Aqui estão as informações solicitadas com base no caso de uso fornecido:

1.1 Resumo do Projeto:

O sistema a ser desenvolvido é o Sistema de Produção da Indústria de Concreto. Ele envolve o registro de pedidos, controle de produção, gerenciamento de estoque, transporte e entrega, recebimento de pagamento, gerenciamento financeiro e registro de vendas.

1.2 Plataforma de desenvolvimento:

As tecnologias utilizadas para o desenvolvimento do projeto de software podem incluir:

- Linguagem de programação: C#

- Framework: ASP.NET

- Controle de versão: Git

- Banco de dados: SQL Server

1.3 Plataforma de operação:

Considerando que o usuário pode usar um desktop ou notebook, a plataforma de operação pode ser:

- Windows (versão mais recente compatível com as ferramentas selecionadas)

- Linux (distribuição mais popular compatível com as ferramentas selecionadas)

- macOS (versão mais recente compatível com as ferramentas selecionadas)

1.4 Definições e siglas:

- DV: Departamento de Vendas

- UC: Caso de Uso

- AE: Almoxarifado e Estocagem

- SRD: Setor de Recebimento e Distribuição

- DF: Departamento Financeiro

- GD: Gerenciador de Qualidade

1.5 Perspectiva do produto:

1.5.1 Modos de operação:

O sistema pode ser acessado por meio de diferentes modos, como Back-End, Front-End, Móvel, Stand-Alone, etc.

1.5.2 Requisitos de adaptação ao ambiente:

O sistema deve estar em conformidade com as regulamentações legais aplicáveis e atender a requisitos específicos de ambientes críticos, como a indústria, automação e protocolos de comunicação.

Requisitos:

Requisito 1:

1.6 Funções do produto:

- Registrar Pedido

- Cancelar Pedido

- Escalonar Pedido

- Controlar Produção

- Gerenciar Estoque

- Gerenciar Transporte e Entrega

- Receber Pagamento

- Registrar Venda

1.7 Características dos usuários:

Os usuários do sistema podem ter diferentes formações e especialidades, como ensino médio, graduação, especialistas, engenheiros, doutores, etc.

1.8 Restrições:

O sistema deve estar em conformidade com a legislação vigente, como a Lei Geral de Proteção de Dados.

1.9 Hipóteses de trabalho:

- Sistema operacional compatível: Windows 10 (ou mais recente versão mais recente), Linux Ubuntu 20.04 (ou outra distribuição popular), macOS Big Sur (ou outra versão)

- Versão específica das ferramentas de software:

- C#: .NET Framework 4.8 (ou versão mais recente)

- ASP.NET: ASP.NET Core 5 (ou versão mais recente)

- Git: Git 2.32 (ou versão mais recente)

- SQL Server: SQL Server 2019 (ou versão mais recente)

- Licenças necessárias para bibliotecas e subsistemas:

2 Requisitos específicos

2.1 Interfaces externas

2.1.1 Visão geral:

O sistema de produção da indústria de concreto possui várias interfaces externas que descrevem as entradas e saídas do produto. Algumas das interfaces relevantes podem incluir:

- Integração com o Departamento de Vendas: O sistema recebe pedidos de venda e informações sobre os produtos solicitados.

- Integração com o Almoxarifado e Estocagem: O sistema interage com o almoxarifado para controlar o estoque de matéria-prima e produtos acabados.

- Integração com o Setor de Recebimento e Distribuição: O sistema troca informações sobre os pedidos e a entrega dos produtos.

- Integração com a Transportadora: O sistema fornece dados sobre os produtos a serem transportados e recebe informações sobre a logística de entrega.

- Integração com o Departamento Financeiro: O sistema registra os pagamentos recebidos e fornece informações financeiras para o departamento.

- Integração com o Gerenciador de Qualidade: O sistema troca informações sobre a qualidade dos produtos e realiza o controle de qualidade.

2.1.2 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;

(a realizar).

2.2 Requisitos funcionais

2.2.1 Diagramas de casos de uso

Incluir todos os casos de uso que se pretende implementar em uma liberação.

Pode-se incluir ainda: um certo caso de uso e seus relacionamentos, todos os casos

de uso para um certo ator.

@startuml

left to right direction

skinparam packageStyle rect

actor Cliente

actor "Departamento de Vendas" as dv

actor Produção

actor "Almoxarifado e Estocagem" as ae

actor "Setor de Recebimento e Distribuição" as srd

actor Transportadora

actor "Departamento Financeiro" as df

actor "Gerenciador de qualidade" as gd

rectangle "Sistema de Produção da Indústria de Concreto" {

usecase "Registrar Pedido" as UC1

usecase "Cancelar Pedido" as UC7

usecase "Escalonar Pedido" as UC2

usecase "Controlar Produção" as UC3

usecase "Gerenciar Estoque" as UC4

usecase "Gerenciar Transporte e Entrega" as UC5

usecase "Receber Pagamento" as UC6

usecase "Registrar venda" as UC8

Cliente -- UC1

Cliente -- UC7 : extend

UC1 -- dv

dv -- UC2

UC2 -- Produção

Produção -- UC3

UC3 -- ae

UC3 -- gd

UC4 -- Produção

UC4 -- ae

srd -- UC4

UC5 -- Transportadora

UC6 -- df

UC8 -- df

UC2 -- Produção : <<extend>>

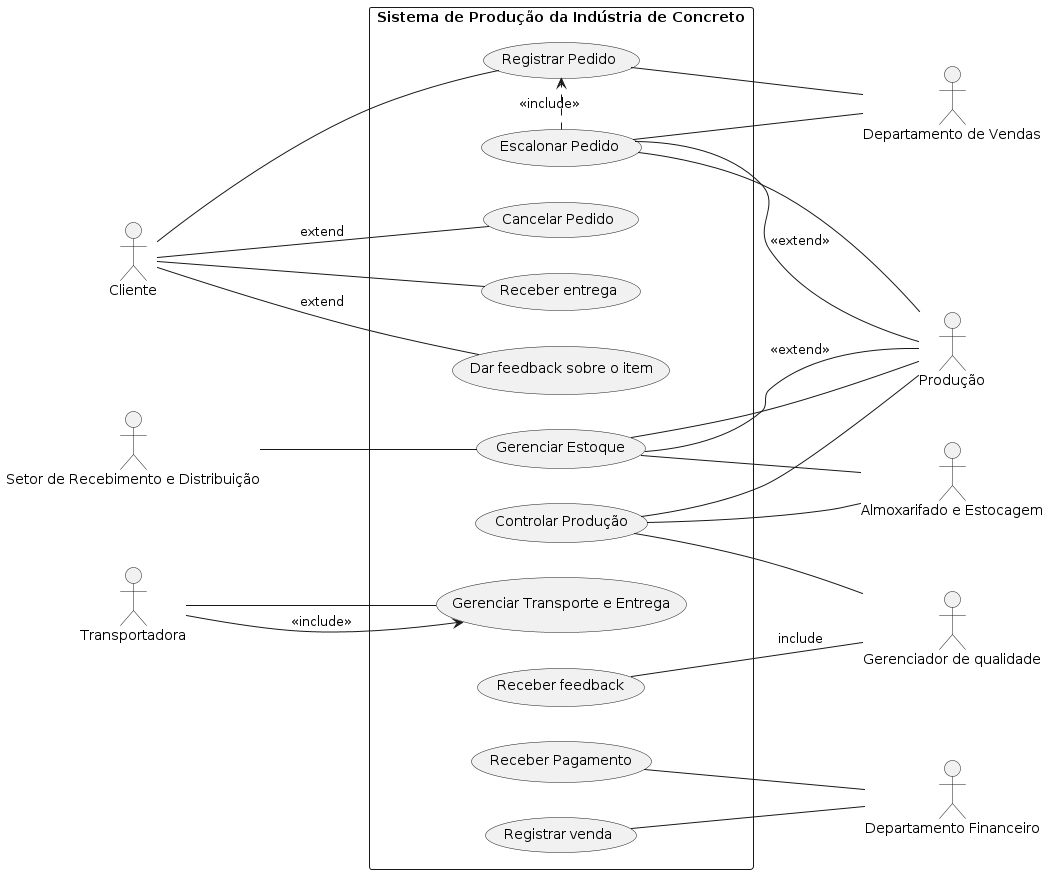
UC4 -- Produção : <<extend>>

UC2 .> UC1 : <<include>>

Transportadora --> UC5 : <<include>>

}

@endum



2.2.2 Fluxos dos casos de uso (caso de uso expandido e diagrama

de atividades)

Nome do caso de uso: Gerenciar Produção e Vendas na Indústria de Concreto

Resumo: Este caso de uso descreve o processo de gerenciamento da produção e vendas na indústria de concreto, desde o registro do pedido até o recebimento do pagamento e feedback dos clientes.

Ator principal: Cliente

Interessados:

- Cliente: Deseja registrar pedidos, receber entregas, fornecer feedback e efetuar o pagamento.

- Departamento de Vendas: Responsável pelo registro de pedidos e escalonamento de pedidos.

- Produção: Responsável por controlar a produção de concreto.

- Almoxarifado e Estocagem: Responsável pelo gerenciamento do estoque de materiais.

- Setor de Recebimento e Distribuição: Responsável pelo recebimento de entregas.

- Transportadora: Responsável pelo transporte e entrega dos produtos.

- Departamento Financeiro: Responsável por receber o pagamento dos pedidos.

- Gerenciador de qualidade: Responsável por receber feedback dos clientes e garantir a qualidade dos produtos.

Pré-condições:

- O sistema de produção está funcionando corretamente.

- O estoque de materiais está disponível.

- A transportadora está disponível para realizar as entregas.

- O cliente possui acesso ao sistema.

Fluxo principal de eventos:

1. O cliente solicita o registro de um pedido (UC1).

2. O Departamento de Vendas registra o pedido no sistema.

3. O Departamento de Vendas escalona o pedido para a Produção (UC2).

4. A Produção controla a produção do concreto (UC3).

5. A Produção verifica o estoque disponível no Almoxarifado e Estocagem (UC4).

6. O Almoxarifado e Estocagem fornece os materiais necessários para a produção.

7. A Produção inicia a produção do concreto.

8. O Gerenciador de qualidade verifica a qualidade do concreto produzido.

9. A Produção finaliza a produção do concreto.

10. A Produção solicita à Transportadora o transporte e entrega do pedido (UC5).

11. A Transportadora realiza o transporte e entrega do pedido.

12. O cliente recebe a entrega e confirma o recebimento (UC9).

13. O Departamento Financeiro recebe o pagamento do cliente (UC6).

14. O Departamento Financeiro registra a venda (UC8).

15. O Gerenciador de qualidade recebe feedback do cliente sobre o item entregue (UC11).

16. O sistema registra o feedback do cliente.

17. O Gerenciador de qualidade analisa o feedback e toma as ações necessárias.

Pós-condições:

- O pedido é registrado no sistema.

- A produção do concreto é controlada e finalizada.

- O estoque é atualizado.

- A entrega do pedido é realizada com sucesso.

- O pagamento é recebido.

- A venda é registrada.

- O feedback do cliente é registrado e analisado pelo Gerenciador de qualidade.

Fluxos alternativos:

- No passo 2, se o cliente decidir cancelar o pedido, o Departamento de Vendas cancela o pedido (UC7).

- No passo 11, se a Transportadora não estiver disponível, o Departamento de Vendas escalona o pedido para outra transportadora.

- No passo 13, se o cliente não efetuar o pagamento, o Departamento Financeiro notifica o cliente sobre o pagamento pendente.

- No passo 15, se o cliente não fornecer feedback, o Gerenciador de qualidade segue com as análises com base nas informações disponíveis.

2.3 Requisitos não-funcionais:

2.3.1 Requisitos de desempenho:

- Velocidade de banda: O sistema requer uma conexão de banda larga com velocidade mínima de 10 Mbps para garantir a transferência eficiente de dados e recursos.

2.3.2 Requisitos de dados persistentes:

- Sistemas de gerenciamento de banco de dados: O sistema utiliza o Microsoft SQL Server 2019(ou mais recente) para armazenar e gerenciar os dados persistentes.

- Modelo de persistência: O sistema possui tabelas principais, como "Pedidos", "Produtos" e "Clientes", com seus respectivos campos e relacionamentos estabelecidos para garantir a integridade dos dados.

2.3.3 Restrições ao desenho:

- Restrições de projeto: O sistema deve estar em conformidade com a legislação fiscal federal e estadual, garantindo a correta emissão de notas fiscais eletrônicas e o cálculo de impostos conforme as regulamentações vigentes.

2.3.4 Atributos de Qualidade:

- Confiabilidade: O sistema deve ser altamente confiável, garantindo a disponibilidade e a integridade dos dados em todos os momentos.

- Usabilidade: As interfaces devem ser intuitivas e de fácil utilização, permitindo que os usuários realizem suas tarefas de forma eficiente e sem dificuldades.

- Eficiência: O sistema deve ter um desempenho eficiente, realizando operações de forma rápida e consumindo recursos adequados.

- Segurança: Devem ser aplicadas medidas de segurança robustas para proteger os dados e garantir a privacidade dos usuários.

- Manutenibilidade: O sistema deve ser facilmente mantido e atualizado, permitindo a incorporação de novas funcionalidades e correção de eventuais problemas de forma ágil.

2.4 Objetos/Classes

2.4.1 Modelo Conceitual/Classes de Análise/Modelo de Domínio

(Classes, Associações, nomes das associações,

Multiplicidades e Atributos):

@startuml

left to right direction

skinparam packageStyle rect

entity Cliente

entity "Departamento de Vendas" as DepartamentoVendas

entity Producao

entity "Almoxarifado e Estocagem" as AlmoxarifadoEstocagem

entity "Setor de Recebimento e Distribuição" as SetorRecebimentoDistribuicao

entity Transportadora

entity "Departamento Financeiro" as DepartamentoFinanceiro

entity "Gerenciador de qualidade" as GerenciadorQualidade

entity "Pedido" as Pedido

entity "Item de Pedido" as ItemPedido

entity "Entrega" as Entrega

Cliente"1" --|> "\*" Pedido : Realiza

Pedido "1" --|> "\*" ItemPedido : Contém

Pedido --|> DepartamentoVendas : Enviado para

Pedido --|> Producao : Requisitado para

Pedido --|> Entrega : Associado a

Entrega "1" --|> "\*" SetorRecebimentoDistribuicao : Recebido por

Entrega --|> Transportadora : Entregue por

Pedido --|> DepartamentoFinanceiro : Pagamento associado

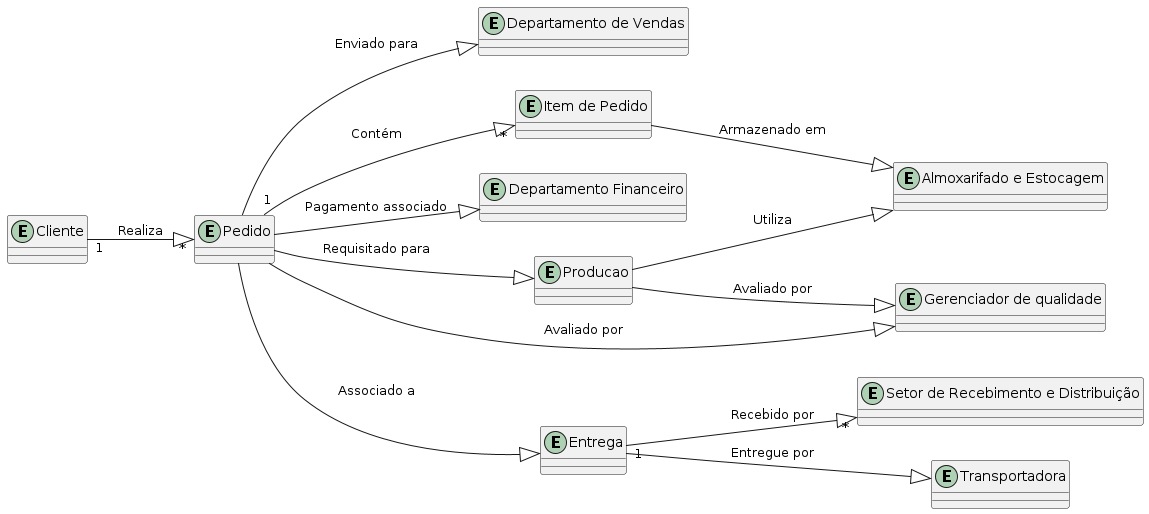
Pedido --|> GerenciadorQualidade : Avaliado por

ItemPedido --|> AlmoxarifadoEstocagem : Armazenado em

Producao --|> AlmoxarifadoEstocagem : Utiliza

Producao --|> GerenciadorQualidade : Avaliado por

@enduml



2.4.2 Eventos e Operações:

Eventos:

1. Pedido registrado

2. Pedido cancelado

3. Pagamento recebido

4. Produção iniciada

5. Produto entregue

Operações:

1. Registrar Pedido:

- Descrição: Realiza o registro de um novo pedido no sistema.

- Atores envolvidos: Cliente, Departamento de Vendas

- Fluxo de eventos:

1. O Cliente informa os detalhes do pedido.

2. O Departamento de Vendas verifica e registra o pedido no sistema.

3. O sistema gera um número de identificação para o pedido.

2. Cancelar Pedido:

- Descrição: Cancela um pedido previamente registrado.

- Atores envolvidos: Cliente, Departamento de Vendas

- Fluxo de eventos:

1. O Cliente solicita o cancelamento do pedido.

2. O Departamento de Vendas verifica a viabilidade do cancelamento.

3. Se o cancelamento for viável, o pedido é marcado como cancelado no sistema.

3. Receber Pagamento:

- Descrição: Registra o recebimento de pagamento de um pedido.

- Atores envolvidos: Cliente, Departamento Financeiro

- Fluxo de eventos:

1. O Cliente realiza o pagamento do pedido.

2. O Departamento Financeiro registra o recebimento do pagamento no sistema.

4. Iniciar Produção:

- Descrição: Inicia a produção de um pedido.

- Atores envolvidos: Departamento de Vendas, Produção

- Fluxo de eventos:

1. O Departamento de Vendas verifica a disponibilidade dos recursos para a produção do pedido.

2. O Departamento de Vendas agenda a produção do pedido.

3. A Produção inicia a produção do pedido.

5. Entregar Produto:

- Descrição: Registra a entrega de um produto ao cliente.

- Atores envolvidos: Setor de Recebimento e Distribuição, Cliente

- Fluxo de eventos:

1. O Setor de Recebimento e Distribuição prepara o produto para entrega.

2. O Setor de Recebimento e Distribuição realiza a entrega do produto ao Cliente.

3. O Cliente confirma o recebimento do produto no sistema.

2.4.3 DSS – Diagramas de Sequência do Sistema, Contratos

@startuml

actor Cliente

actor "Departamento de Vendas" as dv

actor Produção

actor "Setor de Recebimento e Distribuição" as srd

actor "Departamento Financeiro" as df

Cliente -> dv: RegistrarPedido()

dv -> Produção: EscalonarPedido()

Produção -> Produção: ControlarProdução()

Produção -> srd: GerenciarEstoque()

srd -> srd: GerenciarEstoque()

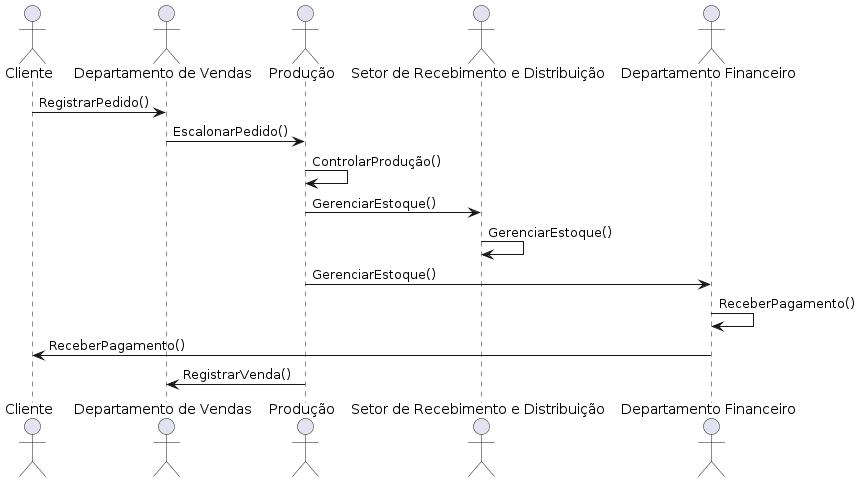
Produção -> df: GerenciarEstoque()

df -> df: ReceberPagamento()

df -> Cliente: ReceberPagamento()

Produção -> dv: RegistrarVenda()

@enduml



2.4.4 Classes de Implementação - Diagrama de Classes

(Classes, Associações, nomes das associações,

Multiplicidades, Atributos e Métodos)

@startuml

class Cliente {

- id: int

- nome: String

- endereco: String

- email: String

+ fazerPedido(): void

+ cancelarPedido(): void

}

class Pedido {

- id: int

- data: Date

- status: String

+ adicionarItem(item: Item): void

+ removerItem(item: Item): void

}

class Item {

- id: int

- descricao: String

- quantidade: int

- preco: double

}

class Venda {

- id: int

- data: Date

- valorTotal: double

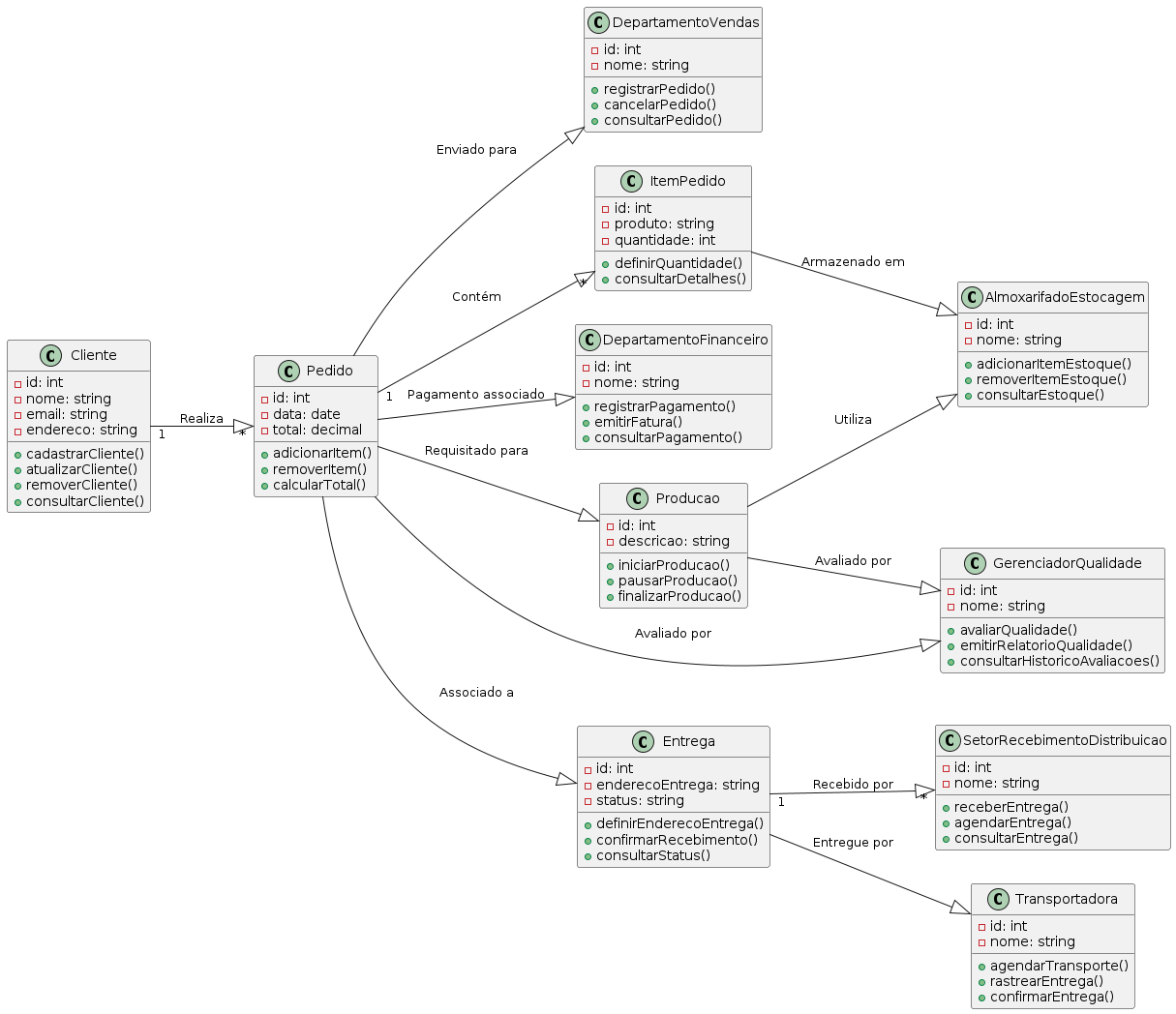
}

Cliente "1" -- "0..\*" Pedido

Pedido "1" -- "0..\*" Item

Pedido "1" -- "1" Venda

@enduml



3 Análise de UCP

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