

UNIVERSIDADE SÃO JUDAS TADEU
CIÊNCIA DA COMPUTAÇÃO

GABRIEL FORNICOLA AMORIM - 824148690
GIOVANNI RIBEIRO IANNACE - 82421986
GIOVANNA FONTES DA SILVA – 823148980
LUCAS GASPARETTO NARCIZO DE MORAIS - 82426494
RAPHAEL MIGUEL FOLEGO - 822163593

WishList
Gestão e Qualidade de Software

São Paulo
2025

1. Planejamento de Testes de Software

O planejamento de testes é uma etapa essencial no ciclo de vida de desenvolvimento do sistema WishList. Ele tem como objetivo garantir que todas as funcionalidades críticas do sistema sejam validadas antes da entrega final, promovendo confiabilidade, robustez e aderência aos requisitos definidos.

1.1 Cronograma de Atividades

O cronograma foi elaborado considerando a ordem lógica do desenvolvimento e a necessidade de validação em ciclos contínuos. As atividades foram distribuídas da seguinte forma:

- Início do desenvolvimento: 29/04
- Implementação das funcionalidades principais: 29/04 a 19/05
- Testes unitários (caixa branca): 13/05 a 23/05
- Testes de API (contrato e resposta): 20/05 a 25/05
- Validação final com usuário fictício: 25/05
- Finalização da documentação: 27/05 a 30/05
- Entrega final: 31/05

Esse planejamento permitiu integração contínua entre desenvolvimento e testes, facilitando a detecção antecipada de falhas.

1.2 Alocação de Recursos

A equipe foi dividida conforme especialidades:

- 1 responsável por front-end
- 1 responsável por back-end
- 2 responsáveis por qualidade e testes (QA)
- 1 responsável pela documentação e organização do projeto

Técnicos:

- Ambiente de desenvolvimento: VS Code, Node.js, React
- Ambiente de testes: Jest (testes unitários), Postman (testes de API), Browser DevTools (testes exploratórios)
- Ferramentas de apoio: Trello (organização), GitHub (versionamento)

1.3 Marcos do Projeto

Os principais marcos definidos para o acompanhamento do progresso do projeto foram:

- MVP funcional concluído: 20/05
- Conclusão dos testes unitários: 23/05
- Validação final com usuário: 25/05
- Documentação fechada e entregue: 30/05
- Apresentação e entrega final: 31/05

2. Documentos de Desenvolvimento de Software

2.1. Plano de Projeto

2.1.1. Planejamento do Projeto

Adotamos a metodologia ágil Scrum, com foco em entregas incrementais e reuniões semanais de acompanhamento. O Trello é utilizado como ferramenta de gestão visual das tarefas, promovendo organização, priorização e visibilidade do progresso de cada etapa do projeto.

- Metodologia Ágil (Scrum)
- Reuniões semanais
- Uso do Trello para gerenciar tarefas

2.1.2. Escopo

O sistema WishList tem como objetivo permitir a gestão de tarefas de forma prática e eficiente. Entre as funcionalidades estão:

- Cadastro de tarefas com descrição e data de execução;
- Edição e exclusão de tarefas;
- Envio automatizado de notificações por e-mail ou via app;
- Interface web responsiva adaptável a dispositivos móveis.
- Sistema de categorias ou etiquetas (tags), para organização personalizada das tarefas;
- Possibilidade de anexar arquivos ou links a cada tarefa, para enriquecer o contexto da atividade;
- Funcionalidade de busca e filtros para facilitar a localização de tarefas específicas;

- Histórico de tarefas concluídas, permitindo o acompanhamento do progresso do usuário ao longo do tempo;
- Painel de resumo com visão geral das tarefas pendentes, em andamento e concluídas;

2.1.3. Recursos

- 4 desenvolvedores
- Hospedagem gratuita no Vercel

2.1.4. Estimativas de Projeto

Com base no planejamento das atividades, estimamos o tempo necessário para cada fase do projeto:

- Desenvolvimento: 3 semanas
- Testes: 1 semana
- Documentação: 1 semana

2.2. Documento de Requisitos

Funcionais:

- RF01: O sistema deve permitir que o usuário cadastre uma tarefa com descrição e data
- RF02: O sistema deve permitir que o usuário edite ou apague uma tarefa existente.
- RF03: O sistema deve enviar uma notificação próxima à data da tarefa
- RF04: O sistema deve permitir o cadastro de categorias ou etiquetas (tags) para organizar as tarefas.
- RF05: O sistema deve disponibilizar um mecanismo de busca e filtros para localização de tarefas específicas.

- RF06: O sistema deve exibir um painel resumo com o número de tarefas pendentes, concluídas e vencidas.
- RF07: O sistema deve manter um histórico de tarefas concluídas.
- RF08: O sistema deve permitir anexar arquivos (ex.: documentos, imagens) a uma tarefa.

Não Funcionais:

- RNF01: O sistema deve possuir interface responsiva (compatível com dispositivos mobile e desktop).
- RNF02: As notificações devem ser entregues no máximo em 5 minutos após a hora programada.
- RNF03: O sistema deve garantir disponibilidade mínima de 99% do tempo.
- RNF04: O sistema deve suportar pelo menos 500 usuários simultâneos sem degradação perceptível de performance.
- RNF05: As informações dos usuários e tarefas devem ser armazenadas de forma segura, com criptografia de dados sensíveis.
- RNF06: O sistema deve seguir as normas de acessibilidade WCAG 2.1, garantindo acesso para pessoas com deficiência.
- RNF07: O tempo de resposta das ações (cadastro, edição, exclusão de tarefas) não deve exceder 2 segundos em 95% das operações.
- RNF08: O sistema deve ser compatível com os principais navegadores (Chrome, Firefox, Edge e Safari).

2.3. Planejamento de Testes

2.3.1. Plano de Testes

2.3.1.1. Introdução

O objetivo dos testes é assegurar que todas as funcionalidades críticas do sistema WishList estejam operando conforme os requisitos estabelecidos. Os testes abrangem desde unidades isoladas até fluxos completos do usuário.

- Verificar o correto funcionamento das funcionalidades da aplicação “WishList”

2.3.1.2. Escopo

- Cobrir todos os requisitos funcionais e principais fluxos

2.3.1.3. Objetivos

- Validar o correto funcionamento das funcionalidades essenciais;
- Identificar e corrigir falhas antes da entrega final;
- Aumentar a confiança dos usuários no sistema.
- Garantir que o sistema atenda aos requisitos especificados

2.3.1.4. Requisitos a Serem Testados

- RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF08

2.3.1.5. Estratégias, Tipos de Testes e Ferramentas

- Testes manuais (funcionais) e unitários (Jest)
- Testes de API com Postman

2.3.1.6. Recursos

- Equipe de testes, ambiente de homologação

2.3.1.7. Cronograma das Atividades

Teste	Responsável	Data
Cadastro de tarefas	Giovanna e Giovanni	22/05
Edição e exclusão	Raphael e Gabriel	23/05
Notificações	Lucas	24/05

2.3.1.8. Marcos

- Conclusão dos testes unitários: 23/05
- Validação final com usuário fictício: 25/05

2.3.2 Casos de Testes

2.3.2.1 Objetivo

Esta fase visa garantir que todas as funcionalidades do sistema *Wish List* estejam de acordo com os requisitos especificados, testando os fluxos principais e alternativos, bem como validando regras de negócio, usabilidade, desempenho e segurança.

2.3.2.2 Estratégia de testes

- Tipo de Testes: Funcionais, Não Funcionais, Caixa preta e Caixa branca
- Ferramentas utilizadas: xUnit, MSTest (para testes automatizados), navegador web (para testes manuais), Postman (para API), Figma (validação visual de UI).
- Critérios de aceitação: Cada funcionalidade deve estar de acordo com o requisito correspondente

2.3.2.3 Funcionais

ID	Funcionalidade	Cenário de Teste	Entrada	Resultado Esperado	Status
CT01	Cadastro de usuários	Cadastro com dados válidos	Nome, e-mail, senha válida	Usuário cadastrado com sucesso	Validado
CT02	Validação de senha	Senha sem número ou símbolo	“senhafraca”	Erro de validação exibido	Validado
CT03	Login com credenciais válidas	Acesso com e-mail e senha corretos	email: teste@email.com senha: admin#123	Redirecionamento para tela de desejos	Em desenvolvimento
CT04	Adição de desejo	Inserção de título, descrição e data	“Viagem”; “Viagem para China do trampo” e “10/11/2025”	Desejo criado e listado	Validado
CT05	Edição de desejo	Modificação de um desejo existente	“Viagem atualizada”; “Viagem com destino e data atualizados: Canadá” e “17/11/2025”	Atualização concluída no desejo	Validado

CT06	Exclusão de desejo	Exclusão e confirmação de um desejo existente	Excluir e confirmar exclusão	Desejo removido	Em testes
CT07	Listagem de desejo	Visualização de desejos existentes	Página inicial	Visualização dos desejos pela página inicial do usuário	Implementado
CT08	Notificação de desejo	Notificação sobre expirar algum desejo	Desejo com a data atual	Envio de e-mail ou alerta	Não iniciado
CT09	Ordenação	Ordenar desejos	Ordenar desejos por data	Lista de desejos ordenada por data	Implementado
CT10	Filtro	Filtrar desejos por título	Inserir título de desejo existente	Exibir desejo filtrado	Em desenvolvimento

2.3.2.4 Não Funcionais

ID	Requisito	Teste	Resultado Esperado	Status
CNF01	Responsividade	Acessar em diferentes dispositivos	Layout adaptável em celular, tablet e desktop	Em testes
CNF02	Desempenho	Carregamento de lista em até 2 segundos	Tempo de resposta < 2s	Em desenvolvimento
CNF03	Segurança (criptografia)	Cadastro de usuário com senha criptografado	Dados criptografados no banco	Validado
CNF04	Compatibilidade	Testar em diferentes navegadores	Interface funcional em todos os navegadores testados	Validado

2.3.2.5 Caixa preta

ID	Cenário	Entrada	Resultado Esperado
CP01	Login válido	E-mail e senha corretos	Redirecionamento para tela inicial
CP02	Login inválido	Senha incorreta	Mensagem de erro exibida
CP03	Cadastro inválido	Senha sem número ou símbolo	Bloqueio com mensagem de erro
CP04	Criação de desejo	Título, descrição e data válidos	Desejo adicionado à lista
CP05	Exclusão de desejo	Confirmação de exclusão	Item removido da lista
CP06	Duplicação de desejo	Título e data já existentes	Mensagem de erro impedindo cadastro
CP07	Notificação de lembrete	Data atual igual à data do desejo	Notificação enviada ao usuário

2.3.3 Roteiro de Testes

- Cenário 1: Cadastro de Tarefa

1. Acessar o sistema
2. Clicar em "Nova Tarefa"
3. Preencher descrição e data
4. Confirmar cadastro

- Cenário 2: Edição de Tarefa

1. Selecionar tarefa existente
2. Clicar em "Editar"
3. Alterar descrição ou data
4. Salvar alterações

- Cenário 3: Exclusão de Tarefa

1. Selecionar tarefa existente
2. Clicar em "Excluir"
3. Confirmar exclusão

- Cenário 4: Verificação de Notificação

1. Cadastrar tarefa para 15 minutos após o horário atual
2. Aguardar
3. Confirmar recebimento da notificação

3. Gestão de Configuração de Software

A Gestão de Configuração de Software (SCM – Software Configuration Management) é um processo fundamental para assegurar que o desenvolvimento e a manutenção do sistema ocorram de forma controlada, rastreável e padronizada. No projeto WishList, a gestão de configuração foi aplicada para:

- Identificação de Itens de Configuração (SCIs): Código-fonte, documentação, casos de teste e artefatos relacionados foram tratados como SCIs, devidamente organizados em diretórios específicos dentro do repositório.
- Controle de Versão: Utilizou-se o sistema Git, hospedado no GitHub, com uma política clara de branches:
 - omain: versão estável e homologada;
 - dev: ambiente de integração contínua;
 - feature/*: novas funcionalidades em desenvolvimento.
- Controle de Alterações: Toda modificação foi realizada via pull requests, obrigatoriamente revisadas por pelo menos um integrante da equipe. Essa prática visou evitar falhas e manter a rastreabilidade das mudanças.
- Auditoria e Rastreabilidade: Utilizou-se o histórico do Git para acompanhar autoria, datas e justificativas das alterações. Cada versão relevante foi marcada com tags, assegurando o versionamento formal.

- Comunicação e Gestão de Impacto: Para cada alteração, foram avaliados seus impactos em outras partes do sistema. Esse controle mitigou o risco de regressões. O Trello foi utilizado como ferramenta de apoio para rastrear tarefas e mudanças relacionadas.
- Relatórios e Documentação: Foram documentadas as versões principais, alterações relevantes e problemas encontrados no changelog do repositório, favorecendo a transparência e a revisão contínua do sistema.
- Essa abordagem garante que o projeto WishList mantenha qualidade, rastreabilidade e controle sobre todas as fases de desenvolvimento, em consonância com as práticas recomendadas nos frameworks de qualidade e engenharia de software.

4. Repositório de Gestão de Configuração

- Link: <https://github.com/RaphaFolego/WishList>
- Estrutura:
 - /src: Código-fonte
 - /docs: Documentação
 - /tests: Casos de teste
 - /notebooks: Rascunhos e pesquisas

Referências Bibliográficas:

- SCRUM. Guia do Scrum: O Guia Definitivo do Scrum – As Regras do Jogo. 2020. Disponível em: <<https://scrumguides.org/>>. Acesso em: 10 jun. 2025.
- POSTMAN. Postman Learning Center – Introduction to Postman. 2025. Disponível em: <<https://learning.postman.com/>>. Acesso em: 10 jun. 2025.

- JEST. Jest – Delightful JavaScript Testing. Meta Open Source, 2025. Disponível em: <<https://jestjs.io/docs/getting-started>>. Acesso em: 10 jun. 2025.
- GITHUB DOCS. About version control with Git. GitHub, 2025. Disponível em: <<https://docs.github.com/en/get-started/using-git>>. Acesso em: 10 jun. 2025.
- TRELLO. Como usar o Trello. Atlassian, 2025. Disponível em: <<https://trello.com/guide>>. Acesso em: 10 jun. 2025.