

## A3 Gestão de qualidade de software

Nome, RA, Email, Turma e Curso

Guilherme Sandoli, 82324873 ADS1AN-BUC1-6272430

[guisandolidna@gmail.com](mailto:guisandolidna@gmail.com) Análise e desenvolvimento de sistemas

Guilherme Camargo Lima, 823147783 ADS1AN-BUC1-6272430

[guilhermeclima.09@gmail.com](mailto:guilhermeclima.09@gmail.com) Análise e desenvolvimento de sistemas

Giovanna Rodrigues Milu 823151263 ADS1AN-BUC1-6272430

[gigimilu@gmail.com](mailto:gigimilu@gmail.com) Análise e desenvolvimento de sistemas

Jadiel Santos de Souza 823212714 ADS1AN-BUC1-6272430

[jadiel174@hotmail.com](mailto:jadiel174@hotmail.com) Análise e desenvolvimento de sistemas

### Sumário

#### 1. Planejamento de testes de software:

- 1.1. Cronograma de atividades;
- 1.2. Alocação de recursos;
- 1.3. Marcos do projeto.

#### 2. Documentos de desenvolvimento de software:

- 2.1. Plano de projeto:
  - 2.1.1. Planejamento do projeto;
  - 2.1.2. Escopo;
  - 2.1.3. Recursos; e
  - 2.1.4. Estimativas de projeto.
- 2.2. Documento de requisitos;
- 2.3. Planejamento de testes:
  - 2.3.1 Plano de testes:
    - 2.3.1.1. Introdução;
    - 2.3.1.2. Escopo;
    - 2.3.1.3. Objetivos;
    - 2.3.1.4. Requisitos a serem testados;
    - 2.3.1.5. Estratégias, tipos de testes e ferramentas a serem utilizadas;
    - 2.3.1.6. Recursos a serem empregados;
    - 2.3.1.7. Cronograma das atividades; e
    - 2.3.1.8. Definição dos marcos do projeto.
  - 2.3.2. Casos de testes; e
  - 2.3.3. Roteiro de testes.

#### 3. Gestão de configuração de software;

#### 4. Repositório de gestão de configuração de software.

1.Planejamento de testes de software:

O objetivo principal dos testes é verificar se o software de gerenciamento de tarefas atende aos requisitos de funcionalidade, desempenho, segurança e usabilidade do software. Serão testadas as funções do teste: Criar, Excluir, Editar e Visualizar as tarefas de forma correta.  
É necessário um ambiente de teste configurado e integrado com banco de dados

1.1 Cronograma de atividades;

Atividade	Início	Fim
Planejamento de Testes	15/11/2024	17/11/2024
Execução de Testes Funcionais	18/11/2024	22/11/2024
Execução de Testes de Usabilidade	23/11/2024	25/11/2024
Execução de Testes de Performance	26/11/2024	27/11/2024
Execução de Testes de Segurança	28/11/2024	29/11/2024
Execução de Testes de Integração	30/11/2024	02/12/2024
Testes de Regressão e Aceitação	03/12/2024	05/12/2024
Análise dos Resultados e Relatório	06/12/2024	08/12/2024

1.2 Alocação de recursos;

Equipe de testes: Gerente de Qualidade (QA), testadores manuais, testadores automatizados e desenvolvedores.  
Ferramentas: Trello, Jira e Selenium.  
Ambiente de desenvolvimento: VSCode e MySQL

1.3 Marcos do projeto.

Marco	Data Estimada	Objetivos
Planejamento de Testes Concluído	17/11/2024	Definir plano de testes, alocar recursos e cronograma.
Entrega para Testes Funcionais	18/11/2024	Disponibilizar o sistema para os testes iniciais.
Conclusão de Testes Funcionais/Usabilidade	25/11/2024	Validar funcionalidades e a experiência do usuário.
Conclusão de Testes de Performance	27/11/2024	Testar desempenho sob carga e identificar gargalos.
Conclusão de Testes de Segurança	29/11/2024	Validar a segurança da aplicação e corrigir vulnerabilidades.
Conclusão de Testes de Integração	02/12/2024	Validar integração com sistemas externos.
Conclusão de Testes de Regressão e Aceitação	05/12/2024	Validar que todas as funcionalidades estão estáveis e aceitas.
Relatório Final e Encerramento	08/12/2024	Finalizar o ciclo de testes e preparar a entrega do sistema.

## 2. Documentos de desenvolvimento de software:

### 2.1.1 Plano de projeto:

O objetivo deste projeto é planejar, executar e garantir a qualidade do software de gerenciamento de tarefas através de testes de funcionalidade, usabilidade, desempenho, segurança, integração e aceitação, assegurando que ele esteja pronto para uso pelos usuários finais e atenda aos requisitos do cliente.

### 2.1.2 Escopo;

Desenvolver um software de gerenciamento de tarefas para equipes e indivíduos, com o objetivo de otimizar a organização de atividades, priorizar tarefas, acompanhar prazos e facilitar a colaboração entre membros. O sistema deve ser acessível via web e responsivo para diferentes dispositivos.

### 2.1.3 Recursos

Os recursos utilizados foram o VScode e o MySQL

### 2.1.4 Estimativas de projeto.

Essa é a estimativa do projeto

Fase	Atividades	Duração Estimada
Planejamento	Definição de escopo, análise de requisitos	2 semanas
Design e Prototipação	Criação de wireframes, protótipos e layout visual	3 semanas
Desenvolvimento	Backend, frontend, integração entre sistemas	8 semanas
Testes e Validação	Testes unitários, integração, usabilidade	2 semanas
Implantação	Configuração do ambiente de produção e deploy	1 semana
Manutenção	Correções pós-implantação e ajustes	Contínuo

## 2.2 Documento de requisitos;

#	Processo	(RF) ou (RNF)	Requisitos
1	Adicionar Tarefas	RF	Os Usuários devem poder adicionar novas tarefas podendo escolher o status da tarefa e definir um responsável a ela.
2	Ler Tarefa	RF	O Usuário deve poder visualizar as tarefas criadas a qualquer momento.
3	Atualizar Tarefas	RF	O usuário deve poder editar as tarefas a qualquer momento podendo mudar o status e a descrição delas.
4	Deletar Tarefas	RF	O usuário deve poder deletar as tarefas apenas informando o ID dela.
5	Usabilidade	RNF	O sistema deve ser fácil de usar e intuitivo, permitindo que os usuários interajam com as funcionalidades de forma eficiente
6	Desempenho	RNF	O sistema deve ser rápido e responsivo, permitindo que os usuários atualizem e visualizem as informações dos cartões sem atrasos significativos
7	Segurança	RNF	O sistema deve garantir a segurança dos dados dos usuários, protegendo as informações pessoais e impedindo o acesso não autorizado.
8	Disponibilidade	RNF	O sistema deve estar disponível e acessível aos usuários a maior parte do tempo, minimizando o tempo de inatividade
9	Escalabilidade	RNF	O sistema deve ser capaz de lidar com o aumento significativo no número tarefas sem comprometer o desempenho

## 2.3 Plano de Testes

### 2.3.1.1 Introdução;

Garantir que o software de gerenciamento de tarefas funcione conforme o esperado, atendendo aos requisitos funcionais e não funcionais, e oferecendo uma experiência de usuário estável, segura e intuitiva. O plano de testes abrange todas as funcionalidades principais do sistema, com foco na detecção de bugs, problemas de usabilidade, e na validação de desempenho e segurança.

#### 2.3.1.2 Escopo;

O software de gerenciamento de tarefas terá como principal objetivo auxiliar equipes e indivíduos na organização de suas atividades e projetos, proporcionando um ambiente centralizado para a criação, edição e acompanhamento de tarefas.

#### 2.3.1.3 Objetivos;

Os objetivos do plano de testes para o software de gerenciamento de tarefas são garantir que o sistema funcione conforme o planejado e proporcione uma experiência de usuário confiável, intuitiva e segura. Isso envolve verificar a funcionalidade de todas as principais características do sistema, como a criação, edição e atribuição de tarefas, além de assegurar que o sistema atenda aos requisitos de desempenho e segurança.

#### 2.3.1.4 Requisitos a serem testados;

Funcionalidades Principais

Usabilidade

Desempenho

Integrações

Manutenção e Atualizações

#### 2.3.1.5. Estratégias, tipos de testes e ferramentas a serem utilizadas;

##### Estratégia baseada em Requisitos

Essa estratégia foca em validar se o software atende a todas as funcionalidades e características especificadas nos requisitos do projeto. O processo é guiado pelos documentos de requisitos funcionais e não funcionais.

##### Estratégia automação de Testes

Essa estratégia busca automatizar casos de teste repetitivos e críticos, reduzindo o esforço manual, acelerando o processo e aumentando a confiabilidade dos testes.

##### Testes Funcionais

Os testes funcionais têm como objetivo verificar se o software atende aos requisitos funcionais especificados, validando cada funcionalidade do sistema em relação ao que foi definido no escopo. Esse tipo de teste avalia o comportamento do sistema sob condições normais e excepcionais.

##### Testes de Usabilidade

Os testes de usabilidade avaliam a experiência do usuário com o software, garantindo que o sistema seja intuitivo, fácil de usar e adequado às expectativas dos usuários finais.

Ferramentas a serem utilizadas

Selenium: Para automação de testes de interface web.

Jira: Para planejar, documentar e acompanhar casos de teste.

Microsoft Teams: Para colaboração entre equipes durante o teste.

2.3.1.6 Recursos a serem empregados;

Os recursos necessários para conduzir os testes em um software de gerenciamento de tarefas incluem equipe, infraestrutura, ferramentas e documentação.

Recursos Humanos

Pessoas qualificadas para planejar, executar e analisar os testes.

Gerente de Testes, Analista de Testes, Automatizador de Testes, Especialista em Usabilidade e Desenvolvedores

Recursos Tecnológicos

Infraestrutura e ferramentas necessárias para a execução dos testes.

Documentação

Materiais e artefatos que orientam e registram o processo de testes, como o plano de testes e casos de testes.

2.3.1.7. Cronograma das atividades

O cronograma de testes é estruturado para garantir que cada etapa seja concluída no tempo planejado, sem comprometer a qualidade do software. A seguir está um exemplo de cronograma para um projeto de gerenciamento de tarefas. Esse cronograma considera um ciclo de desenvolvimento de 8 semanas, que pode ser ajustado conforme a complexidade do projeto.

Semana 1: Planejamento

Semana 2: Criação de Casos de Teste

Semana 3 a 4: Execução de Testes Funcionais

Semana 5: Testes de Usabilidade

Semana 6: Testes de Desempenho e Segurança

Semana 7: Testes de Integração e Regressão

Semana 8: Testes de Aceitação e Relatórios Finais

Semana	Atividades	Duração	Responsáveis
1	Planejamento	5 dias	Gerente de testes, analistas
2	Criação de casos de teste	5 dias	Analistas, automatizadores
3-4	Execução de testes funcionais	10 dias	Analistas, desenvolvedores
5	Testes de usabilidade	5 dias	Especialistas, usuários
6	Testes de desempenho e segurança	5 dias	Especialistas
7	Testes de integração e regressão	5 dias	Automatizadores, analistas
8	Testes de aceitação e relatórios finais	5 dias	Stakeholders, gerente de testes

#### 2.3.1.8. Definição dos marcos do projeto.

Marco	Prazo Estimado	Entrega Associada	Responsáveis
Início do Projeto	Semana 1	Documento de requisitos	Gerente de projeto, stakeholders
Planejamento de Testes	Semana 1	Plano de testes aprovado	Gerente de testes
Criação de Casos de Teste	Semana 2	Casos de teste documentados	Analistas de testes
Primeira Rodada de Testes	Semana 4	Relatório de bugs	Analistas de testes
Feedback de Usabilidade	Semana 5	Relatório de usabilidade	Especialistas em UX, usuários finais
Validação de Desempenho	Semana 6	Relatórios de desempenho e segurança	Especialistas técnicos
Testes de Integração e Regressão	Semana 7	Relatório de testes de integração e regressão	Automatizadores, analistas
Testes de Aceitação	Semana 8	Documento de aprovação	Stakeholders
Liberação para Produção	Semana 8	Sistema implementado	Equipe de desenvolvimento
Encerramento do Projeto	Semana 9	Relatório final do projeto	Gerente de projeto

#### 2.3.2 Casos de testes

##### Criação de Tarefas

Ao criar uma tarefa, é importante verificar que o sistema permite a criação com dados válidos, como título e descrição, e exibe corretamente a nova tarefa na lista. Testes devem validar cenários de erro, como a ausência de título, títulos muito longos que excedem o limite permitido ou a tentativa de criar uma tarefa com todos os campos vazios. Nesses

casos, o sistema deve exibir mensagens de erro claras, informando os campos obrigatórios ou limites excedidos.

#### Edição de Tarefas

A funcionalidade de edição deve permitir atualizar o título, descrição e status de uma tarefa. Por exemplo, marcar uma tarefa como concluída deve refletir a mudança na interface. Testes devem garantir que alterações inválidas, como um título vazio, sejam rejeitadas com mensagens de erro adequadas.

#### Exclusão de Tarefas

Excluir uma tarefa existente deve removê-la da lista, enquanto o cancelamento da exclusão deve manter a tarefa inalterada. Além disso, tentar excluir uma tarefa que já foi removida por outro usuário deve gerar uma mensagem informando que a tarefa não existe mais.

#### Visualização de Tarefas

A lista de tarefas deve exibir corretamente todas as tarefas criadas. Deve ser possível filtrar para visualizar apenas tarefas concluídas ou pendentes e aplicar ordenações, como por prioridade ou data, garantindo que os resultados sejam exibidos conforme o esperado.

#### Busca e Filtros

A busca por palavras-chave no título deve retornar apenas as tarefas correspondentes. Filtros, como exibir tarefas pendentes ou concluídas, devem funcionar conforme configurado. Quando nenhuma tarefa é encontrada, o sistema deve exibir uma mensagem clara informando que não há resultados para a busca.

#### Notificações e Lembretes

O sistema deve notificar o usuário sobre tarefas atrasadas e permitir configurar lembretes para tarefas futuras. Esses lembretes devem ser exibidos no momento configurado, garantindo que o usuário seja informado a tempo.

#### Concurrency (Multiusuário)

Quando múltiplos usuários editam a mesma tarefa simultaneamente, o sistema deve resolver conflitos adequadamente para evitar inconsistências. Além disso, alterações feitas por um usuário devem ser refletidas em tempo real para os demais.

#### Performance

O sistema deve ser capaz de carregar uma lista com centenas ou milhares de tarefas sem atrasos significativos. Da mesma forma, a busca em grandes volumes de dados deve retornar resultados em um tempo aceitável, garantindo boa experiência ao usuário.

### 2.3.3



O roteiro de testes para um gerenciador de tarefas abrange várias funcionalidades-chave para garantir que o sistema funcione corretamente:

#### Preparação do Ambiente:

Verificar se o sistema está instalado e acessível, com login válido ou conta de teste.

#### Criação de Tarefas:

Testar a criação de tarefas com dados válidos e inválidos, garantindo que o sistema exiba mensagens de erro apropriadas.

#### Edição de Tarefas:

Validar a edição de título, descrição e status de tarefas, e testar com dados inválidos.

#### Exclusão de Tarefas:

Garantir que a exclusão de tarefas funcione corretamente, e que o sistema informe ao tentar excluir tarefas inexistentes.

#### Visualização de Tarefas:

Verificar a exibição de tarefas com filtros e ordenação por status e data.

#### Busca e Filtros:

Testar a busca por título e aplicação de filtros, incluindo a busca sem resultados.

#### Notificações e Lembretes:

Validar a configuração e exibição de lembretes e notificações de atraso de tarefas.

#### Concurrency (Multiusuário):

Testar a edição simultânea de tarefas por múltiplos usuários e a atualização em tempo real.

#### Performance:

Avaliar o carregamento de listas com grandes volumes de dados e a busca eficiente.

#### Finalização e Relatório:

Documentar os resultados dos testes, falhas encontradas e sugerir correções.

### 3. Gestão de configuração de software.

A Gestão de Configuração de Software (GCS) organiza e controla mudanças no software, garantindo qualidade, rastreabilidade e estabilidade durante o ciclo de vida do projeto.

Principais Atividades: Identificação de Configuração, Controle de Versões e Controle de Mudanças.

Ferramentas: Git e Github

Benefícios: Melhor rastreabilidade e controle, Redução de erros e conflitos, Suporte a processos ágeis (CI/CD) e Reversão rápida de alterações problemáticas.

4. Repositório de gestão de configuração de software.

O repositório está hospedado no GitHub, com o código a documentação e os slides

[A3---Gestao-e-qualidade-de-software/ at main · quitxc/A3---Gestao-e-qualidade-de-software](#)