**Engenharia de Software**

2023-01

| **Professor(a)(es)** | RADAMÉS PEREIRA |
| --- | --- |

Atributos de uma boa Especificação de Requisitos de Projeto de Software:

* Clareza
* Não Ambígua
* Completa
* Simples
* Bem escrita

**UNIVERSIDADE COMUNITÁRIA REGIONAL DE CHAPECÓ - UNOCHAPECÓ**

**ÁREA DE CIÊNCIAS EXATAS E AMBIENTAIS**

**CIÊNCIA DA COMPUTAÇÃO**

Título

...

Equipe (engenheiro e desenvolvedores): **...**

**...**

**…**

Introdução

## Resumo do Projeto (Descrição textual)

Em uma indústria alimentícia chamada "Delícias Gourmet", o controle de materiais é um processo-chave para garantir a qualidade e a eficiência dos produtos finais. Nesta história, vamos explorar como diferentes atores desempenham funções importantes em seus respectivos setores, interagindo e colaborando para manter o sistema de controle de materiais funcionando sem problemas.

A história começa com João, o gerente de compras, que é responsável por adquirir os ingredientes e os materiais necessários para a produção dos alimentos. Ele trabalha em estreita colaboração com os fornecedores para garantir a qualidade dos produtos e negocia os melhores preços para reduzir os custos de produção.

Maria, a supervisora de estoque, trabalha em conjunto com João. Ela é responsável por gerenciar o armazenamento dos materiais e garantir que todos os itens estejam devidamente organizados e estocados. Quando os ingredientes e materiais chegam ao armazém, Maria confere e verifica se tudo está em conformidade com os pedidos feitos por João.

Dentro da fábrica, Pedro, o coordenador de produção, coordena a utilização dos ingredientes e materiais no processo de fabricação. Ele colabora com Maria para garantir que os itens necessários sejam entregues no momento certo e na quantidade adequada. Pedro também trabalha com os líderes de cada linha de produção para planejar e otimizar a utilização dos materiais.

Na área de controle de qualidade, Ana, a analista de qualidade, monitora o processo de produção, verificando se todos os ingredientes e materiais estão dentro dos padrões exigidos. Ela também é responsável por conduzir inspeções regulares e garantir que a produção esteja em conformidade com as normas de segurança alimentar e regulamentações governamentais.

Por fim, a história se completa com Carlos, o responsável pelas vendas e distribuição dos produtos. Ele coordena a entrega dos produtos acabados aos clientes e mantém um registro de todas as vendas realizadas. Carlos também fornece informações sobre a demanda do mercado e as preferências dos clientes, o que ajuda João na seleção de fornecedores e ingredientes para futuras compras.

Esta história ilustra como os diferentes atores em uma indústria alimentícia colaboram e desempenham funções cruciais para manter o sistema manual de controle de materiais funcionando de forma eficiente. A interdependência entre as atividades de João, Maria, Pedro, Ana e Carlos é fundamental para garantir que a "Delícias Gourmet" mantenha sua reputação de qualidade e satisfação do cliente.

## Plataforma de desenvolvimento (O equipamento dos desenvolvedores e ferramentas de software)

Descreve-se aqui uma primeira visão das tecnologias para desenvolvimento do projeto de software.

## Plataforma de operação (O equipamento do cliente/usuário do sistema)

Descreve-se aqui uma primeira visão das tecnologias para operacionalização.

## Definições e siglas (quaisquer siglas utilizadas no domínio, do vocabulário do usuário)

Descreve-se aqui a definição de todas as siglas, abreviações e termos usados.

## Perspectiva do produto

### Modos de operação (Meios de acesso ao sistema/Arquitetura do sistema)

### Identificam-se aqui os modos requeridos de operação, tais como: Back-End-Front-End, Móvel, Stand-Alone, ...

### Requisitos de adaptação ao ambiente (Aspectos legais para aderência a legislação, ambiente de operação crítica como indústria, automação, protocolos de comunicação específicos)

Definem-se aqui possíveis requisitos de adaptação do produto aos ambientes particulares onde ele será implantado. Por exemplo, parâmetros e métodos de configuração requeridos para ambientes específicos devem ser descritos aqui.

| Número de ordem | Requisito | Detalhes |
| --- | --- | --- |
| 1 | Configuração de ticket de venda e da Nota Fiscal Eletrônica | Configuração dos campos de formulário com interface responsiva. |

## Funções do produto (Funções básicas, R1.1 ..., R1.2 ...)

R1.1 - Registro e monitoramento de estoque: o sistema deve ser capaz de registrar e monitorar o estoque de todos os ingredientes e materiais utilizados na produção.

R1.2 - Compras e fornecedores: o sistema deve permitir que o gerente de compras registre informações sobre fornecedores e realize pedidos de compra.

R1.3 - Gestão de qualidade: o sistema deve ter a capacidade de monitorar a qualidade dos ingredientes e materiais utilizados na produção, incluindo testes de laboratório e conformidade com regulamentações governamentais.

R1.4 - Coordenação de produção: o sistema deve permitir que o coordenador de produção planeje a utilização dos materiais para a produção de produtos acabados, incluindo a previsão de demanda e a otimização do uso de materiais.

R1.5 - Controle de qualidade: o sistema deve ser capaz de monitorar a qualidade dos produtos finais, incluindo a verificação de conformidade com as especificações de qualidade e segurança alimentar.

R1.6 - Vendas e distribuição: o sistema deve permitir que a equipe de vendas gerencie as vendas e a distribuição de produtos acabados aos clientes, incluindo a gestão de estoques e a rastreabilidade de produtos.

R1.7 - Relatórios e análises: o sistema deve permitir a geração de relatórios e análises que permitam a gestão da empresa a tomar decisões informadas sobre as compras, produção e vendas.

R1.8 - Integração com outros sistemas: o sistema de controle de materiais pode precisar se integrar com outros sistemas, como sistemas financeiros e de gerenciamento de produção, para fornecer informações completas e precisas sobre o desempenho da empresa.

R1.9 - Segurança: o sistema deve ser seguro para evitar o acesso não autorizado a informações confidenciais, como detalhes de fornecedores e dados de clientes.

R1.10 - Usabilidade: o sistema deve ser fácil de usar e navegar, com uma interface amigável e intuitiva que permita que os usuários realizem suas tarefas sem dificuldade.

R1.11 - Escalabilidade: o sistema deve ser escalável para atender às necessidades em constante evolução da empresa, como a expansão da linha de produtos, o aumento da demanda do mercado e o crescimento do negócio.

R1.12 - Suporte técnico: o sistema deve ter suporte técnico disponível para ajudar os usuários a resolver problemas técnicos e fornecer treinamento sobre como usar efetivamente o sistema.

## Características dos usuários (identificação da formação dos usuários para suas especialidades no uso do sistema, tal como ensino-médio, graduação, especialista, engenheiro, doutor, etc...)

Descrevem-se aqui as principais características dos grupos de usuários esperados para o produto, tais como cargo ou função, permissão de acesso, frequência de uso, nível de instrução, proficiência no processo de negócio e proficiência em informática.

## Restrições (Lei Geral de proteção de dados, etc...)

Descrevem-se aqui aspectos técnicos e gerenciais que possam limitar as opções dos desenvolvedores, tais como restrições legais.

## Hipóteses de trabalho (Sistema operacional, versão de ferramentas de software, licenças de bibliotecas e de subs-sistemas)

Descrevem-se aqui fatores que não são restrições limitativas do desempenho, como na subseção anterior, mas fatores cuja alteração requer modificações na ER, como, por exemplo, versão a ser utilizada do ambiente operacional ou plataforma de desenvolvimento.

# Requisitos específicos

## Interfaces externas

### Visão geral (Tipos de interface, relatórios, gráficos, dashboards)

Descreve-se aqui, de forma detalhada, todas as entradas e saídas do produto.

### Requisitos para interfaces gráficas de usuário (markups/wireframes)

Sugere-se, no caso de interfaces gráficas, a inclusão dos seguintes elementos:

* Um esboço do layout gráfico sugerido para a interface;
* Uma descrição dos relacionamentos com outras interfaces;
* Um diagrama de estados/atividades, caso necessário para melhor entender-se o comportamento requerido da interface;
* Uma lista dos campos de dados da interface;
* Uma lista dos comandos da interface;
* BPM;

## Requisitos funcionais

### Diagramas de casos de uso (Modelo UML de Casos de Uso)

@startuml DeliciasGourmet

left to right direction

actor "Gerente de Compras" as g

actor "Vendedor" as v

actor "Coordenador de Produção" as c

actor "Supervisor de estoque" as s

actor "Analista de qualidade" as a

rectangle SistemaDeliciasGourmet{

usecase "Adquiri os materiais e ingredientes necessários"

as UC1

usecase "Gerencia os materiais em estoque" as UC2

usecase "Coordena a utilização dos ingredientes e

materiais" as UC3

usecase "Monitora o processo de produção" as UC4

usecase "Conduz inspeçõe e garante conformidade nos

produtos" as UC5

usecase "Coordena a entrega de produtos" as UC6 usecase "Fornece informações sobre a demanda" as UC7

g -- UC1

g --> s

s -- UC2

c -- UC3

UC1 --UC2

a--UC4 c--> UC6

c-->UC7

g-- v

v--UC6

}

@enduml

## 

### Fluxos dos casos de uso (Casos de Uso Expandidos e Diagramas de Atividades)

Caso de Uso: Registro e monitoramento de estoque: o sistema deve ser capaz de registrar e monitorar o estoque de todos os ingredientes e materiais utilizados na produção.

1. Registrar todos os ingredientes e materiais: O sistema deve ser capaz de registrar e armazenar informações detalhadas sobre cada ingrediente e material utilizado na produção, como nome, descrição, fornecedor, quantidade inicial, unidade de medida e data de validade.

2. Atualização em tempo real: O sistema deve permitir atualizações em tempo real do estoque, à medida que os ingredientes e materiais são recebidos, utilizados e descartados. Isso garantirá uma visão precisa do estoque atualizado a qualquer momento.

3. Alertas de estoque baixo: O sistema deve fornecer alertas automáticos quando os níveis de estoque de um determinado ingrediente ou material atingirem um ponto de reabastecimento pré-definido. Isso permitirá que a equipe tome medidas para reabastecer o estoque antes que

ocorram faltas.

4. Rastreamento de data de validade: O sistema deve monitorar e emitir alertas sobre os itens com datas de validade próximas ou expiradas. Isso ajudará a evitar o desperdício de ingredientes e garantir a qualidade dos pratos servidos.

5. Relatórios de estoque: O sistema deve gerar relatórios detalhados sobre o estoque atual, histórico de movimentações e tendências de consumo. Isso ajudará a equipe de gerenciamento a tomar decisões informadas sobre compras, redução de desperdícios e otimização do estoque.

**CÓDIGO EM PUML:**

**@startuml**

**|Main|**

**start**

**:Registrar ingredientes e materiais;**

**:Atualizar estoque;**

**if (Estoque baixo?) then (sim)**

**:Enviar alerta de reabastecimento;**

**else (não)**

**stop**

**endif**

**|Main| :Monitorar data de validade;**

**if (Item vencido?) then (sim)**

**:Enviar alerta de vencimento;**

**else (não)**

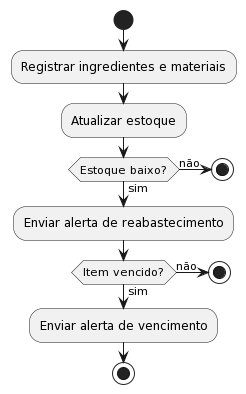
**stop**

**endif**

**|Main| :Gerar relatórios de estoque;**

**stop**

**@enduml**

****

**Descrição do código: Neste diagrama, a ação "Registrar ingredientes e materiais" representa o processo de entrada de novos itens no estoque, atualizando as informações correspondentes. Em seguida, o estoque é atualizado e é feita uma verificação para determinar se o estoque está baixo. Se for o caso, um alerta de reabastecimento é enviado. Além disso, o sistema monitora a data de validade dos itens e envia um alerta se algum item estiver próximo do vencimento ou já vencido. Por fim, o sistema gera relatórios de estoque para análise e tomada de decisões.  
  
==============================================================**

2° Caso de Uso: Vendas e distribuição: o sistema deve permitir que a equipe de vendas gerencie as vendas e a distribuição de produtos acabados aos clientes, incluindo a gestão de estoques e a rastreabilidade de produtos.

Gestão de Vendas:

1.O sistema deve permitir que a equipe de vendas registre pedidos de clientes, incluindo detalhes do produto, quantidade, preço e informações de contato do cliente.

Os vendedores devem ter acesso a informações atualizadas sobre o estoque de produtos para fornecer aos clientes informações precisas sobre disponibilidade e prazos de entrega.

O sistema deve permitir o processamento de pagamentos e a geração de faturas para os pedidos dos clientes.

Deve ser possível rastrear o histórico de vendas de cada cliente, incluindo pedidos anteriores, pagamentos e comunicações.

Gestão de Estoque:

2.O sistema deve permitir que a equipe de vendas monitore o estoque de produtos acabados em tempo real.

Deve haver um recurso de controle de estoque que registre as entradas e saídas de produtos para manter um registro preciso dos níveis de estoque.

O sistema deve alertar automaticamente a equipe de vendas quando o estoque de um determinado produto atingir um nível mínimo predefinido, para que ações adequadas possam ser tomadas para reabastecer os produtos.

Rastreabilidade de Produtos:

3.O sistema deve permitir a rastreabilidade dos produtos acabados desde a fabricação até a entrega ao cliente.

Cada produto deve ser identificado por um número de série exclusivo ou código de barras, permitindo que sua localização e histórico sejam rastreados.

As informações sobre a data de fabricação, lote, data de validade (se aplicável) e detalhes do processo de fabricação devem ser registradas para cada produto.

Relatórios e Análises:

4.O sistema deve fornecer recursos de geração de relatórios para que a equipe de vendas possa obter informações sobre o desempenho das vendas, estoques e outros indicadores relevantes.

Os relatórios podem incluir análises de vendas por produto, por cliente, por região geográfica, por período de tempo, entre outros.

CÓDIGO EM PUML:

**@startuml**

**|Main|**

**start**

**:Registrar Pedido;**

**:Verificar Estoque;**

**if (Produto Disponível?) then (yes)**

**:Processar Pagamento;**

**:Gerar Fatura;**

**if (Cliente Novo?) then (yes)**

**:Cadastrar Cliente;**

**else (no)**

**:Atualizar Histórico de Vendas;**

**endif**

**:Atualizar Estoque;**

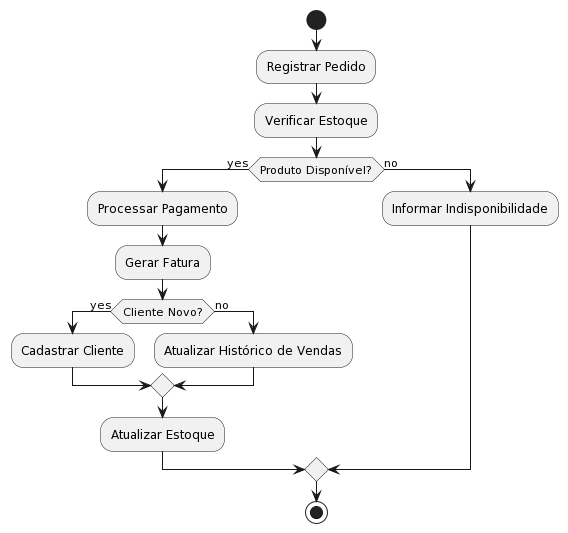
**else (no)**

**:Informar Indisponibilidade;**

**endif**

**stop**

**@enduml**

****

**Descrição do código: Neste diagrama, o processo começa com o registro do pedido pelo vendedor. Em seguida, o sistema verifica se o produto está disponível no estoque. Se o produto estiver disponível, o sistema prossegue para processar o pagamento, gerar uma fatura e, se necessário, cadastrar o cliente. Caso o cliente já exista, o histórico de vendas é atualizado. Após essas etapas, o sistema atualiza o estoque. Se o produto não estiver disponível, o sistema informa a indisponibilidade ao vendedor.**

## Requisitos não-funcionais

### Requisitos de desempenho (Velocidade de banda, tempo de resposta das interfaces e/ou impressão de relatórios)

Requisitos de desempenho devem ser especificados de forma quantitativa e mensurável.

### Requisitos de dados persistentes (Sistemas de gerenciamento de banco de dados e modelo de persistência)

Descrevem-se aqui estruturas lógicas de dados persistentes (que mantém seu valor após a execução do programa) que sejam usadas pelo produto. Cada estrutura de dados pode ser, por exemplo, um arquivo convencional ou uma tabela em um banco de dados.

INCLUIR AQUI O MODELO DE BANCO DE DADOS

### Restrições ao desenho

Restrições de projeto impostas por padrões externos, com influência da legislação, tipo legislação fiscal municipal, estadual e federal.

### Atributos de Qualidade

Indica os atributos de qualidade, seguindo as características e subcaracterísticas recomendadas pela norma [ISO-9126](https://drive.google.com/open?id=14-OfxrvT5pD4sblFt234r8NPBNwt2HmG).

## Objetos/Classes

### Modelo Conceitual/Classes de Análise/Modelo de Domínio (Classes, Associações, nomes das associações, Multiplicidades e Atributos

@startuml

class GerenteCompras {

- nome: String

- adquirirIngredientes(): void

}

class Fornecedor {

- nome: String

- fornecerIngredientes(): void

}

class SupervisoraEstoque {

- nome: String

- gerenciarArmazenamento(): void

}

class CoordenadorProducao {

- nome: String

- coordenarProducao(): void

}

class LiderProducao {

- nome: String

- planejarProducao(): void

}

class AnalistaQualidade {

- nome: String

- monitorarProducao(): void

}

class Cliente {

- nome: String

- fazerPedido(): void

}

class Produto {

- nome: String

- verificarQualidade(): void

}

class Vendas {

- nome: String

- coordenarDistribuicao(): void

}

GerenteCompras "1" -- "\*" Fornecedor

GerenteCompras "1" -- "1" SupervisoraEstoque

SupervisoraEstoque "1" -- "1" CoordenadorProducao

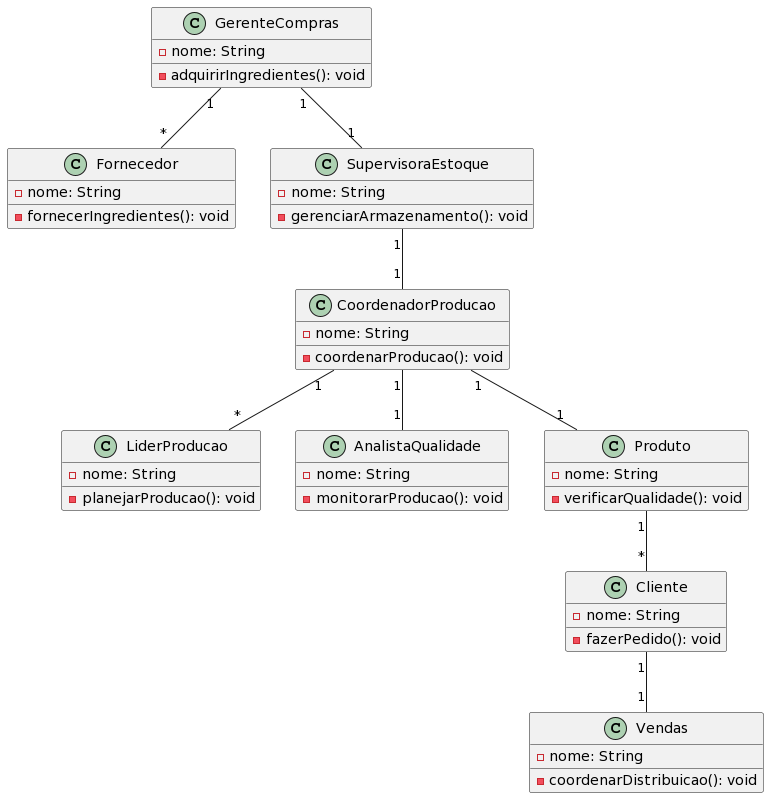
CoordenadorProducao "1" -- "\*" LiderProducao

CoordenadorProducao "1" -- "1" AnalistaQualidade

CoordenadorProducao "1" -- "1" Produto

Produto "1" -- "\*" Cliente

Cliente "1" -- "1" Vendas

@enduml  
  


### DSS – Diagramas de Sequência do Sistema (Eventos e Operações) de Casos de Uso (denotando as mensagens entre os objetos do domínio para atender ao Caso de Uso).

### Contratos (das Operações do DSS)

### Classes de Implementação - Diagrama de Classes (Classes, Associações, nomes das associações, Multiplicidades, Atributos e Métodos). Atribuição de responsabilidades com GRASP (General Responsibility Assignment Software Patterns) que são um conjunto de princípios e diretrizes para atribuição de responsabilidades em projetos de software orientados a objetos.

# Análise de UCP

As tabelas de escopo de valor do produto e tempo de desenvolvimento com Use Case Points - UCP.

Referências:

*IEEE Std. 830 – 1993. IEEE Recommended Practice for Software Requirements Specifications.*

*IEEE ISO/IEC/IEEE 29148 – 2011. IEEE Systems and software engineering — Life cycle processes — Requirements engineering*

**OBSERVAÇÃO: Os itens deste modelo de especificação, recomendado pela IEEE, poderão ser complementados com novos itens caso sejam justificáveis.**