**PROJETO**

**Aplicativo Comedoria da Tia**

**INTEGRANTES DO PROJETO e RA’S**

Bianca Marcondes Camargo - 25028292

Erik dos Santos Raimundo - 24026628

Larissa de Almeida Lira Oliveira - 24026849

Vinicius Cardoso de Lima - 24026851

São Paulo

2025

Sumário

[1. INTRODUÇÃO 4](#_Toc206698897)

[2. Qualidade de Software 4](#_Toc206698898)

[2.1. Modelo que qualidade de software (Diagrama/Design) 4](#_Toc206698899)

[2.2. Processo (plano) de gerenciamento de qualidade de software (texto explicativo) 4](#_Toc206698900)

[2.3. Identificação de atributos de qualidade da norma 25010. 4](#_Toc206698901)

[2.4. Relatório que explica como a norma de qualidade de software 25010 é utilizada no processo de desenvolvimento. 4](#_Toc206698902)

[3. Teste de Software 4](#_Toc206698903)

[3.1. Plano de Teste 4](#_Toc206698904)

[3.2. Apresentar 2 testes unitários. 4](#_Toc206698905)

[3.3. Apresentar 2 testes de integração 5](#_Toc206698906)

[3.4. Apresentar um teste de usuário (SISTEMA) 5](#_Toc206698907)

[4. REFERÊNCIAS BIBLIOGRÁFICAS 5](#_Toc206698908)

A screenshot of a computer screen

AI-generated content may be incorrect.

# INTRODUÇÃO

Fome e pouco tempo? A tia te entende!

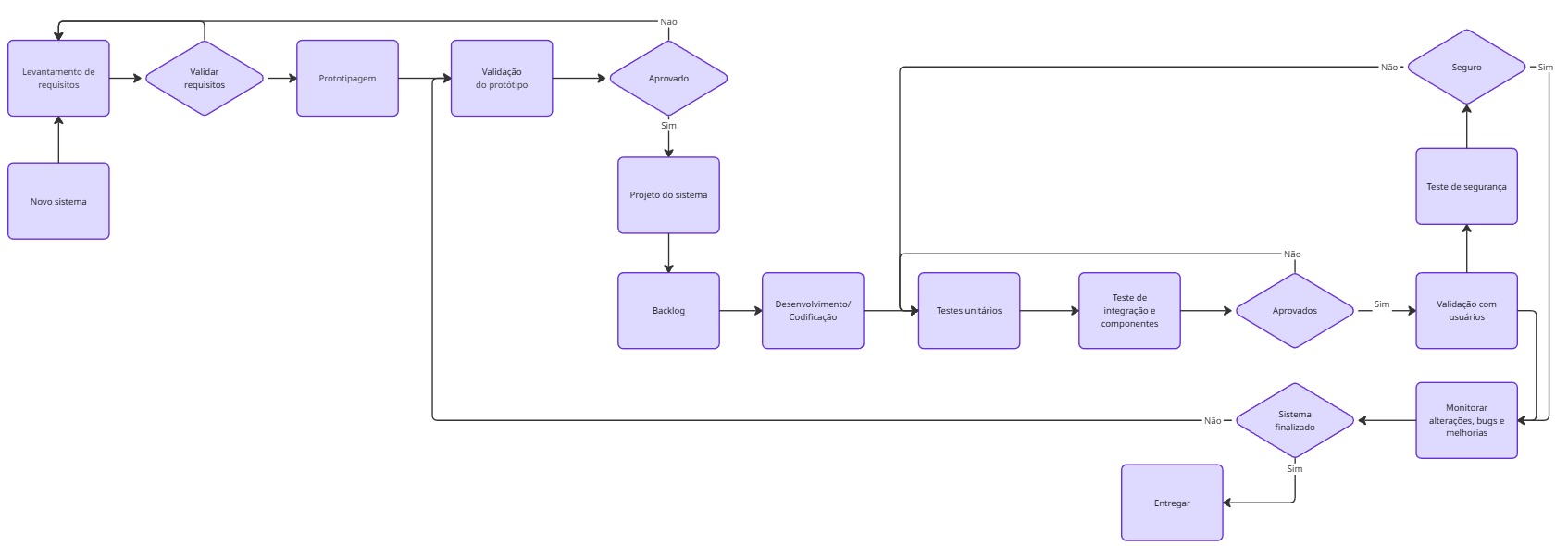
É por isso que o **Comedoria da Tia** foi criado! o seu novo aliado na hora de fazer uma refeição na faculdade. O aplicativo foi pensado para resolver os problemas que vocês já conhecem bem: as filas enormes no horário de pico e a correria de ir e voltar até o refeitório.

Com ele, vocês fazem seu pedido de forma prática e rápida, direto do seu celular, seja da sua sala de aula, da biblioteca ou de onde estiver. É só escolher o que quer comer, pagar pelo app e pronto! A gente avisa quando estiver tudo preparado para você só passar e retirar, **sem precisar enfrentar a fila**.

Assim, você aproveita ao máximo seu intervalo, garantindo uma refeição deliciosa sem estresse. Mais tempo para estudar, relaxar ou conversar com os amigos!

# Qualidade de Software

# Modelo de qualidade de software (Diagrama/Design)



# Processo (plano) de gerenciamento de qualidade de software (texto explicativo)

### **Plano de Gerenciamento de Qualidade de Software**

Com base no diagrama e na norma ISO/IEC 25010, que define as características de qualidade de produto de software, é possível estruturar um plano de gerenciamento de qualidade. O diagrama ilustra um ciclo de vida de desenvolvimento que incorpora etapas de validação e testes, fundamentais para garantir a qualidade do produto.

O processo começa com o Levantamento de Requisitos, seguido pela Validação de Requisitos e Prototipagem. A validação garante que as necessidades do usuário foram compreendidas corretamente, abordando a característica de adequação funcional da ISO 25010.

Se o protótipo for aprovado, o projeto avança para a fase de Projeto do Sistema e a criação do Backlog, que é a lista de funcionalidades a serem desenvolvidas.

A etapa de Desenvolvimento/Codificação é a implementação do projeto. Após a codificação, o processo se concentra em testes para garantir a qualidade.

1. Testes Unitários: Verificam a menor parte do código individualmente, focando na manutenibilidade e eficiência do software.
2. Teste de Integração e Componentes: Avaliam como diferentes módulos do software interagem, assegurando a adequação funcional e a compatibilidade.

Se os testes não forem aprovados, o processo retorna para a etapa de codificação e testes, formando um ciclo de correção e melhoria.

Após a aprovação, o processo foca na qualidade sob a perspectiva do usuário e da segurança.

1. **Teste de Segurança:** É uma etapa crucial para garantir que o software está seguro, protegendo dados e sistemas contra acesso não autorizado. Isso aborda diretamente a característica de segurança da ISO 25010.
2. **Validação com Usuários:** Nesta fase, o software é testado com os usuários finais. Isso garante que o produto atende aos requisitos de usabilidade e adequaçãofuncional, confirmando que o software é eficiente, compreensível e satisfatório para o seu uso.

Finalmente, se o sistema for considerado finalizado, ele é entregue. No entanto, o diagrama sugere um processo contínuo de Monitoramento de alterações, bugs e melhorias, que é essencial para a manutenibilidade do software a longo prazo.

Em resumo, o diagrama demonstra um plano de qualidade que cobre as principais fases do desenvolvimento de software, com foco em:

* **Adequação Funcional:** Garantida pelo levantamento de requisitos, prototipagem e testes de integração.
* **Segurança:** Abordada pela etapa de testes de segurança dedicados.
* **Usabilidade:** Verificada diretamente com os usuários finais.
* **Manutenibilidade:** Assegurada pelos testes unitários e pelo monitoramento contínuo após a entrega.

Este plano alinha-se aos princípios da ISO 25010, que servem como guia para garantir que o software não apenas funcione, mas seja robusto, seguro e satisfaça as expectativas de seus usuários.

# Identificação de atributos de qualidade da norma 25010.

**Usabilidade**

A usabilidade, que se refere à facilidade com que um usuário pode interagir com o sistema, é comprometida. O documento aponta que o acesso ao cardápio é difícil.

**Eficiência de Desempenho**

A eficiência de desempenho, que trata do comportamento do sistema em relação ao uso de recursos e ao tempo de resposta, é o principal problema. As filas grandes são um sintoma direto dessa ineficiência, causada principalmente pelo atraso em pagamentos.

**Adequação Funcional**

A adequação funcional, que se refere à capacidade do software de fornecer as funções necessárias para atender aos requisitos do usuário, apresenta deficiências. A falta de acesso ao estoque impede que o sistema informe sobre produtos em falta.

# Relatório que explica como a norma de qualidade de software 25010 é utilizada no processo de desenvolvimento.

A norma servirá como uma estrutura para garantir que o software atenda às necessidades do refeitório e resolva os problemas identificados, como **filas grandes** e a **distância** das salas de aula.

**Adequação Funcional**

A adequação funcional será o ponto de partida do desenvolvimento, garantindo que o app tenha as funcionalidades necessárias para resolver os problemas. O sistema atual falha em informar sobre **produtos em falta** por **não ter acesso a estoque**. O novo aplicativo resolverá isso ao fornecer uma função que exibe o estoque em tempo real. Além disso, para solucionar o problema da **distância** que leva à **escolha de outras cantinas**, o app oferecerá a capacidade de pré-encomenda e pagamento de qualquer lugar, permitindo que os alunos passem apenas para retirar seu pedido.

**Usabilidade**

A usabilidade se concentrará em criar uma experiência de usuário simples e intuitiva. Os problemas de **acesso difícil ao cardápio**, como o **QR code de difícil acesso**, serão eliminados. O design do aplicativo priorizará uma interface limpa e direta para a navegação do cardápio e para o processo de pagamento. A **separação de ComeDinheiro**, que complica o fluxo de pagamentos, será substituída por um método de pagamento integrado e unificado, facilitando a vida do usuário.

**Eficiência de Desempenho**

Para resolver as **filas grandes** e o **atraso em pagamentos**, a eficiência de desempenho será uma métrica-chave. O app será desenvolvido para otimizar as transações, garantindo que os **pagamentos em cartão** não sejam **lentos** e que o sistema possa lidar com a **alta demanda** em horários de pico. Isso permitirá que o refeitório processe pedidos de forma rápida, sem causar atrasos.

**Confiabilidade**

A confiabilidade do aplicativo será assegurada por meio de testes rigorosos. O problema de o **cardápio não carregar** será eliminado, garantindo que as funcionalidades essenciais do app, como a visualização do cardápio e o processamento de pedidos, operem de forma consistente e sem falhas em qualquer momento.

# Teste de Software

# Plano de Teste

# Apresentar 2 testes unitários.

# Apresentar 2 testes de integração

# Apresentar um teste de usuário (SISTEMA)

# 4. REFERÊNCIAS BIBLIOGRÁFICAS

<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

SOMMERVILLE, I. **Engenharia de Software.** 11ª Edição. São Paulo: Pearson Addison-Wesley, 2017.