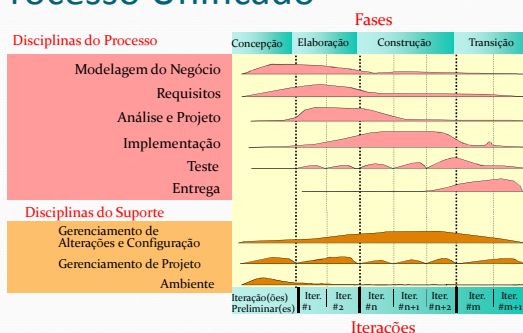


Processo Unificado : Análise e Projeto

Karin Becker
Engenharia de Software N
Instituto de Informática - UFRGS

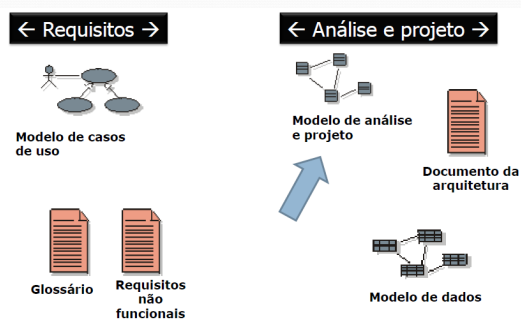
Processo Unificado



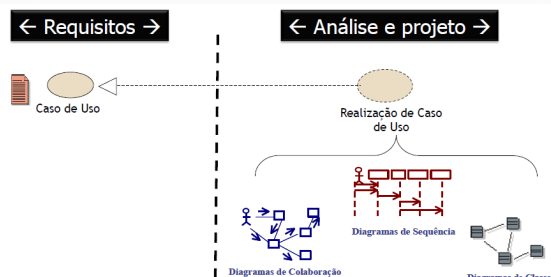
Análise e Projeto

- Objetivos básicos
 - Criar um projeto a partir dos requisitos identificados
 - Derivar uma arquitetura para o sistema
 - Adaptar o projeto para as limitações do ambiente de execução
- Centrado em Realização de Casos de Uso
- Pode ser dividido em dois modelos
 - Modelo de Análise (transitório)
 - Modelo de Projeto (permanente)

Visão Geral dos Artefatos



Realização de Casos de Uso

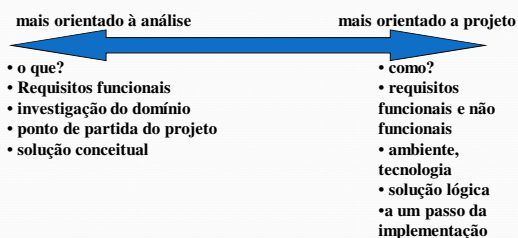


Análise vs Projeto

- Análise: a investigação mais detalhada do problema e esboço de solução
 - melhor compreensão dos objetos do **domínio do problema** e dos requisitos funcionais a serem atendidos pelo sistema
 - O QUE?
- Projeto: desenvolvimento de solução lógica que atende os requisitos funcionais e não funcionais
 - COMO?
 - Requisitos funcionais e não funcionais
 - restrições de ambiente e tecnologia

Análise e Projeto

- **Fronteira mal definida !!!!!**



Modelos de Análise e Projeto

Modelo de Análise

- **TEMPORÁRIO**
- Define uma estrutura que funciona como **entrada fundamental ao processo de projeto**
- **Menos Esforço** (1/5 da disciplina)
- **Modelo conceitual** (uma abstração que evita detalhes de implementação)
- **Genérico** (múltiplas opções de projeto e tecnologia)
- **Menos formal**
- **Divisão de responsabilidades inicial**
- **Não foca muito em coordenação de interações**
- **Esboço do projeto** e de sua arquitetura

Modelo de Projeto

- Idealmente, **mantido** ao longo do ciclo de vida
- **Modelo "físico"** ("planta baixa") que funciona como entrada fundamental para implementação
- **Mais esforço** (4/5 da disciplina)
- **Específico** para uma implementação
- **Mais formal**
- **Não foca muito em coordenação de interações**
- **Divisão de responsabilidades mais detalhada (refinamento)**
- **Detalha o projeto** e sua arquitetura
- **Define a estrutura do sistema**, mantendo ou não contrapartida no modelo de análise
- **Abstração** da implementação

Modelagem : Lembre-se !!!!

- Objetivos
 - ENTENDER
 - COMUNICAR
- Modelos/diagramas não são necessariamente
 - Documentos
 - Permanentes
 - Organização/processo definem
 - Processo unificado é centrado na comunicação via modelos documentados

Modelo de Análise (Modelo Conceitual)

Profª. Karin Becker
Engenharia de Software N
Instituto de Informática - UFRGS

Modelo de Análise vs. Modelo de Casos de Uso

Modelo de Casos de uso

- Descrito na linguagem do **cliente**
- **Visão externa** do sistema
- Estruturada por **casos de uso** que estruturam a **visão externa**
- estabelece contrato entre cliente e desenvolvedores sobre **o quê** o sistema deve fazer
- **Pode** conter redundâncias, inconsistências, etc. entre requisitos
- **Captura** a funcionalidade do sistema
- **Define os casos de uso** que são endereçados no modelo de análise

Modelo de Análise

- Descrito usando a linguagem do **desenvolvedor**
- **Visão interna** do sistema
- Estruturado por **classes** estereotipadas e **pacotes** que estruturam a **visão interna**
- Auxilia desenvolvedores a compreender a **forma inicial do sistema**
- **Serve para reduzir** inconsistências ou ambiguidades
- **Esboça como concretizar** a funcionalidade no sistema
- Define **realizações de casos de uso**

Modelo de Análise

- Classes de Análise
- Realização de Casos de Uso – Análise
 - Diagrama de Classes (**de análise**)
 - Diagrama de Interação
 - colaboração entre classes decorrente de um evento gerado por um ator
 - Delineia fluxo de eventos no sistema
 - Pacotes de serviços

O que são classes de análise?

- Representam o conceito mais abstrato dos elementos do sistema
 - Modelo Conceitual
- Modelo de classes transitório
 - **Ponto de partida**
 - Serão convertidas para **classes de projeto**
 - **Transição**
 - Diminuem a distância entre os requisitos e o projeto
 - Favorecem a rastreabilidade e a **justificativa** de modelagem

Análise - Modelo Conceitual

- modelo conceitual
 - ilustra os conceitos significativos (para os modeladores) em um domínio do problema
 - representação do domínio, e não dos componentes de software
 - decomposição do espaço do problema em unidades compreensíveis (conceitos)
 - compreensão do vocabulário do domínio, comunicando às partes interessadas os termos relevantes e como eles estão relacionados
 - **Na prática, um modelo conceitual não é correto ou incorreto: ele é mais ou menos útil**

Descobrimos Classes de Análise

- Utilize os casos de uso expandidos como fonte de referência
- Estratégias:
 - relacione os conceitos candidatos do domínio do problema a pertencer a lista de categorias de conceitos (objetos físicos, transações, linhas de itens de transações, papéis desempenhados por pessoas, contêineres de coisas, eventos, etc.).
 - identificar os substantivos e frases que podem estar no lugar de um substantivo nas descrições do domínio do problema e considerá-los como candidatos a conceitos ou atributos para o modelo conceitual.

Exemplo: Caso de Uso Comprar Itens

Identificação: UC1
 Caso de uso: Registrar Venda
 Atores: **Atendente**
 Pré-Condições: o Atendente está logado no **terminal**
 Pós-Condições: a **venda** é registrada, o **estoque** do **produtos** vendidos é atualizado, o **pagamento** é registrado e o **recibo** do **cliente** é impresso.

Sequência Típica de Eventos (Fluxo Básico):

1. O **Atendente** inicia uma nova **venda**
2. Para cada **produto**
 1. O Atendente entra com a identificação e quantidade de itens do produto
 2. O sistema registra a venda do produto e exibe no **terminal** sua descrição, preço e o valor total da venda de acordo com a quantidade de itens
3. O Atendente informa o término da venda
4. O sistema exibe o valor total da venda
5. O Atendente registra o pagamento recebido
6. O sistema registra a venda como completa, e imprime um recibo contendo dados da venda e do terminal onde esta foi efetuada, junto com quem o operava.

Descobrendo Classes de Análise

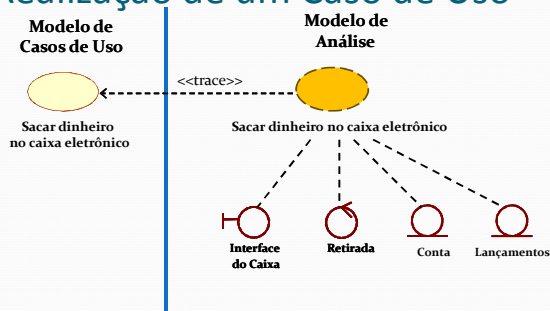
- estratégias para encontrar conceitos (iniciais) -
 - objetos físicos ou tangíveis (Terminal de Venda)
 - especificações de projetos ou descrições de coisas (Especificação de Produto)
 - lugares (loja)
 - transações e elementos de transações (Venda, Item Vendido)
 - papéis desempenhados por pessoas (Gerente, Caixa)
 - containers de coisas e coisas em containers (depósito, Item)
 - dispositivos, sistemas externos (Autorização Pagamento Cartão)
 - organizações, eventos (Departamento de Vendas, Balanço)
 - processo (frequentemente não representados) - (reserva Produto)
 - regras e estratégias (política de Reembolso)
 - registros financeiros, trabalhistas, etc (recibo, contrato)
 - serviços, instrumentos financeiros (linha de Crédito)
 - etc

Craig Larman

RUP : Modelo de Análise

- Objetivo:
 - tem como foco a representação dos requisitos funcionais do sistema (representação de conceitos do domínio de problema)
- RUP é centrado na realização de casos de usos
 - Incentiva a busca das classes envolvidas na realização de cada caso de uso
- Características
 - Classes
 - definem atributos, porém sem o compromisso com detalhamento
 - raramente definem suas interfaces em termos de operações e assinaturas
 - "responsabilidades"
 - estabelecem suas relações com outras classes
 - "isolamento de preocupações" (separation of concerns) padrão
 - Classes com papéis de abstrair aspectos específicos

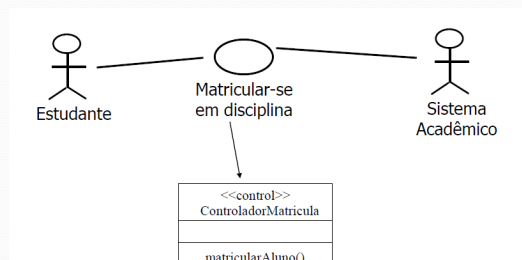
Realização de um Caso de Uso



Realização de Caso de Uso

- Descreve como o caso de uso é realizado, associando o caso de uso com classes e outros elementos de projeto
- Utiliza diagramas:
 - Classes (de análise): para organizar classes em um nível de detalhe relevante para a análise.
 - Interação: para distribuir comportamento

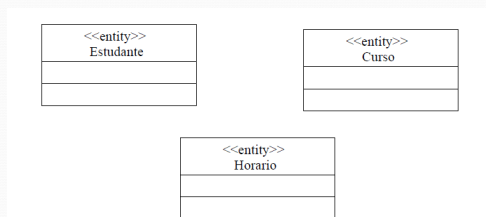
Classe de Controle



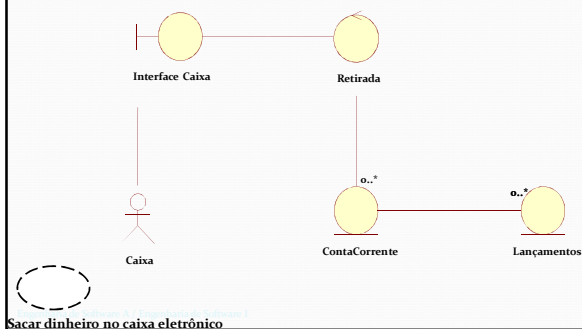
Estereótipos de Classes de Análise

- Classe Entidade (entity)
- modela informações e comportamentos associados a um conceito ou fenômeno do sistema;
- utilizada para abstrair informações relacionadas que têm longa vida no sistema
 - frequentemente (mas não necessariamente) são persistentes
 - pode ter um comportamento complexo relacionado à informação que representa;
- Frequentemente mostra uma estrutura lógica de dados e contribui à compreensão de qual informação o sistema depende.

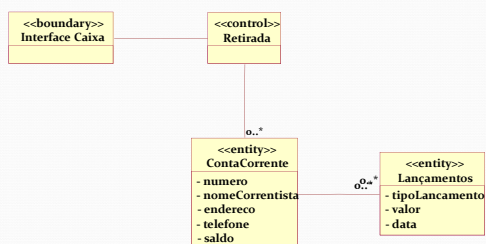
Classes Entidade



Estereótipos de Classes de Análise



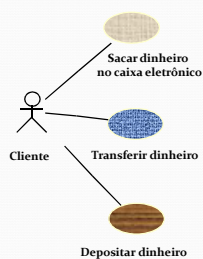
Estereótipos de Classes de Análise



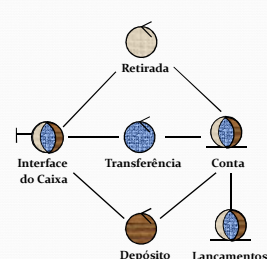
Sacar dinheiro no caixa eletrônico

Classes de Análise

Modelo de Casos de Uso



Modelo de Análise



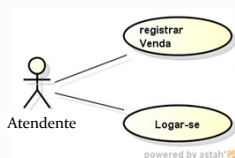
Diretrizes

- Atores interagem apenas com objetos de fronteira (boundary)
- Objetos de fronteira podem interagir com atores e controladores.
- Objetos de entidade interagem apenas com controladores
- Controladores interagem com qualquer outro tipo de objeto e também com outros controladores, mas nunca com atores.

Refinando Classes de Análise

- Identificação dos atributos
- Identificação de associações
- Identificar generalizações
- Identificação das responsabilidades, a partir dos papéis desempenhados nos casos de uso

Exemplo



Exemplo: Caso de Uso Log In

Identificação: UC1

Caso de uso: Logar-se

Atores: Atendente

Pré-Condições:

Pós-Condições: é registrada a operação de um terminal por um atendente identificado.

Sequência Típica de Eventos (Fluxo Básico):

1. Este caso de uso começa quando o Atendente fornece sua identificação (login e senha) ao terminal
2. O sistema habilita o registro de vendas naquele terminal, registrando data e hora do início da operação do terminal pelo Atendente.

Exemplo: Caso de Uso Comprar Itens

Identificação: UC2

Caso de uso: Registrar Venda

Atores: Atendente

Pré-Condições: o Atendente está logado no terminal

Pós-Condições: a venda é registrada, o estoque do produtos vendidos é atualizado, o pagamento é registrado e o recibo do cliente é impresso.

Sequência Típica de Eventos (Fluxo Básico):

1. O Atendente inicia uma nova venda
2. Para cada produto
 1. O Atendente entra com a identificação e quantidade de itens do produto
 2. O sistema registra a venda do produto e exibe no terminal sua descrição, preço e o valor total da venda de acordo com a quantidade de itens
3. O Atendente informa o término da venda
4. O sistema exibe o valor total da venda
5. O Atendente registra o pagamento recebido
6. O sistema registra a venda como completa, e imprime um recibo contendo dados da venda e do terminal onde esta foi efetuada, junto com que o operava.

Classes de Análise - Fronteira



Atendente

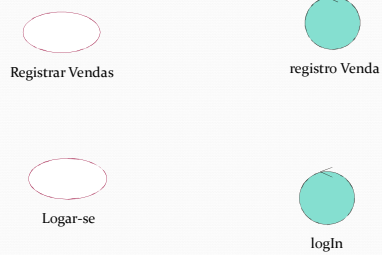


GUI Autenticação



GUI Vendas

Classes de Análise - Controle



Exemplo: Caso de Uso Registrar vendas

Identificação: UC2

Caso de uso: Registrar vendas

Atores: Atendente

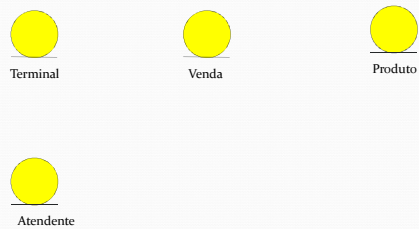
Pré-Condições: o Atendente está logado no terminal

Pós-Condições: a venda é registrada, o estoque do produtos vendidos é atualizado, o pagamento é registrado e o recibo do cliente é impresso.

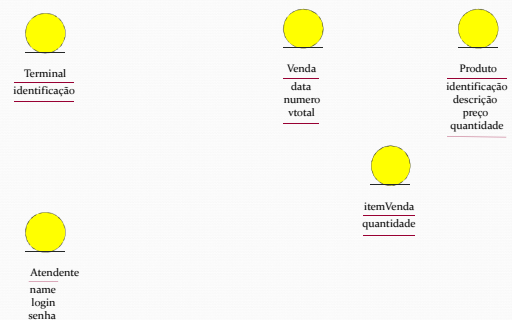
Sequência Típica de Eventos (Fluxo Básico):

1. O **Atendente** inicia uma nova **venda**
2. Para cada **produto**
 1. O Caixa entra com a identificação e quantidade de itens do produto
 2. O sistema registra a venda do produto e apresenta a sua descrição, preço e o valor total da venda de acordo com a quantidade de **itens**.
3. O Atendente informa o término da venda
4. O sistema apresenta o valor total da venda
5. O Atendente registra o pagamento recebido
6. O sistema registra a venda como completa, e imprime um recibo contendo dados da venda e do **terminal** onde esta foi efetuada, junto com que o operava.

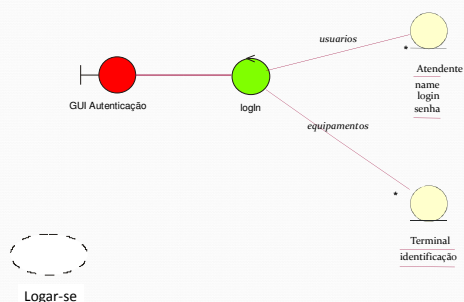
Classes de Análise - Entidades



Classes de Análise - Refinamento



Classes de Análise - Refinamento



Exemplo: Caso de Uso Logar-se

Identificação: UC2

Caso de uso: Logar-se

Atores: Atendente

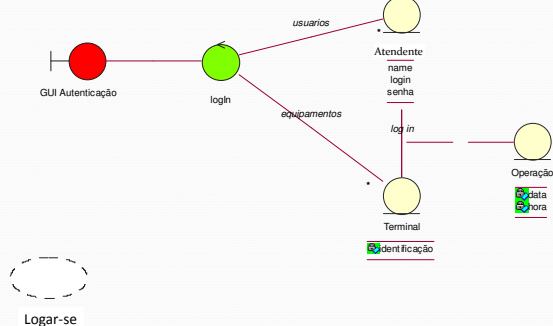
Pré-Condições:

Pós-Condições: é registrada a operação de um terminal por um Atendente identificado.

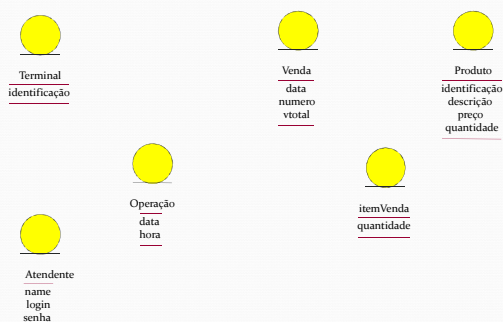
Sequência Típica de Eventos (Fluxo Básico):

1. Este caso de uso começa quando o **Atendente** fornece sua identificação (login e senha) ao **terminal**.
2. O sistema habilita o registro de vendas naquele terminal, registrando data e hora do início da **operação** do terminal pelo Atendente.

Classes de Análise - Refinamento



Classes de Análise - Refinamento



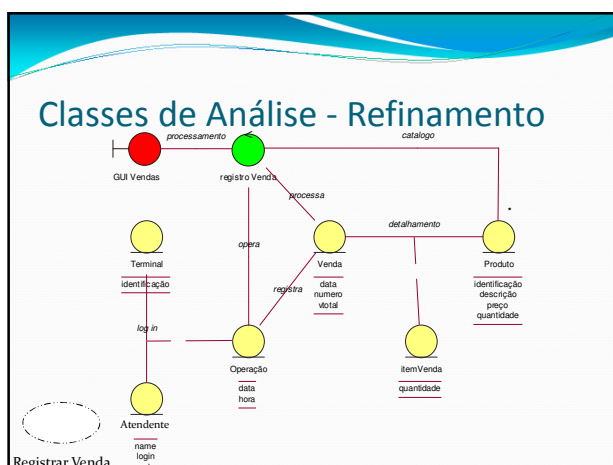
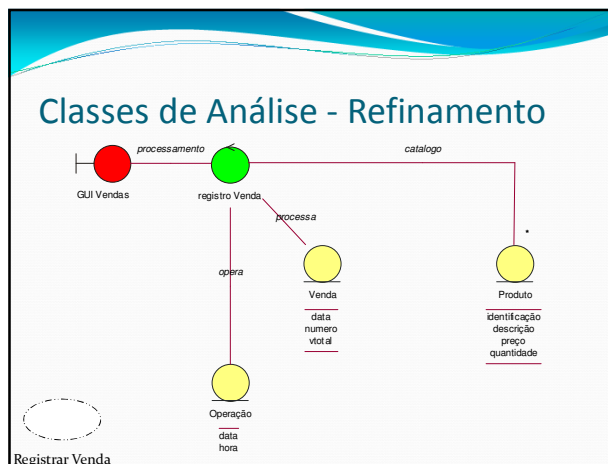
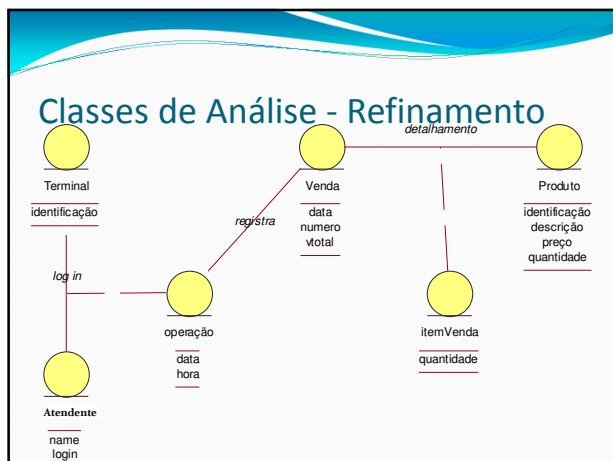
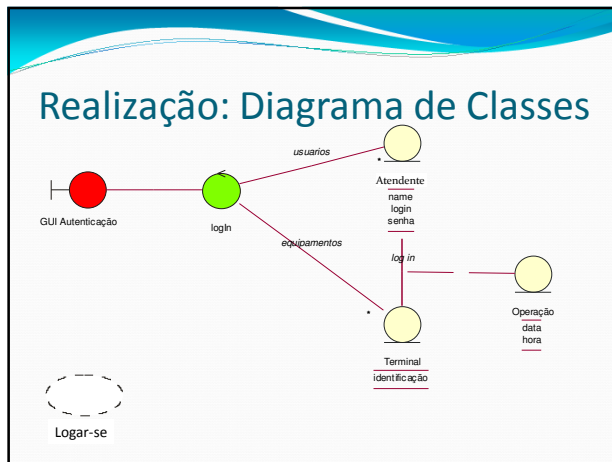


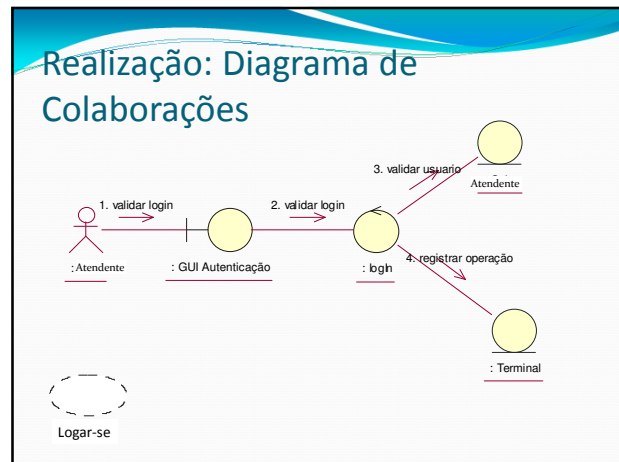
Diagrama de Classes

- Diagramas de Classes são utilizados para **modelagem estática**. A modelagem estática deve dar suporte às necessidades funcionais do sistema, isto é, os serviços que o sistema deve providenciar aos seus usuários finais.
- Os Diagramas de Classes são utilizados para modelar:
 - o vocabulário do sistema**: especificação das abstrações que estão contidos dentro do domínio do sistema, identificando suas responsabilidades.
 - colaboração**: colaboração envolve trabalho conjunto entre objetos do sistema visando um comportamento cooperativo. Esta cooperação traduz-se no diagrama de classes através das relações existente entre as classes identificadas.
 - esquema lógico do banco de dados** do sistema.

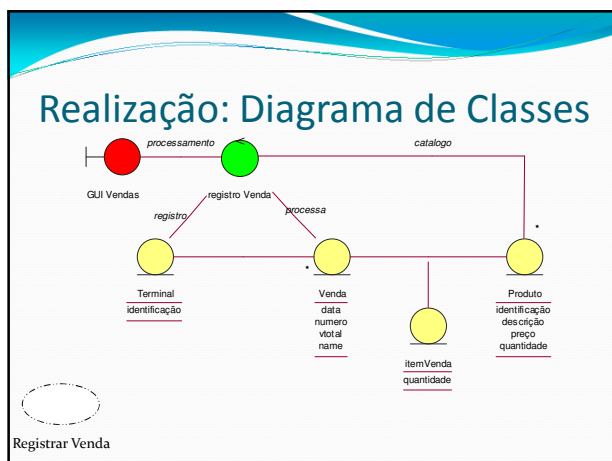
Realização: Diagrama de Classes



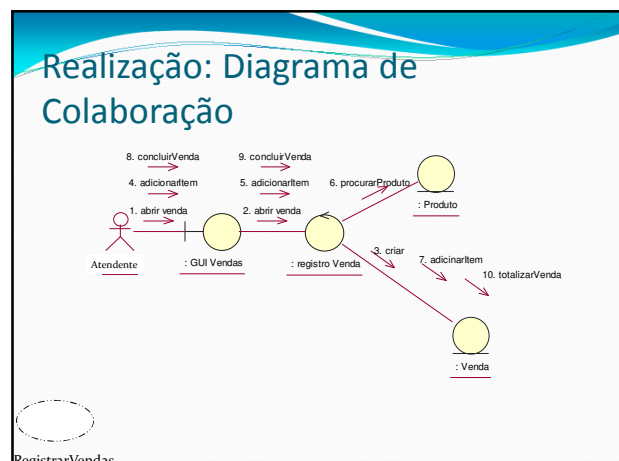
Realização: Diagrama de Colaborações



Realização: Diagrama de Classes



Realização: Diagrama de Colaboração



Modelo de Análise: Considerações

- **Temporário:** Um modelo conceitual preliminar de como o sistema funcionará
 - Evolui rapidamente na fase de projeto
 - Permite estudar implicações de diferentes alternativas de projeto
 - Permite entender com mais detalhes os requisitos do sistema e o domínio
- Lembre-se
 - Modelo de análise tipicamente não é certo ou errado: ele é mais ou menos útil
- Classes de análise evoluem durante projeto
 - "Proto-classes" representando agrupamento de comportamento
 - Dão origem a novas classes (ou conjuntos de classes), ou mesmo a subsistemas
 - Frequentemente representam colaborações de múltiplos objetos
 - Permitem explorar distribuições alternativas de responsabilidades para atingir uma boa separação de preocupações

Classe de Análise → Classe de Projeto

- Possíveis transformações
 - Uma classe de análise pode vir a ser uma única classe no modelo de design.
 - Uma classe de análise pode fazer parte de uma classe no modelo de design.
 - Uma classe de análise pode vir a ser uma classe agregada no modelo de design (ou seja, as partes dessa classe agregada podem não estar definidas explicitamente no modelo de análise).
 - Uma classe de análise pode vir a ser um grupo de classes que herda da mesma classe no modelo de design.
 - Uma classe de análise pode vir a ser um grupo de classes relacionadas funcionalmente no modelo de design.
 - Uma classe de análise pode vir a ser um pacote no modelo de design (ou seja, pode tornar-se um componente).
 - Uma classe de análise pode vir a ser um relacionamento no modelo de design.
 - Um relacionamento entre as classes de análise pode vir a ser uma classe no modelo de design.
 - As classes de análise lidam principalmente com requisitos funcionais e objetos de modelo provenientes do domínio "problema". As classes de design lidam com requisitos não-funcionais e objetos de modelo provenientes do domínio "solução".
 - As classes de análise podem ser usadas para representar "os objetos aos quais desejamos que o sistema dê suporte", sem que você precise definir quantas suportarão o hardware e quantas suportarão o software. Assim, parte de uma classe de análise pode ser usada pelo hardware e não estar definida no modelo de design.

<http://www.wthreex.com/rup/portugues/index.htm>

Modelo de Análise : Considerações

- Modelo de análise
 - Um esboço inicial do sistema
 - Orientado a domínio
- Abstração do projeto
 - Simplificação
 - Independente de plataforma ou tecnologia
 - Compreensão aprofundada dos requisitos (perguntas, muitas perguntas)
- Temporário
 - O trabalho de manter o modelo de projeto consistente com o modelo de análise deve ser cuidadosamente avaliado
 - Benefício de ter uma visão de alto nível que abstrai os principais detalhes de um sistema
 - Ex: sistemas legados

RUP: Modelo de Análise é opcional

- Quando um modelo de análise é útil?
 - ainda não se conhece o domínio (inexperiência no desenvolvimento de aplicações no domínio)
 - o sistema deve ser projetado para vários ambientes alvo, com arquiteturas de projeto distintas
 - Modelo de Análise é uma abstração do Modelo de Projeto
 - Ele omite a maioria dos detalhes do design para fornecer uma visão geral da funcionalidade do sistema
 - Permite estudar implicações de diferentes alternativas de projeto
 - o design é tão complexo que é preciso um "design" abstrato simplificado para introduzir o design aos novos membros da equipe
 - a aplicação é complexa: o processo de investigação do domínio é extenso

Para saber mais ...

- **Fortemente recomendada**
 - Craig Larman. Utilizando UML e Padrões. Bookman.
(2d edição mais enxuta, 3ed mais completa)
 - Capítulos 9-11 da edição 2
 - Capítulos 9 da edição 3