | T004 - Testes Unitários | Integração Validada | Planejado |
| T006 | Testes Funcionais | Execução de testes funcionais baseados nos casos de teste | Analista de Testes | Analista de Testes Junior, Analista de Testes Senior | 23/06/2025 | 26/06/2025 | 4 | T002 - Criação de Casos de Teste, T005 - Testes de Integração | Funcionalidades Validadas | Planejado |
| T007 | Testes de Usabilidade | Avaliação da experiência do usuário e interface | UX Designer | UX Designer, Usuários de Teste | 27/06/2025 | 28/06/2025 | 2 | T006 - Testes Funcionais | Usabilidade Aprovada | Planejado |
| T008 | Testes de Performance | Validação do desempenho e capacidade do sistema | Analista de Performance | Analista de Performance, Ferramentas de Load Test | 29/06/2025 | 30/06/2025 | 2 | T006 - Testes Funcionais | Performance Validada | Planejado |
| T009 | Testes de Segurança | Verificação de vulnerabilidades e aspectos de segurança | Especialista em Segurança | Especialista em Segurança, Ferramentas de Segurança | 01/07/2025 | 02/07/2025 | 2 | T006 - Testes Funcionais | Segurança Validada | Planejado |
| T010 | Correção de Defeitos | Correção dos defeitos encontrados durante os testes | Desenvolvedor | Desenvolvedor Backend, Desenvolvedor Frontend | 03/07/2025 | 05/07/2025 | 3 | T007, T008, T009 - Testes Especializados | Defeitos Corrigidos | Planejado |
| T011 | Testes de Regressão | Validação de que as correções não introduziram novos defeitos | Analista de Testes | Analista de Testes Senior, Ferramentas de Automação | 06/07/2025 | 07/07/2025 | 2 | T010 - Correção de Defeitos | Regressão Validada | Planejado |
| T012 | Testes de Aceitação | Validação final com o cliente/usuário final | Product Owner | Product Owner, Usuários Finais | 08/07/2025 | 09/07/2025 | 2 | T011 - Testes de Regressão | Sistema Aceito | Planejado |
| T013 | Relatório Final de Testes | Elaboração do relatório final com resultados e métricas | Analista de Testes | Analista de Testes Senior | 10/07/2025 | 10/07/2025 | 1 | T012 - Testes de Aceitação | Relatório de Testes Entregue | Planejado |

Gestão de Configuração

Controle sistemático de mudanças e versionamento

Controle de Versões

Git + GitHub

Sistema distribuído com repositório remoto

Estratégia de Branches

Main protegido, develop para integração


Versionamento Semântico

Tags v1.0.0, v1.1.0 para releases

Itens de Configuração

 Código Fonte

Controlado

 Documentação

Versionado

 Configurações

Rastreado

 Dados de Teste

Incluído

Fluxo de Trabalho Git



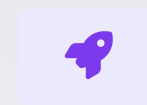
Desenvolvimento
Feature branches



Pull Request
Code review



Merge
Branch principal



Release
Tag + Deploy

Controle de Mudanças

Solicitação Formal

Descrição, justificativa e impacto

Avaliação de Impacto

Análise técnica e de recursos

Aprovação e Verificação

Stakeholders e testes pós-implementação

Backup e Recuperação

Replicação Automática

Múltiplas localizações e nuvem

Histórico Completo

Recuperação de qualquer versão

Procedimentos Testados

Recuperação validada regularmente



Resultados e Conclusões

Síntese dos resultados obtidos e lições aprendidas

Projeto Concluído com Sucesso

Todos os objetivos alcançados dentro do prazo estabelecido

100%

Requisitos Atendidos

12/12

Casos de Teste Aprovados

4

Semanas de Desenvolv.

7/7

Marcos Atingidos



Principais Conquistas



Sistema Funcional Completo

Todas as funcionalidades implementadas e testadas



Documentação Abrangente

Todos os artefatos exigidos pelo Projeto A3



Qualidade Validada

Testes sistemáticos e controle de qualidade



Gestão de Configuração

Controle completo de versões e mudanças



Lições Aprendidas



Planejamento é Fundamental

Decisões bem fundamentadas facilitam execução



Requisitos Claros

Especificação detalhada evita ambiguidades



Testes Sistemáticos

Identificação precoce garante qualidade



Controle de Mudanças

Ferramentas adequadas facilitam gestão



Contribuições do Projeto

- + Demonstração prática de conceitos teóricos
- + Sistema funcional como prova de conceito
- + Documentação como referência futura
- + Aplicação das bibliografias da disciplina



Trabalhos Futuros

- ➔ Funcionalidades avançadas (autenticação, colaboração)
- ➔ Aplicação de metodologias ágeis
- ➔ Implementação de testes automatizados
- ➔ Arquiteturas mais complexas (microserviços)



Agradecimentos

Projeto desenvolvido para a disciplina UC – Gestão e Qualidade de Software, demonstrando a aplicação prática dos conceitos estudados.



Universidade São Judas Tadeu



Prof. Calvetti



Junho 2025