**Engenharia de Software**

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

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

Título

Fábrica de concreto

Equipe (engenheiro e desenvolvedores):

**Willian Binda**

Introdução

## Resumo do Projeto (Descrição textual)

Estudo de caso baseado no artigo de modelagem de sistemas industriais,

Palazzo e Edelweizz (1998), apud Lutherer E. et al., 1994.

O estudo representa uma fábrica de concreto que inclui vendas, escritório e a planta industrial.

A planta é composta de vários receptáculos de armazenagem de cimento, areia, cascalho e água;

um misturador para preparação do concreto; duas balanças para pesar os componentes;

uma válvula de medição de componentes; controle de descarga de água;

duas esteiras de transporte para levar os ingredientes sólidos ao misturador.

Os clientes preenchem uma ordem de pedido entregue pelo representante com o tipo e quantidade de concreto

para entrega em um ou mais dias pela transportadora da empresa. O departamento de vendas escalona a ordem

e eventualmente encaminha a indústria para o encarregado de produção. A indústria providência a execução

do pedido, através da área de produção e de matéria prima. A produção executa a ordem solicitando materiais

que são entregues no local pelo encarregado do almoxarifado e estocagem. O operador das máquinas executa

os controles de fabricação tendo como auxiliares os operários responsáveis pela mistura dos componentes

e pelo funcionário encarregado do controle de fluxo de trabalho. Caso algum produto da matéria prima fique

abaixo do nível de segurança, o encarregado da produção emite ordem para o departamento de compras

providenciar o pedido da mercadoria que após a entrega é verificado pelo setor de recebimento e

distribuição se está entregue conforme foi feito o pedido. Após a fabricação, o produto é entregue na

construção e o cliente efetua o pagamento que é recebido pelo departamento financeiro da indústria.

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

Editor de código: Visual Studio Code;

Banco de dados: Postgresql;

Linguagem de programação: C#;

Frontend: Django;

Backend: API REST em python (Django REST)

Ferramentas adicionais:

Docker;

Postman;

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

Desktop: Windows;

## 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)

### Back-End-Front-End.

### 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 ...)

O sistema deve ser capaz de realizar as seguintes funcionalidades:

1.6.1 Controle de Produção

Registrar e acompanhar as ordens de produção, desde o recebimento até a entrega do concreto.

Monitorar o progresso da produção e o status de cada ordem de produção.

Manter um registro histórico das ordens de produção.

1.6.2 Atendimento de Pedidos

Permitir que os clientes preencham uma ordem de pedido com informações detalhadas, como tipo e quantidade de concreto desejado.

Escalonar as ordens de pedido recebidas pelo departamento de vendas.

Encaminhar as ordens de pedido para o encarregado de produção na planta industrial.

1.6.3 Controle de Matéria Prima

Controlar os estoques de cimento, areia, cascalho e água nos receptáculos de armazenagem.

Emitir ordens de compra para reposição de matéria prima quando o nível de segurança for atingido.

Verificar a entrega da matéria prima e realizar a conferência com o pedido efetuado.

1.6.4 Controle de Entrega de Concreto

Agendar a entrega do concreto de acordo com as solicitações dos clientes.

Coordenar a logística de transporte e roteamento para a entrega dos pedidos.

Registrar as informações de entrega e confirmação do cliente.

1.6.5 Controle de Compras de Matéria Prima

Emitir ordens de compra para aquisição de matéria prima quando necessário.

Acompanhar o status das ordens de compra e notificar sobre eventuais atrasos ou problemas.

1.6.6 Considerações Finais

O sistema proposto visa proporcionar um controle eficiente da produção,

atendimento aos pedidos, gestão de matéria prima, controle de entrega de concreto e compras de matéria prima.

Através do uso adequado do sistema, espera-se otimizar os processos e melhorar a eficiência operacional da fábrica de concreto.

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

Gerente de compras, engenheiros civis, Construtoras, PJ, Pf.

## 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)

Serviço hospedado em um servidor na nuvem Arch linux, 4 GB RAM, 1TB armazenamento interno com expansão para 2TB.

# 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 projIncConc\_usecase

left to right direction

actor Cliente as Cl

actor Transportadora as Tr

package Fábrica\_de\_concreto {

usecase "Preencher ordem de pedido" as UC1

usecase "Escalonar ordem" as UC2

usecase "Enviar ordem para produção" as UC3

usecase "Solicitar materiais" as UC4

usecase "Entregar materiais" as UC5

usecase "Executar pedido" as UC6

usecase "Controlar fabricação" as UC7

usecase "Misturar componentes" as UC8

usecase "Emitir ordem de compra de produtos" as UC9

usecase "Verificar recebimento" as UC10

usecase "Entregar produto" as UC11

usecase "Efetuar pagamento" as UC12

}

Cl -- UC1

UC2 -- Departamento\_vendas

UC3 -- Departamento\_vendas

UC4 -- Encarregado\_produção

UC5 -- Encarregado\_almoxarifado

UC6 -- Encarregado\_produção

UC7 -- Operador\_maquinas

UC8 -- Operarios

UC9 -- Encarregado\_produção

UC10 -- Departamento\_recebimentos\_distribuicao

Cl -- UC12

Tr -- UC11

UC12 --> Departamento\_financeiro

UC2 ..> UC1 : <<Include>>

UC3 ..> UC2 : <<Include>>

UC6 ..> UC5 : <<Include>>

UC5 ..> UC4 : <<Include>>

UC7 ..> UC6 : <<Include>>

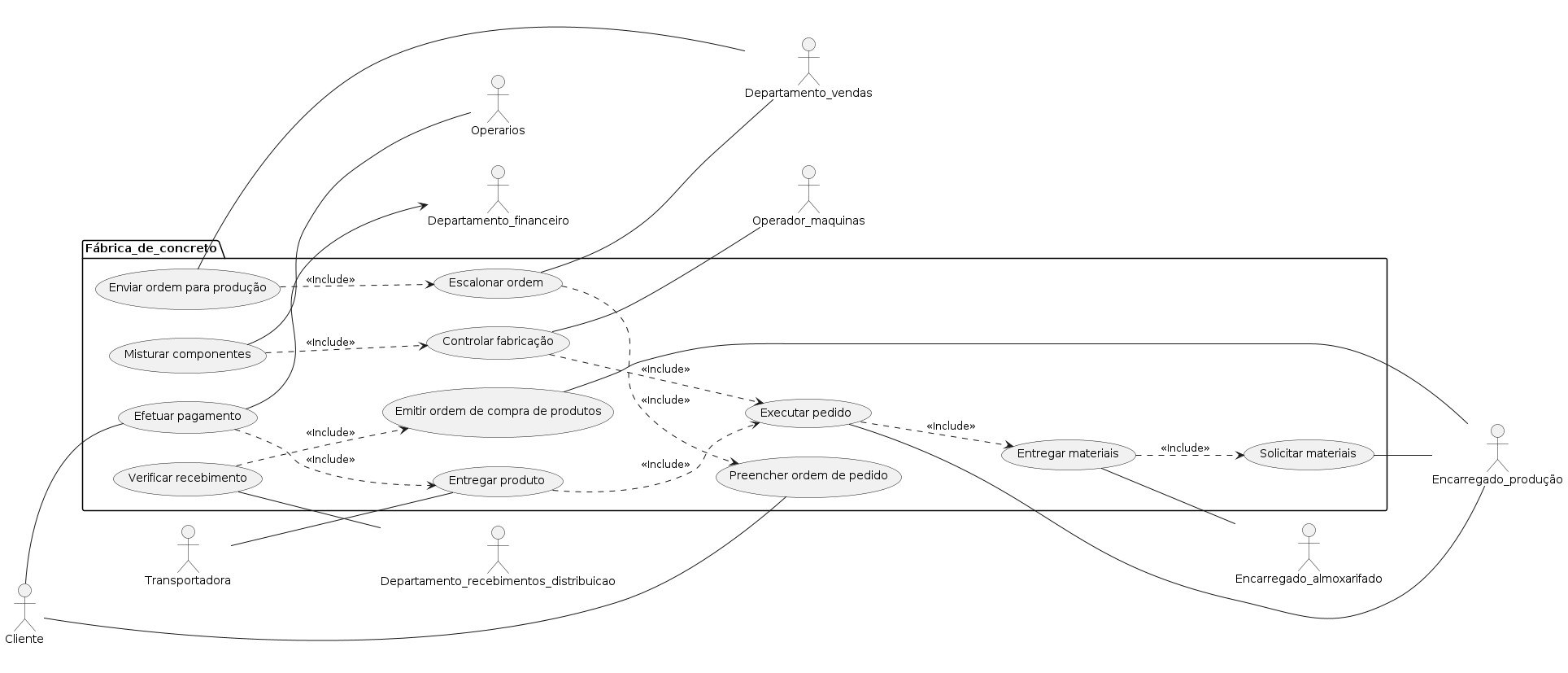
UC8 ..> UC7 : <<Include>>

UC10 ..> UC9 : <<Include>>

UC11 ..> UC6 : <<Include>>

UC12 ..> UC11 : <<Include>>

@enduml



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

Casos de uso expandidos:

2.2.2.1 - Caso de Uso: Preencher ordem de pedido

Ator principal: Cliente

Resumo: Este caso de uso descreve o processo de preenchimento de

uma ordem de pedido pelo cliente.

Pré-condições: O cliente está registrado no sistema.

Fluxo principal:

2.2.2.2 - O caso de uso começa quando o cliente inicia o preenchimento

de uma ordem de pedido.

2.2.2.3 - O sistema exibe um formulário para o cliente preencher as

informações do pedido,

incluindo o tipo de concreto e a quantidade desejada.

2.2.2.4 - O cliente preenche as informações necessárias no formulário.

2.2.2.5 - O sistema valida as informações do pedido.

2.2.2.6 - O sistema registra o pedido no sistema, associando-o

ao cliente.

2.2.2.7 - O sistema exibe uma confirmação de que o pedido foi

registrado com sucesso.

Pós-condições: O pedido é registrado no sistema.

2.2.3 - Caso de Uso: Entregar produto

Ator principal: Transportadora

Resumo: Este caso de uso descreve o processo de entrega do produto

ao cliente pela transportadora.

Pré-condições: O pedido foi produzido e está pronto para entrega.

Fluxo principal:

2.2.3.1 - O caso de uso começa quando a transportadora recebe uma

solicitação de entrega de um pedido.

2.2.3.2 - A transportadora verifica as informações do pedido, incluindo

o endereço de entrega e os detalhes do cliente.

2.2.3.3 - A transportadora planeja a rota de entrega com base nas

informações do pedido.

2.2.3.4 - A transportadora coleta o produto na fábrica de concreto.

2.2.3.5 - A transportadora entrega o produto ao cliente no endereço

especificado.

2.2.3.6 - O cliente confirma o recebimento do produto.

2.2.3.7 - A transportadora registra a entrega no sistema.

2.2.3.8 - O sistema atualiza o status do pedido para "entregue".

Fluxo alternativo:

2.2.3.9 - Se o produto não estiver disponível na fábrica de concreto:

2.2.3.10- A transportadora notifica a fábrica sobre a falta de produto.

2.2.3.11 - A fábrica verifica a disponibilidade do produto.

2.2.3.12 - Se o produto estiver disponível, a fábrica prepara o pedido e o

entrega à transportadora.

2.2.3.13 - A transportadora continua o processo de entrega normalmente.

Pós-condições: O produto é entregue ao cliente e o status do pedido é

atualizado no sistema.

-----------------------------------------------------------------------------------------------------------------------

| Caso de Uso: Preencher ordem de pedido |

-----------------------------------------------------------------------------------------------------------------------

| Ator Principal: Cliente |

-----------------------------------------------------------------------------------------------------------------------

| Resumo: Este caso de uso descreve o processo de |

| preenchimento de uma |

| ordem de pedido pelo cliente. |

------------------------------------------------------------------------------------------------------------------------

| Pré-condições: O cliente está registrado no sistema. |

-----------------------------------------------------------------------------------------------------------------------

| Fluxo Principal: |

-----------------------------------------------------------------------------------------------------------------------

| Passo | Ação do Ator | Resposta do Sistema |

------------------------------------------------------------------------------------------------------------------------

| 2.2.2.2 | Iniciar o preenchimento | |

| | de uma ordem de pedido | |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.2.3 | | Exibir um formulário para |

| | | o cliente preencher as |

| | | informações do pedido, |

| | | incluindo o tipo de |

| | | concreto e a quantidade |

| | | desejada. |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.2.4 | Preencher as informações | |

| | necessárias no formulário | |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.2.5 | | Validar as informações do pedido |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.2.6 | | Registrar o pedido no sistema, |

| | | associando-o ao cliente |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.2.7 | | Exibir uma confirmação de que o |

| | | pedido foi registrado com sucesso |

-----------------------------------------------------------------------------------------------------------------------

| Pós-condições: O pedido é registrado no sistema. |

-----------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------

| Caso de Uso: Entregar produto |

-----------------------------------------------------------------------------------------------------------------------

| Ator Principal: Transportadora |

-----------------------------------------------------------------------------------------------------------------------

| Resumo: Este caso de uso descreve o processo de |

| entrega do produto ao |

| cliente pela transportadora. |

-----------------------------------------------------------------------------------------------------------------------

| Pré-condições: O pedido foi produzido e está pronto |

| para entrega. |

-----------------------------------------------------------------------------------------------------------------------

| Fluxo Principal: |

-----------------------------------------------------------------------------------------------------------------------

| Passo | Ação do Ator | Resposta do Sistema |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.1 | Receber uma solicitação | |

| |de entrega de um pedido | |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.2 | | Verificar as informações do |

| | | pedido, incluindo o endereço |

| | | de entrega e os detalhes do |

| | | cliente |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.3 | | Planejar a rota de entrega com |

| | | base nas informações do pedido |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.4 | Coletar o produto na | |

| | fábrica de concreto | |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.5 | Entregar o produto ao | |

| | cliente no endereço | |

| | especificado | |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.6 | Confirmar o recebimento do | |

| | produto pelo cliente | |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.7 | | Registrar a entrega no sistema |

-----------------------------------------------------------------------------------------------------------------------

| 2.2.3.8 | | Atualizar o status do pedido |

| | | para "entregue" |

-----------------------------------------------------------------------------------------------------------------------

| Fluxo Alternativo: |

------------------------------------------------------------------------------------------------------------------------

| 2.2.3.9 | Se o produto não estiver | |

| | disponível na fábrica | |

-----------------------------------------------------------------------------------------------------------------------

|2.2.3.10 | | Notificar a fábrica sobre a |

| | | falta de produto |

-----------------------------------------------------------------------------------------------------------------------

|2.2.3.11 | | Verificar a disponibilidade |

| | | do produto na fábrica |

-----------------------------------------------------------------------------------------------------------------------

|2.2.3.12 | | Se o produto estiver disponível, |

| | | a fábrica prepara o pedido e o |

| | | entrega à transportadora |

-----------------------------------------------------------------------------------------------------------------------

|2.2.3.13 | | A transportadora continua o |

| | | processo de entrega normalmente |

-----------------------------------------------------------------------------------------------------------------------

| Pós-condições: O produto é entregue ao cliente e o |

| status do |

| pedido é atualizado no sistema. |

-----------------------------------------------------------------------------------------------------------------------

Diagrama de atividades:

@startuml

|Cliente|

start

:Iniciar Preenchimento de Pedido;

if (Cliente está registrado?) then (Sim)

:Exibir Formulário de Pedido;

else (Não)

:Registrar Cliente;

:Exibir Formulário de Pedido;

endif

while (Cliente preenche informações do pedido) is (Sim)

:Preencher Informações do Pedido;

endwhile (Não)

:Validar Informações do Pedido;

if (Informações válidas?) then (Sim)

:Registrar Pedido;

else (Não)

:Exibir Mensagem de Erro;

stop

endif

|Transportadora|

:Receber Solicitação de Entrega;

:Verificar Informações do Pedido;

if (Produto disponível na fábrica?) then (Sim)

:Planejar Rota de Entrega;

:Coletar Produto;

:Entregar Produto;

:Confirmar Recebimento do Cliente;

:Registrar Entrega;

else (Não)

:Notificar Fábrica sobre Falta de Produto;

:Verificar Disponibilidade do Produto;

if (Produto disponível?) then (Sim)

:Preparar Pedido;

:Coletar Produto;

:Entregar Produto;

:Confirmar Recebimento do Cliente;

:Registrar Entrega;

else (Não)

:Exibir Mensagem de Erro;

stop

endif

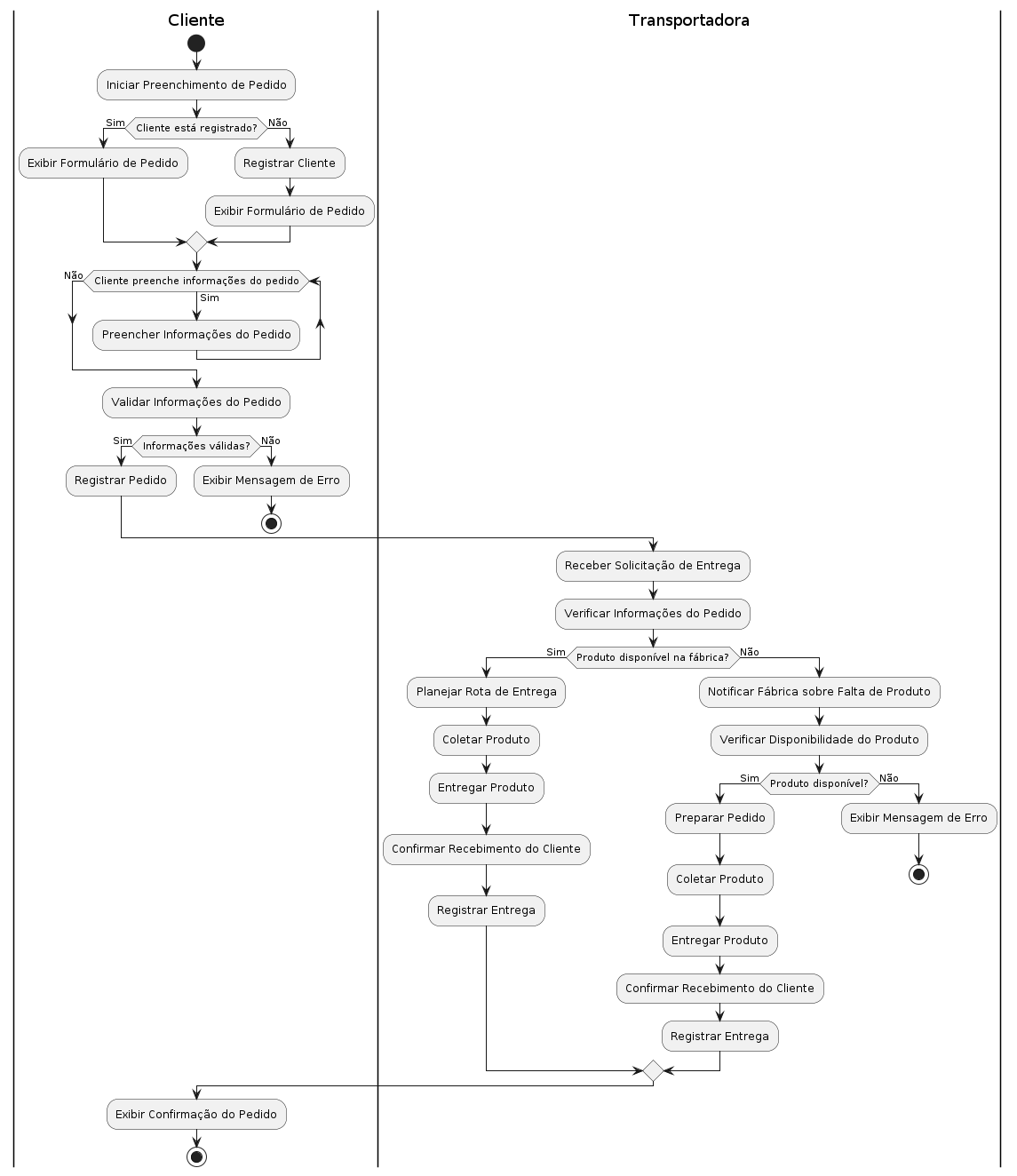
endif

|Cliente|

:Exibir Confirmação do Pedido;

stop

@enduml



## Requisitos não-funcionais

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

Como tempo de resposta para gerar relatórios como por exemplo relatórios de venda ou de quantidade de materiais, etc. Deve-se manter uma média de 0.5 a 3 segundos. Conforme a quantidade de dados.

### 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 projIncConc\_class

class Cliente {

nome

endereco

telefone

}

class OrdemDePedido {

tipoConcreto

quantidade

status

dataPedido

dataEntrega

}

class Transportadora {

nome

endereco

telefone

}

class DepartamentoVendas {

nome

responsavel

ordensDePedido

}

class Encarregado {

nome

cargo

}

class DepartamentoCompras {

nome

responsavel

ordensDeCompra

}

class OperadorMaquinas {

nome

maquinas

}

class DepartamentoRecebimentos {

nome

responsavel

produtosRecebidos

}

class Operario {

nome

cargo

pagamentos

}

class Planta {

tipo

equipamentos

armazenagens

}

class Produto {

nome

estoque

comprasMateriaPrima

}

class CompraMateriaPrima {

produto

quantidade

}

Cliente "1" -- "\*" OrdemDePedido : realiza

OrdemDePedido "1" -- "1" Transportadora : entrega

DepartamentoVendas "1" -- "\*" OrdemDePedido : escalona

DepartamentoVendas "1" -- "1" Encarregado : envia para

DepartamentoCompras "1" -- "\*" OrdemDeCompra : realiza

DepartamentoCompras "1" -- "1" Encarregado : envia pedido

OperadorMaquinas "1" -- "\*" Maquina : utiliza

DepartamentoRecebimentos "1" -- "\*" Produto : recebe

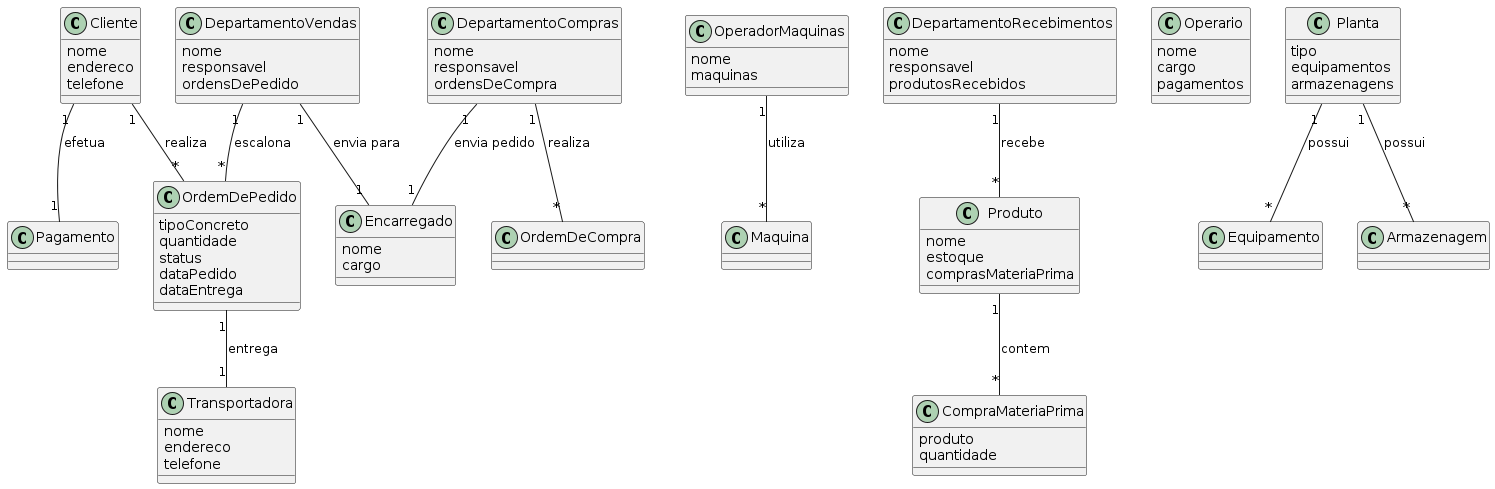
Cliente "1" -- "1" Pagamento : efetua

Planta "1" -- "\*" Equipamento : possui

Planta "1" -- "\*" Armazenagem : possui

Produto "1" -- "\*" CompraMateriaPrima : contem

@enduml



### Eventos e Operações

### DSS – Diagramas de Sequência do Sistema, Contratos

Cliente:

@startuml Cliente

actor Cliente

participant "Sistema" as sistema

activate Cliente

Cliente -> sistema: iniciarPreenchimentoPedido()

sistema -> Cliente: exibirFormularioPedido()

Cliente -> sistema: preencherInformacoesPedido(tipoConcreto, quantidade)

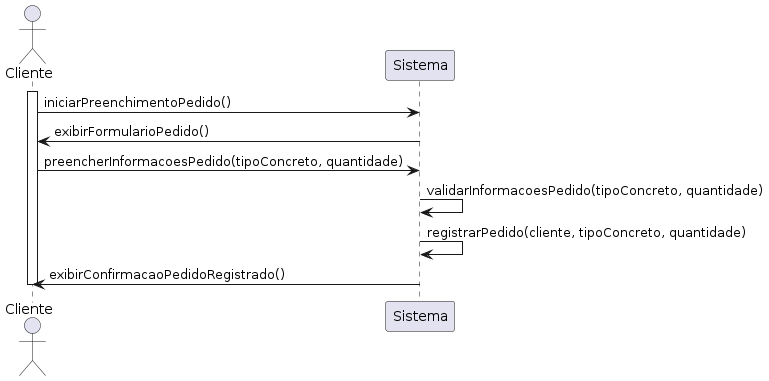
sistema -> sistema: validarInformacoesPedido(tipoConcreto, quantidade)

sistema -> sistema: registrarPedido(cliente, tipoConcreto, quantidade)

sistema -> Cliente: exibirConfirmacaoPedidoRegistrado()

deactivate Cliente

@enduml



Nome: registrarPedido(cliente: Cliente, tipoConcreto: String, quantidade: int)

Responsabilidade: Registrar um novo pedido no sistema com as informações fornecidas pelo cliente. Tipo: Sistema

Referências cruzadas: Funções do sistema: R2.2.2.5 R2.2.2.6

Caso de Uso: Comprar Itens

Notas: Utilize acesso super-rápido ao banco de dados para registrar o pedido.

Exceções: Se as informações do pedido forem inválidas, indique o erro.

Saída: Pré-condições: Nenhuma

Pós-condições:

* Um novo objeto OrdemDePedido foi criado (c.i.).
* O atributo cliente da OrdemDePedido recebeu o valor do parâmetro cliente (m.a.).
* O atributo tipoConcreto da OrdemDePedido recebeu o valor do parâmetro tipoConcreto (m.a.).
* O atributo quantidade da OrdemDePedido recebeu o valor do parâmetro quantidade (m.a.).
* O status da OrdemDePedido foi definido como "Registrado" (m.a.).
* A data do pedido foi definida como a data atual (m.a.).

Entregador:

@startuml Entregador

actor Transportadora

participant "Sistema" as sistema

participant "Fábrica de Concreto" as fabrica

participant Cliente

activate Transportadora

Transportadora -> sistema: receberSolicitacaoEntrega(pedido)

sistema -> Transportadora: verificarInformacoesPedido(pedido)

alt Produto Disponível na Fábrica

Transportadora -> sistema: planejarRotaEntrega(pedido)

Transportadora -> fabrica: coletarProduto(pedido)

Transportadora -> Cliente: entregarProduto(pedido)

Cliente -> Transportadora: confirmarRecebimento(pedido)

Transportadora -> sistema: registrarEntrega(pedido)

else Produto Indisponível na Fábrica

Transportadora -> fabrica: notificarFaltaProduto(pedido)

fabrica -> fabrica: verificarDisponibilidadeProduto(pedido)

alt Produto Disponível na Fábrica

fabrica -> fabrica: prepararPedido(pedido)

fabrica -> Transportadora: coletarProduto(pedido)

Transportadora -> Cliente: entregarProduto(pedido)

Cliente -> Transportadora: confirmarRecebimento(pedido)

Transportadora -> sistema: registrarEntrega(pedido)

else Produto Indisponível na Fábrica

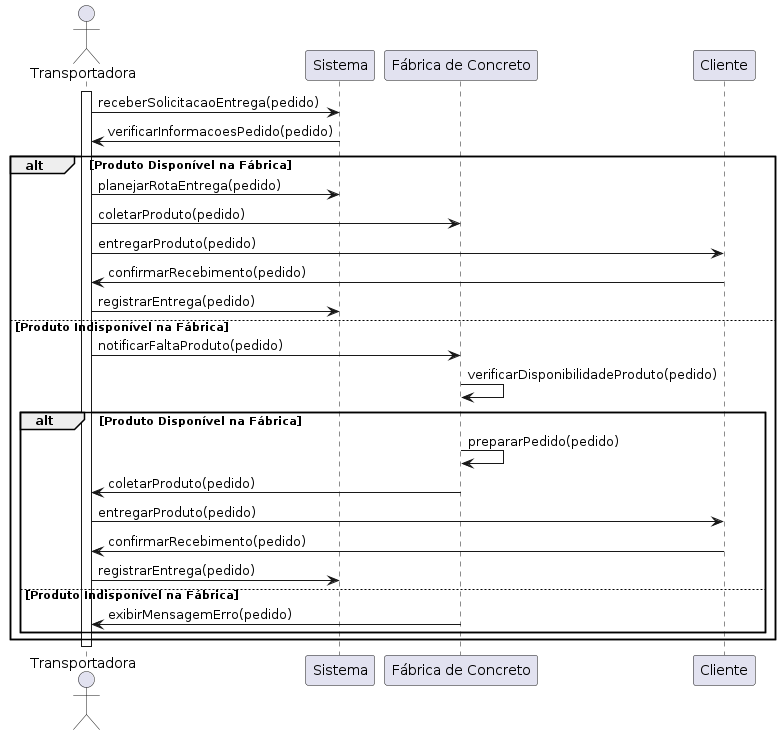
fabrica -> Transportadora: exibirMensagemErro(pedido)

end

end

deactivate Transportadora

@enduml



Contrato: receberSolicitacaoEntrega(pedido)

Responsabilidade: Receber a solicitação de entrega de um pedido.

Tipo: Ator principal (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.1

Pré-condições: Nenhuma

Pós-condições: Nenhuma

Contrato: verificarInformacoesPedido(pedido)

Responsabilidade: Verificar as informações do pedido, incluindo o endereço de entrega e os detalhes do cliente.

Tipo: Sistema (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.2

Pré-condições: Pedido existe e possui informações válidas

Pós-condições: Nenhuma

Contrato: planejarRotaEntrega(pedido)

Responsabilidade: Planejar a rota de entrega com base nas informações do pedido.

Tipo: Sistema (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.3

Pré-condições: Informações do pedido estão completas e válidas

Pós-condições: Nenhuma

Contrato: coletarProduto(pedido)

Responsabilidade: Coletar o produto na fábrica de concreto.

Tipo: Sistema (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.4

Pré-condições: Produto está disponível na fábrica de concreto

Pós-condições: Nenhuma

Contrato: entregarProduto(pedido)

Responsabilidade: Entregar o produto ao cliente no endereço especificado.

Tipo: Ator principal (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.5

Pré-condições: Produto foi coletado e endereço de entrega é válido

Pós-condições: Nenhuma

Contrato: confirmarRecebimento(pedido)

Responsabilidade: Confirmar o recebimento do produto pelo cliente.

Tipo: Ator secundário (Cliente)

Referências cruzadas: Caso de Uso 2.2.3.6

Pré-condições: Cliente recebeu o produto e confirma o recebimento

Pós-condições: Nenhuma

Contrato: registrarEntrega(pedido)

Responsabilidade: Registrar a entrega no sistema.

Tipo: Sistema (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.7

Pré-condições: Entrega foi concluída com sucesso

Pós-condições: Status do pedido é atualizado para "entregue"

Contrato: exibirMensagemErro(pedido)

Responsabilidade: Exibir uma mensagem de erro informando a falta de produto na fábrica de concreto. Tipo: Sistema (Fábrica de Concreto)

Referências cruzadas: Caso de Uso 2.2.3.9

Pré-condições: Produto não está disponível na fábrica de concreto

Pós-condições: Nenhuma

Contrato: notificarFaltaProduto(pedido)

Responsabilidade: Notificar a fábrica sobre a falta de produto.

Tipo: Sistema (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.10

Pré-condições: Produto não está disponível na fábrica de concreto

Pós-condições: Nenhuma

Contrato: verificarDisponibilidadeProduto(pedido)

Responsabilidade: Verificar a disponibilidade do produto na fábrica de concreto.

Tipo: Sistema (Fábrica de Concreto)

Referências cruzadas: Caso de Uso 2.2.3.11

Pré-condições: Pedido foi recebido e está aguardando disponibilidade do produto

Pós-condições: Nenhuma

Contrato: prepararPedido(pedido)

Responsabilidade: Preparar o pedido na fábrica de concreto e entregá-lo à transportadora.

Tipo: Sistema (Fábrica de Concreto)

Referências cruzadas: Caso de Uso 2.2.3.12

Pré-condições: Produto está disponível na fábrica de concreto

Pós-condições: Nenhuma

Contrato: atualizarStatusPedido(pedido)

Responsabilidade: Atualizar o status do pedido para "entregue" no sistema.

Tipo: Sistema (Transportadora)

Referências cruzadas: Caso de Uso 2.2.3.8

Pré-condições: Entrega foi registrada com sucesso

Pós-condições: Status do pedido é atualizado para "entregue"

### Classes de Implementação - Diagrama de Classes (Classes, Associações, nomes das associações, Multiplicidades, Atributos e Métodos)

@startuml projIncConc\_class

class Cliente {

- nome: String

- endereco: String

- telefone: String

+ preencherPedido(tipoConcreto: String, quantidade: int): void

}

class OrdemDePedido {

- tipoConcreto: String

- quantidade: int

- status: String

- dataPedido: Date

- dataEntrega: Date

+ getStatus(): String

+ setDataEntrega(data: Date): void

}

class Transportadora {

- nome: String

- endereco: String

- telefone: String

+ entregarPedido(pedido: OrdemDePedido): void

}

class DepartamentoVendas {

- nome: String

- responsavel: String

- ordensDePedido: List<OrdemDePedido>

+ escalonarPedido(pedido: OrdemDePedido): void

+ enviarPedidoProducao(pedido: OrdemDePedido): void

}

class Encarregado {

- nome: String

- cargo: String

+ solicitarMateriais(): void

+ emitirOrdemCompra(produto: Produto, quantidade: int): void

}

class DepartamentoCompras {

- nome: String

- responsavel: String

- ordensDeCompra: List<OrdemDeCompra>

+ realizarCompra(produto: Produto, quantidade: int): void

+ enviarPedido(produto: Produto, quantidade: int): void

}

class OperadorMaquinas {

- nome: String

- maquinas: List<Maquina>

+ controlarFabricacao(): void

}

class DepartamentoRecebimentos {

- nome: String

- responsavel: String

- produtosRecebidos: List<Produto>

+ receberProduto(produto: Produto): void

}

class Operario {

- nome: String

- cargo: String

- pagamentos: List<Pagamento>

+ executarTrabalho(): void

}

class Planta {

- tipo: String

- equipamentos: List<Equipamento>

- armazenagens: List<Armazenagem>

+ adicionarEquipamento(equipamento: Equipamento): void

+ adicionarArmazenagem(armazenagem: Armazenagem): void

}

class Produto {

- nome: String

- estoque: int

- comprasMateriaPrima: List<OrdemDeCompra>

+ realizarCompra(quantidade: int): void

+ verificarEstoque(): int

}

class OrdemDeCompra {

- produto: Produto

- quantidade: int

+ getProduto(): Produto

+ getQuantidade(): int

}

class Equipamento {

- nome: String

- tipo: String

+ realizarOperacao(): void

}

class Armazenagem {

- nome: String

- tipo: String

- capacidade: int

+ verificarNivelEstoque(): int

}

class Pagamento {

- valor: double

- data: Date

+ efetuarPagamento(): void

}

Cliente "1" -- "\*" OrdemDePedido : realiza

Transportadora "1" -- "\*" OrdemDePedido : entrega

DepartamentoVendas "1" -- "\*" OrdemDePedido : escalona

DepartamentoVendas "1" -- "1" Encarregado : envia para

DepartamentoCompras "1" -- "\*" OrdemDeCompra : realiza

OperadorMaquinas "1" -- "\*" Maquina : controla

DepartamentoRecebimentos "1" -- "\*" Produto : recebe

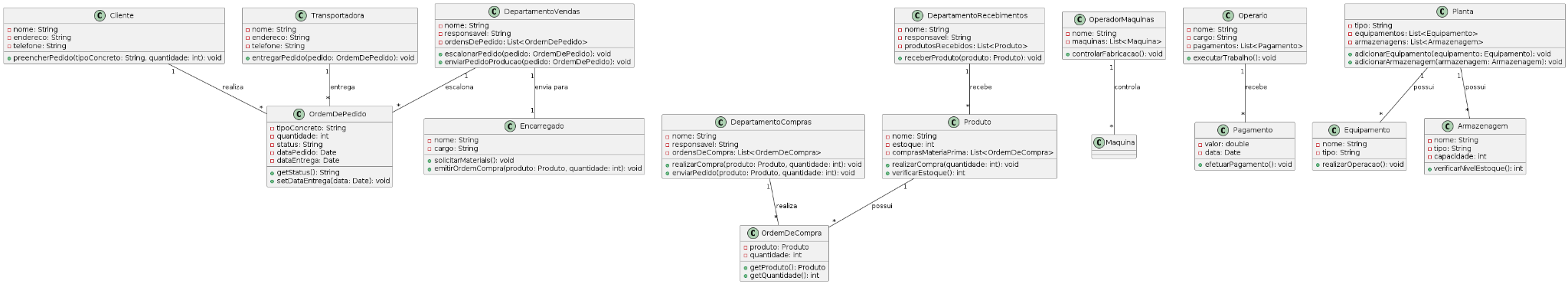
Operario "1" -- "\*" Pagamento : recebe

Planta "1" -- "\*" Equipamento : possui

Planta "1" -- "\*" Armazenagem : possui

Produto "1" -- "\*" OrdemDeCompra : possui

@enduml



# 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.**