

PROJETO MENU-UX**Requisitos da disciplina Testes e Qualidade de Software 2025_02****INTEGRANTES DO PROJETO e RA'S**

Eriane Dias	-	24026778
Kauã Daniel	-	24026846
Vitor Manuel	-	24026801
Nathan Lucena	-	24026563

São Paulo
2025

Sumário

1. INTRODUÇÃO	4
2. Qualidade de Software	4
2.1. Modelo que qualidade de software (Diagrama/Design)	4
2.2. Processo (plano) de gerenciamento de qualidade de software (texto explicativo)	4
2.3. Identificação de atributos de qualidade da norma 25010.	4
2.4. Relatório que explica como a norma de qualidade de software 25010 é utilizada no processo de desenvolvimento.	4
3. Teste de Software	4
3.1. Plano de Teste	4
3.2. Apresentar 2 testes unitários.	4
3.3. Apresentar 2 testes de integração	5
3.4. Apresentar um teste de usuário (SISTEMA)	5
4. REFERÊNCIAS BIBLIOGRÁFICAS	5

Data das entregas	22/set.	10/nov.	17/11 - 25/11
Disciplinas	Entrega 1 (3,0 Pontos)	Entrega 2 (4,0 Pontos)	Apresentação (3,0 Pontos)
Testes de Software	<p>Instruções de Entrega: Qualidade de Software</p> <p>Descrição: O aluno deverá demonstrar o entendimento dos conceitos de qualidade de software, aplicando-os ao projeto. Isso inclui a criação de um diagrama do processo de qualidade de software, a identificação de atributos de qualidade. Apresentar um relatório que explica como a norma de qualidade de software 25010 é utilizada no processo de desenvolvimento. Entrega em documento pdv seguindo o template fornecido pelo professor.</p>	<p>Instruções de Entrega: Testes Unitários, de Integração e de usuário.</p> <p>Descrição: Os alunos deverão desenvolver 4 testes unitários e 2 testes de integração e 1 teste de usuário. Esses testes devem garantir que o sistema esteja funcionando corretamente e validando os principais componentes e integrações e a geração de um relatório de teste de carga do servidor.</p> <p>Entrega em documento pdv seguindo o template fornecido pelo professor.</p>	<p>Critérios: Criatividade, Impacto Social,Tempo de apresentação e Embasamento.</p>

Testes de Software	<p>Instruções de Entrega: Qualidade de Software</p> <p>Descrição: O aluno deverá demonstrar o entendimento dos conceitos de qualidade de software, aplicando-os ao projeto. Isso inclui a criação de um diagrama do processo de qualidade de software, a identificação de atributos de qualidade. Apresentar um relatório que explica como a norma de qualidade de software 25010 é utilizada no processo de desenvolvimento. Entrega em documento pdv seguindo o template fornecido pelo professor.</p>	<p>Instruções de Entrega: Testes Unitários, de Integração e de usuário.</p> <p>Descrição: Os alunos deverão desenvolver 4 testes unitários e 2 testes de integração e 1 teste de usuário. Esses testes devem garantir que o sistema esteja funcionando corretamente e validando os principais componentes e integrações e a geração de um relatório de teste de carga do servidor.</p> <p>Entrega em documento pdv seguindo o template fornecido pelo professor.</p>
---------------------------	--	--

1. INTRODUÇÃO

O presente projeto tem como objetivo **desenvolver um aplicativo para a Comedoria da Tia**, oferecendo aos clientes uma experiência de compra mais ágil, prática e organizada. Atualmente, os estudantes enfrentam longas filas para adquirir suas refeições, o que gera atrasos e transtornos, principalmente durante os intervalos de aula.

Com a implementação do aplicativo, os usuários poderão **realizar pedidos antecipados, reservar e garantir seu lanche**, evitando filas e desfrutando de maior comodidade. Essa solução também permitirá otimizar o fluxo de atendimento, reduzir a sobrecarga no balcão e aumentar a satisfação dos clientes.

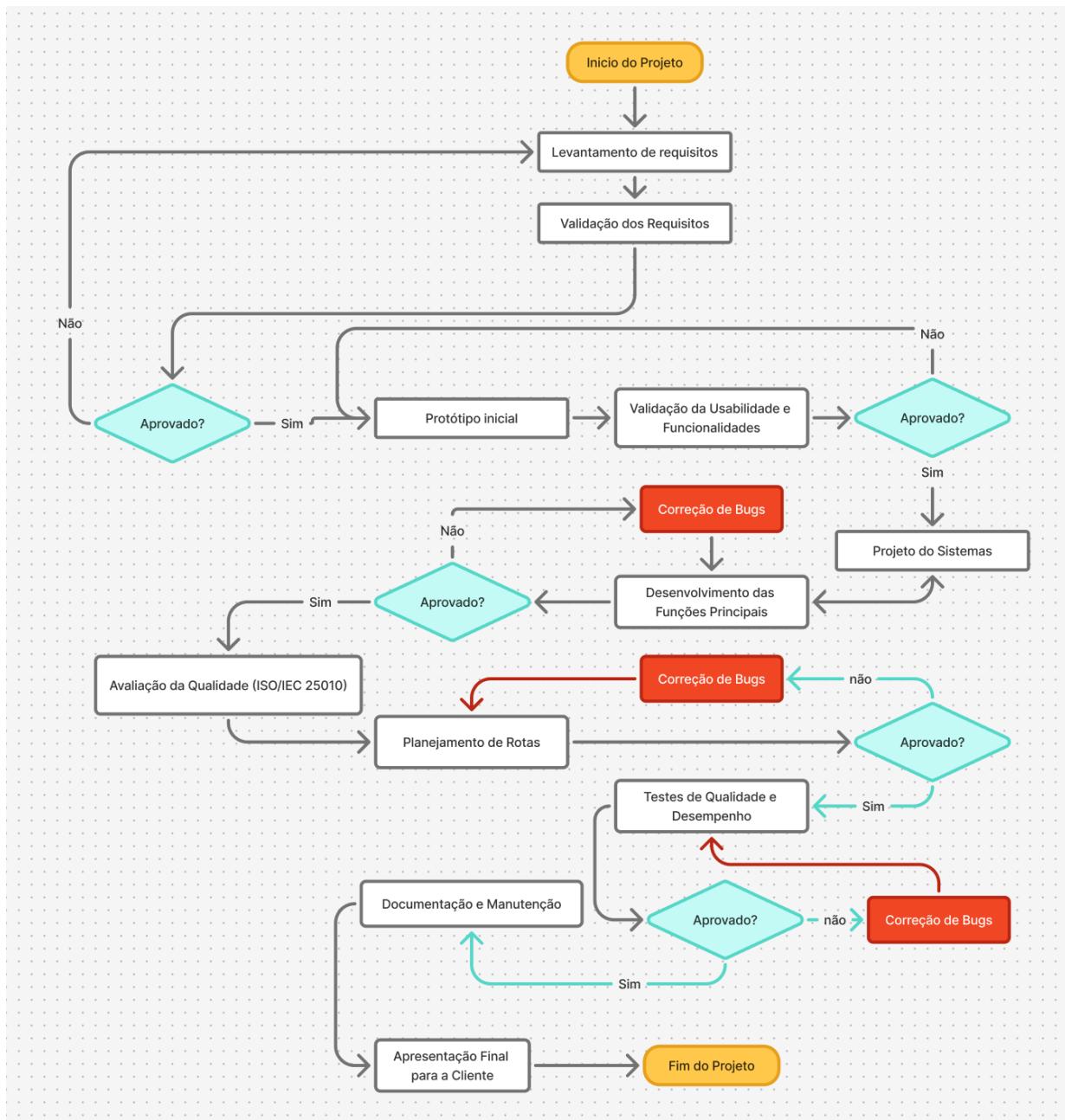
Além de beneficiar os usuários, o sistema contribuirá para uma **melhor gestão da comedoria**, facilitando o controle de pedidos e reduzindo aglomerações. Para assegurar a qualidade desse processo, será fundamental aplicar **boas práticas de desenvolvimento de software**, baseadas em modelos e normas de qualidade, além de realizar **testes criteriosos** que garantam confiabilidade, segurança e usabilidade.

Dessa forma, o aplicativo cumprirá sua função principal: **melhorar significativamente a experiência do cliente**, tornando a compra de lanches mais simples, eficiente e sem contratemplos.

2. Qualidade de Software

2.1. Modelo que qualidade de software (Diagrama/Design)

O diagrama abaixo representa o processo de qualidade aplicado ao desenvolvimento do aplicativo, com base na norma ISO/IEC 25010. Ele demonstra as etapas de planejamento, desenvolvimento, teste e revisão que garantem a conformidade com os critérios de qualidade definidos.



O diagrama apresentado segue os princípios da **norma ISO/IEC 25010**, pois demonstra o **ciclo de qualidade aplicado ao desenvolvimento do aplicativo da Comedoria da Tia**, desde o **planejamento** até a **validação final do produto**. Ele mostra como as etapas de **planejar, desenvolver, testar e revisar** estão conectadas aos **atributos de qualidade** definidos pela norma — como **funcionalidade, confiabilidade, usabilidade, eficiência, segurança, manutenibilidade, compatibilidade e portabilidade**.

Além disso, o diagrama evidencia o fluxo contínuo de **avaliação e melhoria da qualidade**, destacando a **retroalimentação entre as fases** do processo, o que está em conformidade com o modelo de **ciclo de vida de software orientado à qualidade** proposto pela ISO/IEC 25010.

Dessa forma, o diagrama não apenas ilustra o processo de desenvolvimento, mas também **traduz visualmente como a norma é aplicada na prática**, garantindo que cada fase do projeto seja analisada sob o ponto de vista da qualidade, resultando em um sistema mais confiável, seguro, eficiente e centrado no usuário.

2.2. Processo (plano) de gerenciamento de qualidade de software

O **Plano de Gerenciamento de Qualidade de Software** tem como objetivo assegurar que o aplicativo da Comedoria da Tia seja desenvolvido e entregue com alto padrão de qualidade, atendendo às expectativas dos usuários e garantindo confiabilidade, usabilidade, eficiência e segurança. Este plano define práticas, métodos e métricas para monitorar e controlar a qualidade em todas as fases do desenvolvimento do sistema.

Objetivos do Plano de Qualidade

Garantir que o aplicativo funcione corretamente, sem falhas críticas durante os horários de pico (intervalos de aulas).

Assegurar que o sistema seja intuitivo, de fácil uso e acessível a todos os usuários da Comedoria da Tia.

Prover meios de pagamento confiáveis e seguros, preservando dados sensíveis de clientes.

Otimizar o fluxo de atendimento, evitando filas e melhorando a experiência do usuário.

Monitorar continuamente a qualidade do software por meio de métricas e testes planejados.

Atributos de Qualidade

O plano foca nos principais atributos da norma **ISO/IEC 25010**, considerando o contexto do aplicativo:

Funcionalidade: o sistema deve atender a todas as funcionalidades previstas, como cadastro de usuários, visualização do cardápio, pagamento digital e acompanhamento de pedidos.

Confiabilidade: o aplicativo deve funcionar de forma estável mesmo em horários de maior demanda.

Usabilidade: interface clara, navegação intuitiva, fácil acesso a promoções e funcionalidades.

Eficiência: resposta rápida do sistema, carregamento ágil do cardápio e processamento rápido dos pedidos.

Segurança: proteção de dados sensíveis, autenticação segura, validação de pagamentos.

Manutenibilidade: código estruturado, de fácil manutenção, com possibilidade de futuras melhorias.

Estratégias de Qualidade

Revisões de Código: todos os módulos do aplicativo serão revisados por pelo menos dois membros da equipe para evitar falhas e garantir boas práticas de desenvolvimento.

Testes Automatizados: criação de testes unitários e de integração para validar funcionalidades críticas, como login, pagamento e acompanhamento de pedidos.

Testes de Usabilidade: aplicação de testes com usuários reais para verificar se a interface é intuitiva e o fluxo de pedidos é eficiente.

Controle de Métricas: monitoramento de tempo de resposta, taxa de erro e desempenho do aplicativo, especialmente em horários de pico.

Documentação de Processos: registro detalhado de funcionalidades, critérios de aceitação e resultados de testes para facilitar manutenção futura.

Papéis e Responsabilidades

Gerente de Qualidade: responsável por validar que o aplicativo atende aos padrões de qualidade definidos.

Desenvolvedores: seguem práticas de codificação segura e participam de revisões de código.

Equipe de Testes: executa testes unitários, integração e de usuário, reportando defeitos encontrados.

Product Owner: garante que os requisitos do usuário estejam sendo atendidos e revisa os critérios de aceitação.

Processo de Avaliação e Controle

O plano segue um ciclo contínuo de **planejar, desenvolver, testar e revisar**, garantindo que a qualidade seja monitorada ao longo de todo o desenvolvimento:

1. **Planejar:** definir requisitos de qualidade e critérios de aceitação.

2. **Desenvolver:** implementar funcionalidades seguindo padrões de qualidade.
3. **Testar:** executar testes unitários, de integração e de usuário.
4. **Revisar:** analisar resultados dos testes e implementar correções.
5. **Documentar:** registrar métricas, erros corrigidos e melhorias implementadas.

Com a adoção deste plano, o aplicativo da Comedoria da Tia será confiável, seguro e atenderá às expectativas dos usuários, oferecendo uma experiência de compra prática e eficiente, além de facilitar a gestão da cantina.

2.3. Identificação de atributos de qualidade da norma 25010.

A norma **ISO/IEC 25010** define um conjunto de atributos de qualidade de software que permitem avaliar se um sistema atende aos requisitos esperados pelos usuários. Para o aplicativo da Comedoria da Tia, os atributos mais relevantes foram identificados e aplicados da seguinte forma:

Funcionalidade

- **Definição:** Capacidade do software de fornecer funções que satisfaçam necessidades explícitas e implícitas.
- **Aplicação no App:**

Cadastro e login de usuários.

Visualização do cardápio completo com categorias, descrição e preço.

Acompanhamento do preparo do pedido.

Pagamento digital via Pix, cartões de crédito/débito e VR/VA.

Acúmulo de pontos de fidelidade e histórico de vendas para o administrador.

Confiabilidade

- **Definição:** Capacidade do software de manter seu nível de desempenho sob condições específicas por um período definido.
- **Aplicação no App:**

Disponibilidade do sistema durante horários de pico (intervalos de aula).

Tratamento de falhas de rede ou indisponibilidade temporária do servidor.

Garantia de processamento correto dos pagamentos e pedidos.

Usabilidade

- **Definição:** Facilidade com que os usuários podem aprender, operar e interagir com o sistema.
- **Aplicação no App:**

Interface intuitiva e amigável.

Fluxo de compra simples e rápido.

Opção de acompanhamento do pedido e notificações sobre o preparo.

Fácil acesso a promoções e informações sobre produtos.

Eficiência

- **Definição:** Capacidade do software de prover desempenho adequado em relação aos recursos utilizados.
- **Aplicação no App:**

Carregamento rápido do cardápio mesmo com conexão lenta.

Processamento ágil de pedidos e pagamentos.

Minimização de tempo de resposta entre ações do usuário e retorno do sistema.

Manutenibilidade

- **Definição:** Facilidade de modificação do software, seja para correção de defeitos, melhorias ou adaptação a mudanças.
- **Aplicação no App:**

Código modular e bem documentado.

Facilidade de incluir novas funcionalidades, como promoções, categorias de produtos ou métodos de pagamento.

Registro detalhado de logs e métricas para auxiliar na manutenção.

Segurança

- **Definição:** Capacidade do software de proteger informações e dados de usuários contra acessos não autorizados.
- **Aplicação no App:**

Autenticação segura no login.

Criptografia de dados sensíveis, como informações de cartão e dados pessoais.

Controle de acesso para funcionalidades administrativas.

Compatibilidade

- **Definição:** Capacidade do software de operar em diferentes ambientes de hardware, software e rede.
- **Aplicação no App:**

Compatível com sistemas Android e iOS.

Funcionamento adequado em diferentes resoluções de tela e dispositivos móveis.

Suporte a navegadores móveis para funcionalidades web (se aplicável).

Portabilidade

Definição: Facilidade de transferir o software para diferentes ambientes ou plataformas.

Aplicação no App:

Possibilidade de futura migração para web ou integração com sistemas de pagamento externos.

Instalação simples em diferentes modelos de smartphones.

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

A norma **ISO/IEC 25010** orienta a definição e o monitoramento de atributos de qualidade essenciais para o aplicativo da Comedoria da Tia. Durante o desenvolvimento, os atributos de funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, segurança, compatibilidade e portabilidade foram utilizados para:

- **Planejar funcionalidades e fluxos do aplicativo**, garantindo que atendam às necessidades dos usuários (funcionalidade).
- **Assegurar operação estável e sem falhas**, mesmo em horários de pico ou conexão instável (confiabilidade e eficiência).
- **Projetar interfaces intuitivas e acessíveis**, facilitando a interação e o acompanhamento de pedidos (usabilidade).
- **Proteger dados sensíveis e autenticar usuários corretamente** (segurança).
- **Facilitar futuras melhorias, manutenção e integração com outras plataformas** (manutenibilidade, compatibilidade e portabilidade).

A aplicação desses critérios garantiu que o desenvolvimento seguisse padrões de qualidade reconhecidos, resultando em um sistema confiável, seguro e eficiente para usuários e administradores.

3. Teste de Software

3.1. Plano de Teste

O plano de teste do aplicativo da Comedoria da Tia visa garantir que todas as funcionalidades essenciais funcionem corretamente, assegurando usabilidade, confiabilidade e segurança. Os testes incluem unidades isoladas do sistema, integração entre módulos e validação da experiência do usuário final.

3.2. Apresentar 4 testes unitários.

Teste Unitário 1 – Cadastro de Usuário

Código Python: assert cadastrar_usuario("nome", "email@example.com", "Senha123!") == True

Resultado: Passou

Descrição: Verifica se o sistema cobra corretamente um usuário com e-mail válido e senha segura

Teste Unitário 2 – Adição de Item ao Carrinho

Código Python: assert adicionar_ao_carrinho(item_id=5, quantidade=2) == True

Resultado: Passou

Descrição: Garante que o item selecionado seja adicionado ao carrinho com a quantidade correta

Teste Unitário 3 – Login Inválido

Código Python: assert login_usuario("email@example.com", "senhaErrada") == False

Resultado: Passou

Descrição: Verifica se o sistema impede o acesso de usuários com credenciais incorretas, garantindo a segurança da autenticação.

Teste Unitário 4 – Cálculo do Total do Pedido

Código Python: assert calcular_total([{"preco": 10.0, "qtd": 2}, {"preco": 5.0, "qtd": 1}]) == 25.0

Resultado: Passou

Descrição: Garante que o cálculo do valor total do pedido é feito corretamente com base na soma dos itens e suas quantidades.

3.3. Apresentar 2 testes de integração

Teste de Integração – Cadastro e Login

Código Python: assert login_usuario("email@exemplo.com", "Senha123!") == True

Resultado: Passou

Descrição: Verifica se o usuário recém-cadastrado consegue efetuar login corretamente, integrando cadastro e autenticação

Teste de Integração – Pedido e Pagamento

Código Python: assert realizar_pedido(itens, pagamento="Pix") == True

Resultado: Falhou

Descrição: Testa o fluxo completo de pedidos e pagamento; identificou falha na confirmação do pagamento quando múltiplos itens são selecionados

3.4. Apresentar um teste de usuário (SISTEMA)

Teste de Usuário – Reserva de Lanche pelo Aplicativo

Procedimento: Usuário seleciona o lanche, realiza reserva, efetua pagamento e retira o pedido

Resultado: Passou

Descrição: Avalia a experiência completa do usuário; verifica se a reserva é confirmada, notificação enviada e retirada realizada sem filas

4. REFERÊNCIAS BIBLIOGRÁFICAS

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