Renan Seiki de Almeida

RA: 823117418

Renan Assensio Barbosa

RA: 82318879

Willians Gabriel Gomes Pereira

RA: 823133377

Caique Coelho de Oliveira Ramos

RA: 82312240

Caio Ryan Prado Sobral

RA: 825112919

Uniplano

SÃO PAULO

2025

Sumário

Documento de Software - Uniplano

1. Planejamento de Testes de Software

Esta seção complementa o detalhamento do Plano de Testes, focando em aspectos práticos de tempo, pessoas e entregas.

1.1. Cronograma de Atividades

O cronograma a seguir detalha as fases e as datas estimadas para as principais atividades do processo de testes do Uniplano.

Atividade	Data de Início	Data de Fim	Duração (Dias Úteis)	Responsável(is)
Planejamento de Testes	07/07/2025	11/07/2025	5	Líder de Testes
Criação de Casos de Teste	14/07/2025	17/07/2025	4	Testadores
Preparação do Ambiente	16/07/2025	18/07/2025	3	Equipe de Infra
Execução de Testes Unitários	21/07/2025	22/07/2025	2	Desenvolvedores
Execução de Testes de Integração	23/07/2025	24/07/2025	2	Testadores
Execução de Testes de Sistema	25/07/2025	29/07/2025	3	Testadores

Testes de Usabilidade	25/07/2025	28/07/2025	2	Testadores
Testes de Desempenho	29/07/2025	29/07/2025	1	Testadores
Análise de Resultados e Relatórios	30/07/2025	31/07/2025	2	Testadores, Líder de Testes
Correções de Defeitos (Fase 1)	30/07/2025	05/08/2025	4	Desenvolvedores
Re-testes e Validação (Fase 1)	06/08/2025	08/08/2025	3	Testadores
Homologação Final	09/08/2025	10/08/2025	2	Stakeholders, Líder de Testes

1.2. Alocação de Recursos

A alocação de recursos é crucial para o sucesso das atividades de teste. Os seguintes recursos serão empregados:

- **Equipe:**
  - **1 Líder de Testes:** Responsável pelo planejamento, supervisão e garantia da qualidade dos testes.
  - **2 Testadores:** Responsáveis pela criação, execução e documentação dos casos de teste.
  - **Equipe de Desenvolvimento:** Responsável pela correção de defeitos e suporte técnico.
  - **Equipe de Infraestrutura (se aplicável):** Responsável pela configuração e manutenção dos ambientes de teste.
- **Ambiente de Testes:**
  - **Ambiente de Desenvolvimento:** Máquinas locais dos desenvolvedores para testes unitários e de integração iniciais.
  - **Ambiente de Homologação/Teste:** Um ambiente dedicado que replica, o mais próximo possível, o ambiente de produção. Inclui máquinas com

Node.js configurado para o front-end e um servidor de banco de dados PostgreSQL.

- **Ferramentas:**
  - **Gerenciamento de Testes:** Planilhas ou uma ferramenta de gerenciamento de casos de teste (ex: TestLink, Jira com plugin).
  - **Automação de Testes Front-end:** Cypress.
  - **Teste de API:** Postman.
  - **Teste de Desempenho:** JMeter.
  - **Controle de Versão:** Git/GitHub.
  - **Análise de Código:** SonarQube.

1.3. Marcos do Projeto

Os marcos do projeto são pontos de controle importantes que indicam o progresso e a conclusão de fases críticas.

- **11/07/2025: Plano de Testes Validado.** Aprovação formal do documento de planejamento de testes.
- **17/07/2025: Casos de Teste Finalizados.** Conclusão da escrita e revisão de todos os casos de teste.
- **25/07/2025: Testes de Sistema Iniciados.** Início formal da fase de testes mais abrangente, envolvendo as funcionalidades principais.
- **29/07/2025: Execução de Testes Principais Concluída.** Finalização da execução dos testes de sistema, usabilidade e desempenho.
- **10/08/2025: Correções Finais e Homologação.** Todos os defeitos críticos corrigidos e validação final pelo time de testes e stakeholders.

---

2. Documentos de Desenvolvimento de Software

Esta seção aborda o planejamento geral do projeto, crucial antes mesmo do início do desenvolvimento e testes.

2.1. Plano de Projeto

O Plano de Projeto do Uniplano delineia a estratégia geral para a execução, controle e fechamento do projeto, garantindo que os objetivos sejam alcançados de forma eficiente.

2.1.1. Planejamento do Projeto

O planejamento do projeto Uniplano envolveu as seguintes etapas:

- **Definição do Problema:** Identificação da necessidade de um sistema de gestão de projetos e tarefas simplificado para estudantes universitários.
- **Análise de Viabilidade:** Avaliação técnica, operacional e econômica para garantir que o projeto é realizável.
- **Levantamento de Requisitos:** Coleta e documentação detalhada das necessidades dos usuários e do sistema (conforme seção 2.2. Documento de Requisitos).
- **Definição da Arquitetura:** Escolha das tecnologias e estrutura do sistema (ex: front-end em React, back-end em Node.js, banco de dados PostgreSQL).

- **Decomposição do Trabalho (WBS):** Quebra do projeto em entregas menores e gerenciáveis (ex: módulo de autenticação, módulo de projetos, módulo de tarefas).
- **Estimativa de Esforço e Cronograma:** Definição das horas de trabalho necessárias e prazos para cada atividade (detalhado na seção 2.1.4).
- **Identificação de Riscos:** Análise de potenciais problemas e planos de mitigação (ex: atrasos, problemas técnicos).
- **Plano de Qualidade:** Definição das atividades de garantia de qualidade, incluindo o plano de testes (seção 2.3).

2.1.2. Escopo

O escopo do projeto Uniplano é construir uma **aplicação web para gerenciamento de projetos e tarefas**, com foco em **login de usuário, criação e organização de projetos e tarefas, e visualização por prazos**.

O que está dentro do escopo:

- Módulo de Autenticação (cadastro, login, logout).
- Módulo de Gerenciamento de Projetos (criação, edição, exclusão, visualização).
- Módulo de Gerenciamento de Tarefas (criação, edição, exclusão, marcação de status, associação a projetos).
- Funcionalidade de organização e visualização por prazos.
- Interface de usuário responsiva para navegadores web.

O que está fora do escopo inicial (e pode ser considerado para fases futuras):

- Notificações por e-mail ou push.
- Integrações com calendários externos (ex: Google Calendar, Outlook Calendar).
- Compartilhamento de projetos entre múltiplos usuários.
- Relatórios avançados e gráficos.
- Aplicativos móveis nativos.

2.1.3. Recursos

Os recursos necessários para o desenvolvimento do Uniplano incluem:

- **Equipe do Projeto:**
    - **1 Gerente de Projeto:** Responsável pela coordenação geral, comunicação e gestão de riscos.
    - **3 Desenvolvedores Full-stack:** Responsáveis pelo desenvolvimento do front-end e back-end.
    - **1 Designer UI/UX (se aplicável):** Responsável pela criação da interface e experiência do usuário.
    - **3 Testadores (conforme seção 1.2):** Responsáveis pela garantia da qualidade.
- **Tecnologias e Ferramentas:**
    - **Linguagens de Programação:** JavaScript/TypeScript.
    - **Frameworks/Bibliotecas:** React (front-end), Node.js (back-end).
    - **Banco de Dados:** PostgreSQL.
    - **Controle de Versão:** Git.

    - **Plataforma de Colaboração:** GitHub.
    - **Ferramentas de Design:** Figma (se aplicável).
    - **Ambientes de Desenvolvimento:** IDEs como VS Code.
- **Infraestrutura:**
    - Servidor para ambiente de desenvolvimento/homologação (máquinas locais).
    - Servidor para ambiente de produção (serviço de cloud, ex: AWS, Heroku).

2.1.4. Estimativas de Projeto

As estimativas a seguir são baseadas na complexidade das funcionalidades e na experiência da equipe, apresentadas em semanas de trabalho

Fase do Projeto	**Esforço Estimado (Semanas)**	**Data de Início**	**Data de Fim**
Iniciação e Planejamento	2	03/06/2025	13/06/2025
Levantamento de Requisitos	1	09/06/2025	13/06/2025
Design e Arquitetura	2	16/06/2025	27/06/2025
Desenvolvimento			
Módulo de Autenticação	2	30/06/2025	11/07/2025
Módulo de Projetos	2	30/06/2025	11/07/2025
Módulo de Tarefas	2	14/07/2025	25/07/2025
Organização por Prazos	1	21/07/2025	25/07/2025

Testes e Validação	1	28/07/2025	01/08/2025
Implantação	1	04/08/2025	08/08/2025
Pós-Implantação (Suporte)	Contínuo	11/08/2025	-

Total estimado do projeto (sem contar suporte contínuo): aproximadamente 12 semanas de desenvolvimento + testes + implantação.

Observações:

- As estimativas são aproximadas e podem sofrer ajustes conforme o andamento do projeto e a descoberta de novas complexidades.
- As datas de início e fim podem se sobrepor em alguns casos, indicando atividades paralelas.
- Será utilizada uma abordagem ágil, com sprints curtas, para permitir flexibilidade e feedback contínuo.

2.2. Documento de Requisitos

Campo de definições dos requisitos funcionais e não funcionais do projeto Uniplano.

2.2.1. Requisitos Funcionais (RF)

- **RF001 - Autenticação de Usuário:** O sistema deve permitir que usuários se autentiquem usando credenciais válidas (usuário e senha).
    - **Descrição:** O usuário deve ser capaz de informar seu nome de usuário/e-mail e senha para acessar o sistema.
    - **Prioridade:** Alta
    - **Observações:** Deve incluir validação de credenciais inválidas e recuperação de senha.
- **RF002 - Cadastro de Projetos:** O sistema deve permitir que usuários autenticados criem, editem e excluam projetos.
    - **Descrição:** Usuários podem definir nome, descrição e prazo para cada projeto.
    - **Prioridade:** Alta
    - **Observações:** Projetos devem ser associados ao usuário que os criou.
- **RF003 - Gerenciamento de Tarefas:** O sistema deve permitir que usuários atribuam, atualizem e concluam tarefas dentro dos projetos.
    - **Descrição:** Cada tarefa deve ter nome, descrição, prazo e status (pendente, em andamento, concluída).
    - **Prioridade:** Alta
    - **Observações:** Tarefas podem ser vinculadas a um projeto específico.

- **RF004 - Organização por Prazos:** O sistema deve permitir visualizar projetos e tarefas organizados por prazos.
    - **Descrição:** O usuário deve poder filtrar e ordenar projetos e tarefas com base em suas datas de entrega.
    - **Prioridade:** Média
    - **Observações:** Pode incluir visualizações como calendário ou lista de tarefas.

2.2.2. Requisitos Não Funcionais (RNF)

- **RNF001 - Usabilidade:** O sistema deve ser intuitivo e de fácil utilização para o público-alvo.
    - **Descrição:** A interface do usuário deve ser clara, consistente e responsiva, facilitando a navegação e a realização das tarefas.
    - **Prioridade:** Alta
    - **Observações:** Avaliar a usabilidade através de testes de usabilidade e feedback de usuários.
- **RNF002 - Desempenho:** O sistema deve apresentar um tempo de resposta adequado para as operações principais.
    - **Descrição:** O tempo máximo para o login, cadastro de projetos e tarefas não deve exceder 3 segundos sob condições normais de uso.
    - **Prioridade:** Alta
    - **Observações:** Testes de carga e estresse serão utilizados para validar este requisito.
- **RNF003 - Segurança:** O sistema deve garantir a segurança dos dados dos usuários e das informações dos projetos.
    - **Descrição:** As senhas dos usuários devem ser armazenadas de forma criptografada. Acesso não autorizado deve ser impedido.
    - **Prioridade:** Alta
    - **Observações:** Uso de HTTPS, autenticação e autorização adequadas.
- **RNF004 - Compatibilidade:** O sistema deve ser compatível com os principais navegadores web (Chrome, Firefox, Edge).
    - **Descrição:** A aplicação web deve funcionar corretamente em diferentes navegadores sem perda de funcionalidade ou problemas de layout.
    - **Prioridade:** Média
    - **Observações:** Testes de compatibilidade em diferentes versões de navegadores.

2.3. Planejamento de Testes

2.3.1. Plano de Testes

Plano de testes para evitar o máximo de bugs antes da homologação.

2.3.1.1. Introdução

Este plano de testes tem como objetivo definir o escopo, os objetivos, os requisitos a serem testados, as estratégias e ferramentas utilizadas no processo de testes do sistema Uniplano. O público-alvo deste documento são os desenvolvedores, testadores e stakeholders envolvidos no projeto, garantindo que todos compreendam o processo de garantia de qualidade.

2.3.1.2. Escopo

Serão testadas as **funcionalidades principais do sistema Uniplano**, incluindo:

- **Login de usuário**
- **Cadastro e gerenciamento de projetos**
- **Cadastro e organização de tarefas**
- **Organização de projetos e tarefas por prazos**

Funcionalidades externas, como notificações por e-mail e integração com Google Calendar, **não serão testadas nesta fase** do projeto.

2.3.1.3. Objetivos

Os principais objetivos deste plano de testes são:

- **Garantir que as funcionalidades principais estejam implementadas corretamente** e sem defeitos críticos.
- **Validar a usabilidade do sistema**, assegurando que o Uniplano seja intuitivo e fácil de usar para o público-alvo.
- **Assegurar que o sistema apresente desempenho adequado** nas operações essenciais, conforme os requisitos não funcionais.
- **Identificar e reportar defeitos** de forma clara e objetiva para que sejam corrigidos antes da implantação.

2.3.1.4. Requisitos a Serem Testados

Os requisitos que serão testados neste plano incluem:

- **Requisitos Funcionais (RF):** RF001 (Autenticação), RF002 (Cadastro de Projetos), RF003 (Gerenciamento de Tarefas) e RF004 (Organização por Prazos), conforme detalhados na seção 2.2.1.
- **Requisitos Não Funcionais (RNF):** RNF001 (Usabilidade) e RNF002 (Desempenho), conforme detalhados na seção 2.2.2.

A rastreabilidade entre os casos de teste e esses requisitos será mantida para garantir a cobertura completa.

2.3.1.5. Estratégias, Tipos de Testes e Ferramentas a Serem Utilizadas

- **Estratégia de Teste:** Será adotada uma **estratégia de testes de caixa-preta**, focando nas funcionalidades e comportamento do sistema do ponto de vista do usuário final, sem a necessidade de conhecimento interno do código.
- **Tipos de Testes:**
    - **Testes Unitários:** Serão realizados para verificar as menores unidades de código de forma isolada (geralmente pelos próprios desenvolvedores).

    - **Testes de Integração:** Destinados a verificar a comunicação e interação entre diferentes módulos e componentes do sistema.
    - **Testes de Sistema:** Abrangem o sistema como um todo, validando o comportamento completo da aplicação em relação aos requisitos.
    - **Testes de Usabilidade:** Focarão na experiência do usuário, avaliando a facilidade de uso e a intuitividade da interface.
    - **Testes de Desempenho:** Medirão a responsividade e estabilidade do sistema sob diferentes cargas de trabalho.
- **Ferramentas a Serem Utilizadas:**
    - **Cypress:** Para testes de interface (front-end) e fluxos de usuário end-to-end.
    - **Postman:** Para testes de APIs (back-end), validando as requisições e respostas do servidor.
    - **JMeter:** Para testes de desempenho e carga, simulando múltiplos acessos simultâneos.
    - **SonarQube:** Para análise estática de qualidade do código, embora não seja uma ferramenta de execução de testes, contribui para a qualidade geral.

2.3.1.6. Recursos a Serem Empregados

- **Equipe de Testes:** Composta por **3 testadores** dedicados, responsáveis pela criação, execução e relatório de testes.
- **Ambiente de Testes:** Serão utilizadas **máquinas locais com ambiente Node.js configurado** para o front-end e **banco de dados PostgreSQL** para o back-end, replicando o ambiente de produção.
- **Ferramentas:** As ferramentas listadas na seção anterior (Cypress, Postman, JMeter) estarão configuradas e disponíveis para a equipe.

2.3.1.7. Cronograma das Atividades

Atividade	Data de Início	Data de Fim
Planejamento de Testes	07/07/2025	11/07/2025
Criação de Casos de Teste	14/07/2025	17/07/2025
Execução de Testes	21/07/2025	29/07/2025
Correções e Validação	30/07/2025	08/08/2025

2.3.1.8. Definição dos Marcos do Projeto

Marco do Projeto	Data Limite
Plano de Testes Validado	11/07/2025
Casos de Teste Finalizados	17/07/2025
Testes de Sistema Iniciados	25/07/2025
Correções Finais e Homologação	08/08/2025

2.3.2. Casos de Testes

Os casos de teste detalham os passos para verificar funcionalidades específicas.

- **CT001: Login Válido**
    - **Pré-condições:** Usuário cadastrado e com credenciais válidas.
    - **Passos:**
        1. Acessar a tela de login do Uniplano.
        2. Inserir um e-mail válido no campo "Usuário".
        3. Inserir uma senha válida no campo "Senha".
        4. Clicar no botão "Entrar".
    - **Resultado Esperado:** O usuário deve ser redirecionado para a dashboard principal do sistema, exibindo uma mensagem de "Bem-vindo" ou similar.
    - **Status:** Pass (no momento da redação)
    - **Requisito(s) Coberto(s):** RF001
- **CT002: Cadastro de Projeto**
    - **Pré-condições:** Usuário autenticado e com permissão para criar projetos.
    - **Passos:**
        1. Acessar a tela de "Projetos" ou "Novo Projeto".
        2. Preencher o campo "Nome do Projeto" com um nome válido (ex: "Projeto Faculdade").
        3. Preencher o campo "Descrição" com detalhes do projeto (ex: "Desenvolvimento do sistema Uniplano").
        4. Selecionar uma data de prazo no campo "Prazo" (ex: 30/07/2025).
        5. Clicar no botão "Salvar" ou "Criar Projeto".
    - **Resultado Esperado:** O projeto deve ser criado com sucesso, exibindo uma mensagem de confirmação e listado na área de "Meus Projetos".

    - **Status:** Pass (no momento da redação)
    - **Requisito(s) Coberto(s):** RF002
- **CT003: Login Inválido (Cenário de Erro)**

    - **Pré-condições:** Nenhuma.
    - **Passos:**
        1. Acessar a tela de login do Uniplano.
        2. Inserir um e-mail inválido (ex: "usuario@dominio.com") no campo "Usuário".
        3. Inserir uma senha incorreta (ex: "senhaErrada123") no campo "Senha".
        4. Clicar no botão "Entrar".
    - **Resultado Esperado:** O sistema deve exibir uma mensagem de erro clara, indicando que as credenciais são inválidas (ex: "Usuário ou senha incorretos") e o usuário deve permanecer na tela de login.
    - **Status:** (A ser testado)
    - **Requisito(s) Coberto(s):** RF001

2.3.3. Roteiro de Testes

O roteiro de testes a seguir define as principais áreas e sequências de testes a serem executados no sistema Uniplano para garantir a cobertura dos requisitos e a qualidade geral da aplicação.

1. **Testes de Autenticação:**
    - Login com credenciais válidas.
    - Login com credenciais inválidas.
    - Tentativas de acesso sem credenciais.
    - Recuperação de senha (se aplicável e dentro do escopo).
2. **Cadastro e Gerenciamento de Projetos:**
    - Criação de novos projetos com dados válidos e inválidos.
    - Edição de informações de projetos existentes.
    - Exclusão de projetos.
    - Visualização da lista de projetos.
3. **Cadastro e Atribuição de Tarefas:**
    - Criação de tarefas associadas a projetos.
    - Edição de detalhes de tarefas (descrição, status).
    - Conclusão de tarefas.
    - Exclusão de tarefas.
4. **Validação de Prazos e Organização:**
    - Verificação da correta exibição de projetos e tarefas por ordem de prazo.
    - Testes de filtros e ordenação (se implementados).
    - Comportamento do sistema ao atingir ou ultrapassar prazos.
5. **Testes de Integração:**
    - Fluxos completos que envolvem múltiplos módulos (ex: criar projeto, depois criar tarefas para ele).
    - Verificação da comunicação entre front-end e back-end (APIs).

6. **Testes de Desempenho e Carga:**
    - Simulação de múltiplos acessos simultâneos ao sistema.
    - Avaliação do tempo de resposta para operações críticas (login, criação de projeto).
    - Monitoramento do uso de recursos do servidor sob carga.
7. **Testes de Usabilidade:**
    - Navegação geral pelo sistema.
    - Avaliação da clareza da interface e mensagens de erro.
    - Facilidade de aprendizado e uso para novos usuários.

---

3. Gestão de Configuração de Software

A gestão de configuração de software (GCS) é essencial para manter a integridade, rastreabilidade e organização do projeto ao longo do seu ciclo de vida.

3.1. Ferramentas e Metodologias Utilizadas

A gestão de configuração do projeto Uniplano foi realizada com o uso do **Git**, um sistema de controle de versão distribuído. A metodologia adotada foi o **Git Flow**, que organiza o desenvolvimento em branches específicas para funcionalidades, correções de bugs, releases e o branch principal de desenvolvimento.

3.2. Fluxo de Trabalho (Git Flow)

O fluxo de trabalho Git Flow permitiu uma gestão robusta das diferentes fases do desenvolvimento:

- **`main` (ou `master`):** Contém o código de produção, estável e pronto para implantação.
- **`develop`:** Branch principal de desenvolvimento, onde as features são integradas após serem concluídas.
- **`feature branches`:** Criadas a partir de `develop` para o desenvolvimento de novas funcionalidades de forma isolada. Após a conclusão e revisão, são integradas de volta em `develop`.
- **`release branches`:** Criadas a partir de `develop` quando uma nova versão está prestes a ser lançada. Permitem a preparação para o release, correções finais e testes de homologação, antes de serem mescladas em `main` e `develop`.
- **`hotfix branches`:** Criadas a partir de `main` para correções urgentes em produção, sendo mescladas de volta em `main` e `develop`.

3.3. Versionamento e Histórico

Cada funcionalidade, correção ou melhoria foi implementada em **branches específicas**. Após o desenvolvimento e testes locais, essas branches foram revisadas e integradas à branch de desenvolvimento principal (`develop`). Periodicamente, a cada sprint ou conjunto de funcionalidades concluídas, um **release era gerado** e disponibilizado para testes em um ambiente de homologação. Isso garantiu um histórico completo de alterações e a possibilidade de reverter para versões anteriores, se necessário.

4. Repositório de Gestão de Configuração de Software

O repositório principal do projeto Uniplano foi hospedado no **GitHub**, uma plataforma líder para controle de versão e colaboração. O GitHub forneceu:

- **Controle de versionamento completo:** Com histórico detalhado de todas as alterações, autores e datas.
- **Colaboração em equipe:** Facilitação para que múltiplos desenvolvedores trabalhem simultaneamente no código.
- **Gerenciamento de issues:** Para rastrear bugs e tarefas de desenvolvimento.
- **Integração contínua (CI/CD):** Possibilidade de configurar pipelines para automação de testes e implantação.

O link para o repositório do projeto, contendo todo o histórico de commits e branches, está disponível privadamente para avaliação do professor.

---

5. Conclusão

A elaboração deste projeto permitiu a aplicação prática de conceitos de engenharia de software, com foco especial na qualidade e organização dos processos. O uso de documentação estruturada, planejamento de testes e controle de configuração garantiu a entrega de um produto funcional, confiável e bem documentado. A disciplina destacou-se pela ênfase na responsabilidade em grupo e no uso de ferramentas profissionais.

---

6. Referências Bibliográficas

- PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software.** 8ª ed. Bookman, 2016.
- SOMMERVILLE, Ian. **Engenharia de Software.** 9ª ed. Pearson, 2011.
- GONÇALVES, Priscila F. et al. **Testes de software e gerência de configuração.** Soluções Educacionais Integradas, 2019.
- PFLEEGER, Shari Lawrence. **Engenharia de software: teoria e prática.** 2ª ed. Prentice Hall, 2004.
- BRAGA, Pedro Henrique Cacique. **Teste de Software.** São Paulo: Pearson Education, 2016.
- GALLIOTTI, Giocondo MARINO. **Qualidade de Software.** São Paulo: Pearson Education, 2016.
- TONINI, Antonio Carlos. **Métricas de Software.** 2004.
