

Especificação de Requisitos e Análise Orientada o Objetos

- Exemplo -

Terminal de Ponto de Vendas (PDV)

- Um sistema computadorizado que registra vendas e trata de pagamentos
- Tipicamente usado numa loja de varejo
- Tipicamente está acoplado a um leitor de código de barra

Arquitetura em Camadas

- Um sistema de informação típico é organizado usando as camadas apresentadas abaixo

Apresentação



The screenshot shows a window titled 'Object Store'. It contains several input fields and buttons. On the left, there are three vertically stacked input fields labeled 'UPC', 'Total', and 'Tendered'. On the right, there are two vertically stacked input fields labeled 'Quantity' and 'Balance'. At the bottom, there are three buttons: 'Enter Item', 'End Sale', and 'Make Payment'.

Lógica de aplicação

objetos do domínio do problema

Venda

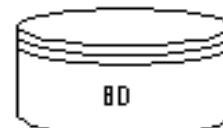
Pagamento

objetos de serviço

Acesso a BD

Gerente de Segurança

Armazenamento



Arquitetura em Camadas

- Apresentação
 - Interface gráfica, janelas
- Lógica de aplicação - Objetos do domínio de problema
 - Objetos representando conceitos do domínio (business objects)
 - Contém o que se chama também business logic
- Lógica de aplicação - Objetos de serviço
 - Objetos que não pertencem ao domínio do problema mas oferecem serviços tais como interfaceamento para um banco de dados, etc.
- Armazenamento
 - Mecanismos de armazenamento persistente (SGBDOO, SGBDR, SGBDOR)

Requisitos

- Breve descrição do sistema
- Descrição dos clientes alvo
- Descrição das metas do sistema
- Descrição dos requisitos funcionais do sistema
- Descrição dos requisitos não funcionais do sistema

Requisitos do Terminal PDV

- Breve descrição do sistema
 - O objetivo do projeto é de criar um sistema para um Terminal Ponto de Venda (TPDV) a ser usado no comércio varejista.
- Descrição dos clientes alvo
 - O cliente é Xpto Ltda., que vende TPDVs a lojas varejistas.
- Descrição das metas do sistema
 - A meta básica é de melhorar a automação do balcão de vendas, incluindo:
 - Checkout mais rápido para o cliente
 - Análise rápida e precisa das vendas
 - Automatizar o controle de estoque

Requisitos Funcionais do Terminal PDV

- Funcionalidades associados à Venda
 - R1.1. Registrar a venda corrente (os itens comprados).
 - R1.2. Calcular o total de venda, incluindo impostos e descontos aplicáveis.
 - R1.3. Capturar a informação do item sendo comprado através de um leitor de código de barra, ou manualmente usando um código de produto tal como o Universal Product Code (UPC).
 - R1.4. Dar baixa no inventário ao terminar uma venda
 - R1.5. Manter um log de vendas feitas
 - R1.6. O caixa deve fazer login com uma identificação e uma senha antes de usar o sistema.
 - R1.7. Exibir a descrição e o preço do item sob consideração.

Requisitos Funcionais do Terminal PDV

- Funcionalidades associados ao Pagamento
 - R2.1. Tratar de pagamentos à vista com dinheiro, capturando o valor entregue e calculando o troco.
 - R2.2. Tratar de pagamentos por cartão de crédito, capturando a informação do cartão através de um leitor de cartões ou por digitação manual e autorizando o pagamento usando o serviço de autorização de crédito da loja (um sistema externo) usando uma conexão via modem.
 - R2.3. Tratar de pagamentos por cheque, capturando a informação de identidade/CPF por digitação manual e autorizando o pagamento usando o serviço de verificação de cheques da loja (um sistema externo) usando uma conexão via modem.
 - R2.4 Lançar os pagamentos via cartão de crédito no sistema de contas a receber, já que o serviço de cartão de crédito deve dinheiro à loja.

Requisitos Não Funcionais do Terminal PDV

- Tempo de resposta
 - Ao registrar um item sendo vendido, a descrição e preço devem aparecer em 2 segundos.
- Tipo de interface
 - Usar formulários para entrada de dados e dialog boxes.
 - Maximizar a facilidade de uso via teclado e não via mouse
- Tolerância a falhas
 - Deve fazer log dos pagamentos autorizados via cartão de crédito em 24 horas, mesmo com falhas de energia ou de dispositivo.
- Plataformas operacionais
 - Microsoft Windows XP, Vista e 7

Requisitos do Terminal PDV

- A lista de requisitos apresentada nos slides anteriores não é exaustiva, servem apenas como exemplo.

Cenários de Uso

- A especificação dos requisitos funcionais deve ser completada com casos de uso (use cases) ou histórias do usuário (User Stories).
 - São cenários de uso do sistema

Casos de Uso (Use Cases)

- São uma excelente forma de explorar e documentar os requisitos funcionais.
- Antes de elaborar use cases, pode valer a pena elaborar uma lista de funções básicas como vimos nos slides anteriores.
 - Servem para nortear o levantamento de casos de uso.
 - Mas alguns desenvolvedores usam apenas casos de uso para levantar os requisitos funcionais.

Casos de Uso (Use Cases)

- Um caso de uso é um documento narrativo que descreve uma sequência de eventos feitos por um ator no uso de um sistema para completar um processo de interesse deste ator.
 - Casos de uso são "estórias" ou "casos" no uso de um sistema
 - As estórias acabam revelando as funcionalidade desejada do sistema

Caso de Uso: Comprar Item com Dinheiro

Caso de Uso: Comprar item com dinheiro

Atores: Cliente (iniciador), Caixa

Propósito: Capturar uma venda e seu pagamento em dinheiro.

Resumo: Um cliente chega ao caixa com itens a comprar. O caixa registra os itens comprados e recebe pagamento. No fim, o cliente sai com os itens comprados.

Referência cruzada: R1.1, R1.2, R1.3, R1.7, R2.1

(pode fazer referência a outros Casos de Uso)

Caso de Uso: Comprar Item com Dinheiro

Sequência típica de eventos

Ação do ator	Resposta do Sistema
1. O Caso de Uso inicia quando um cliente chega a um caixa com itens a comprar.	
2. O caixa registra a identificação de cada item. Se houver mais itens, o caixa pode informar a quantidade também	3. Determina o preço do item e adiciona a informação ao total da transação de venda. A descrição e preço do item corrente são exibidos
4. Ao completar a entrada dos itens, o caixa indica este fato ao TPDV.	5. Calcula e apresenta o total da venda
6. O caixa informa o total da venda ao cliente.	
7. O cliente efetua o pagamento com dinheiro, possivelmente maior que o total da venda.	
8. O caixa registra a quantidade de dinheiro recebida.	9. Mostra o valor do troco ao cliente. Gera um recibo impresso.
10. O caixa deposita o dinheiro recebido e extrai o troco a devolver. O caixa entrega o troco e o recibo impresso ao cliente.	11. Faz log da venda completada.
12. O cliente sai da loja com os itens comprados.	

Sequências alternativas:

Linha 2: Entrada de um identificador inválido. Indica erro.

Linha 7: Cliente não tinha dinheiro suficiente. Cancela transação de venda.

Análise

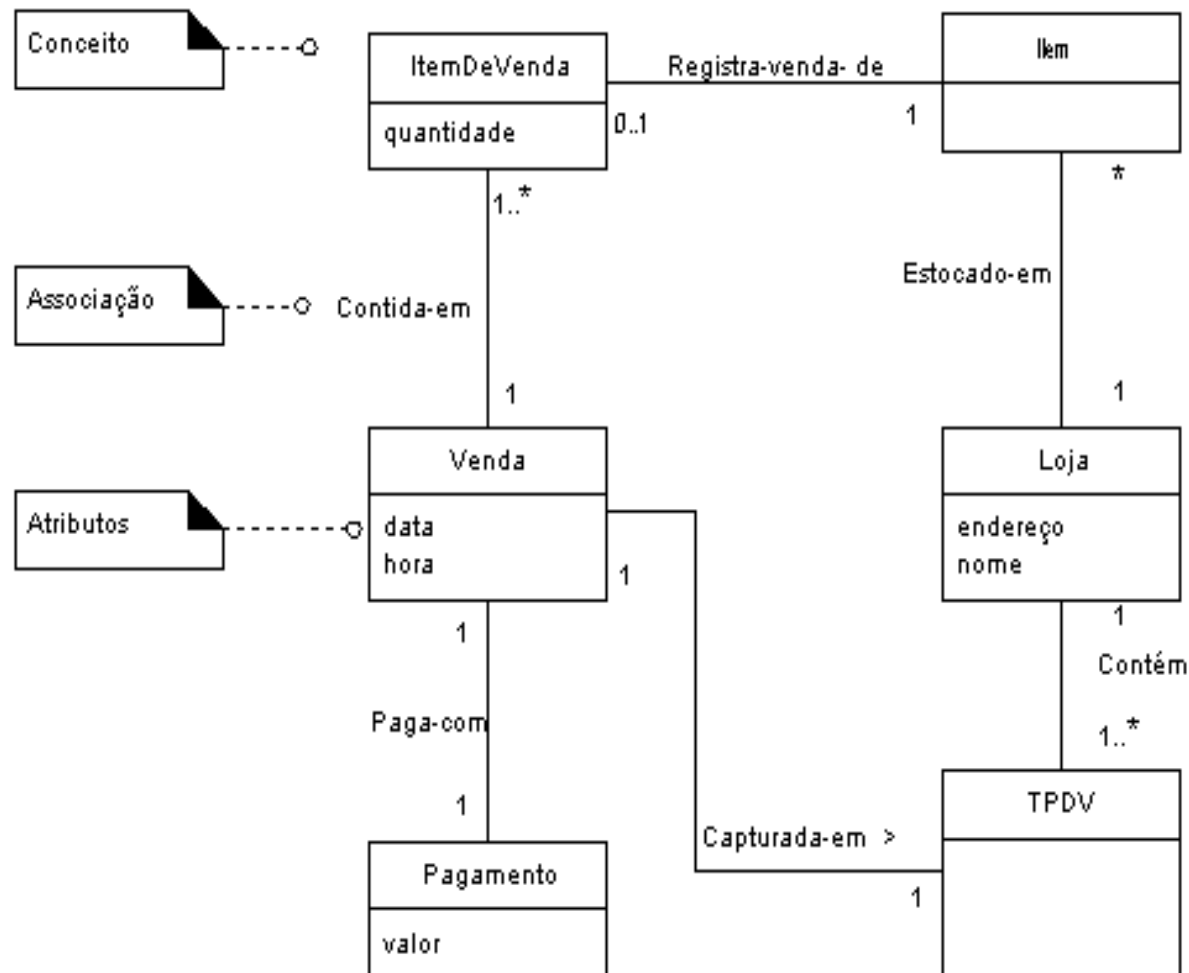
- O modelo conceitual é o artefato mais importante criado durante a análise.
- O modelo conceitual ilustra os conceitos importantes do domínio do problema, suas associações e atributos.
- É importante lembrar que os conceitos levantados aqui são do domínio do problema e não conceitos associados a software.

Modelo Conceitual

- Ao fazer análise orientada a objetos, a decomposição do domínio do problema utiliza objetos e não funções ou processos como na análise estruturada

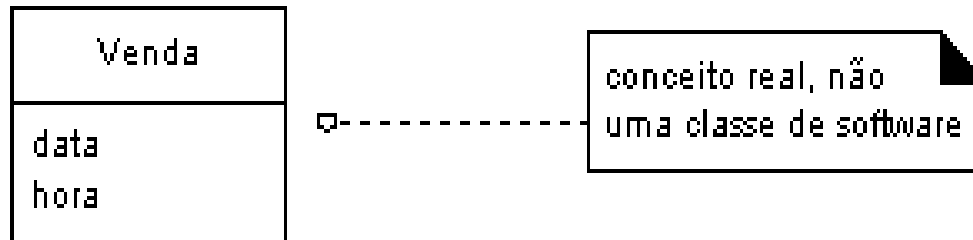
Modelo Conceitual do TPDV

- Possível modelo inicial

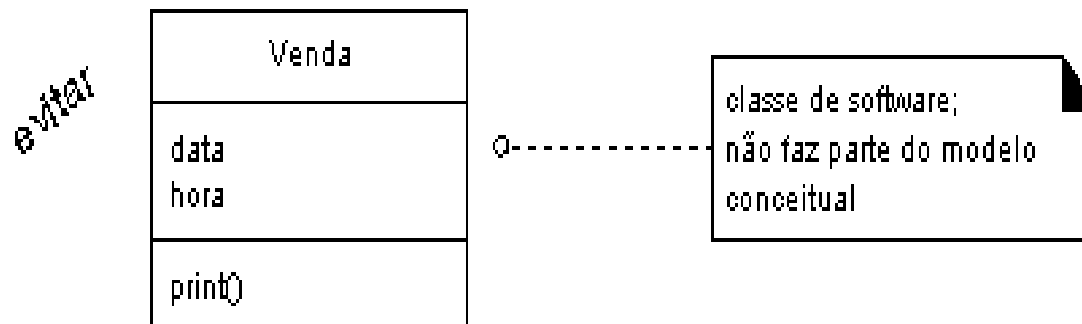
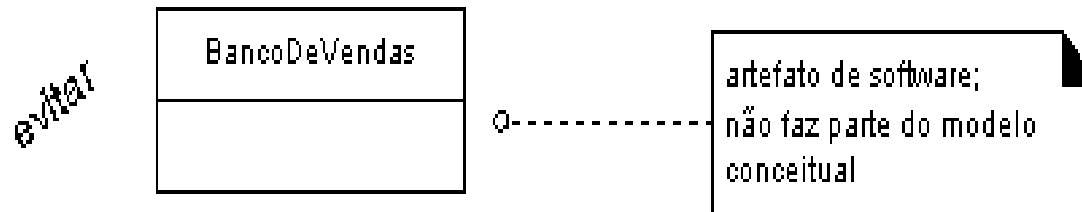


Modelo Conceitual

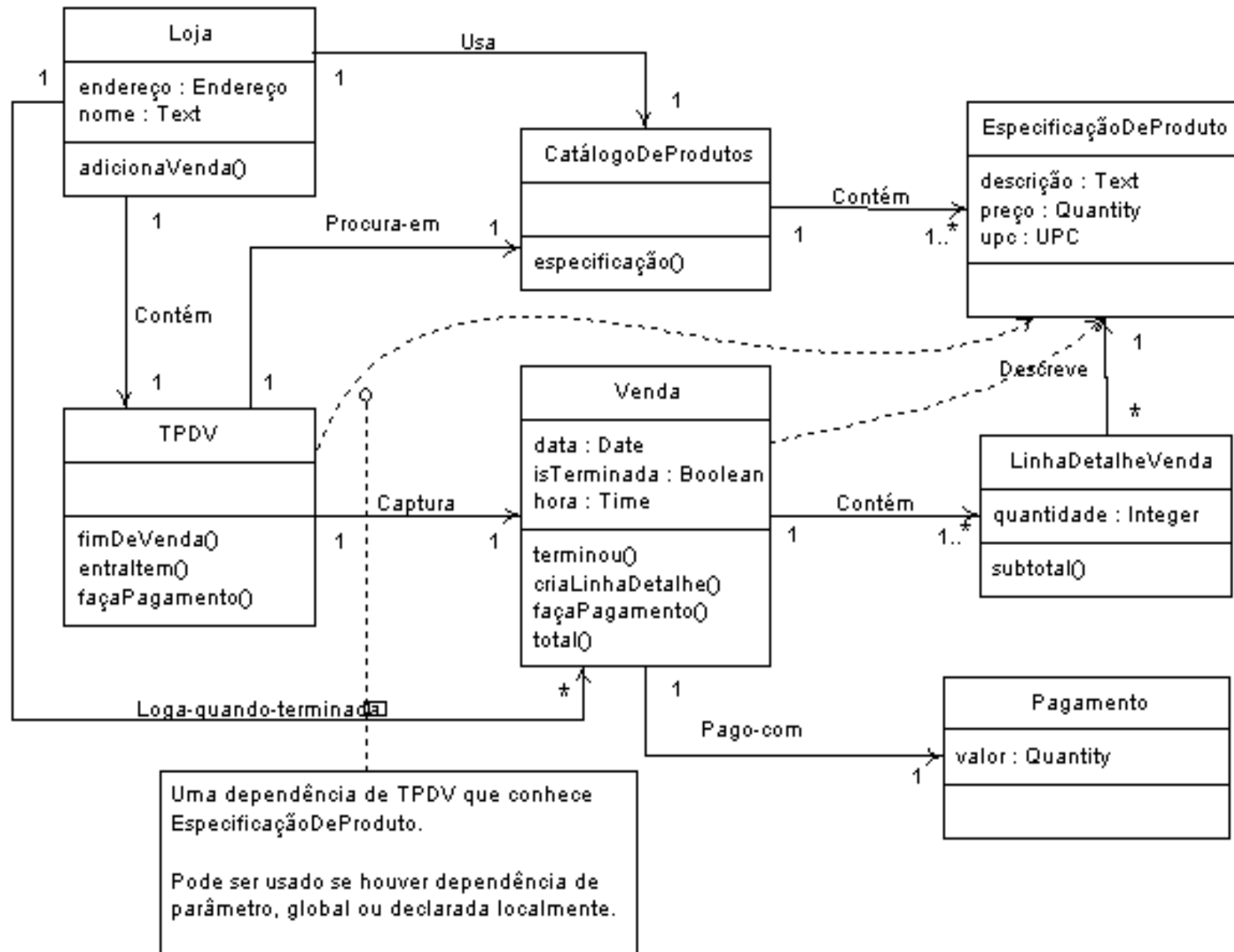
Exemplo correto



Exemplo errado

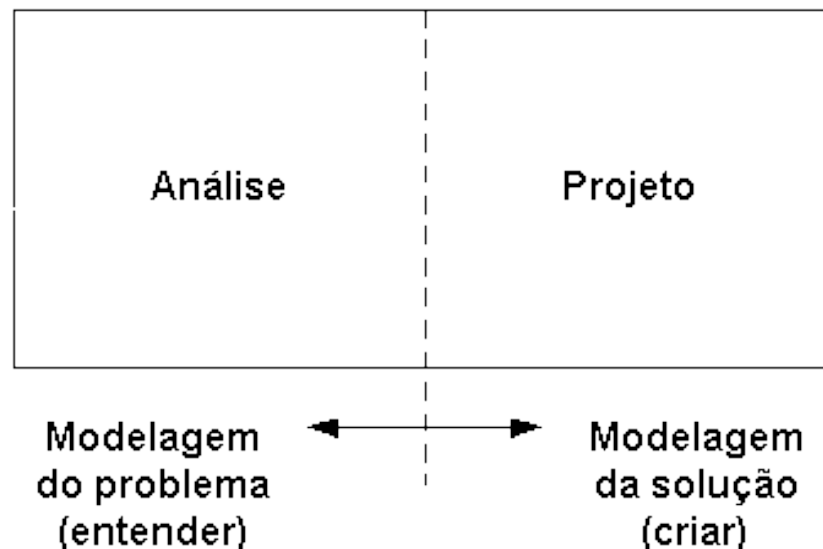


Possível Modelo de Design



Análise vs. Projeto

- Primeira alternativa de definição:
 - A análise modela o problema e consiste das atividades necessárias para entender o domínio do problema (o que deve ser feito). É uma atividade de investigação.
 - O projeto modela a solução e consiste das atividades de criação (como pode ser feito)

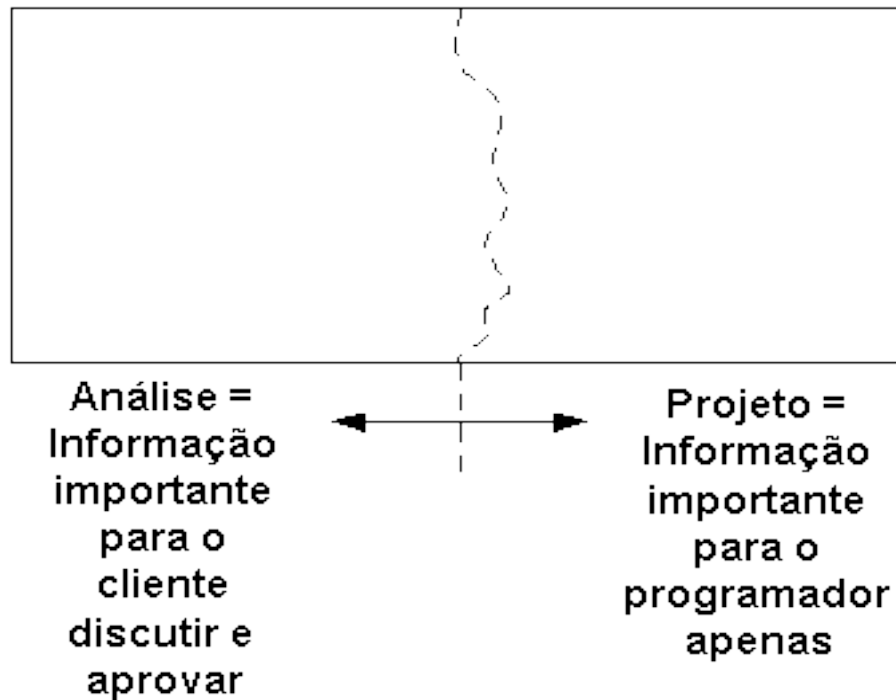


Análise vs. Projeto

- Segunda alternativa:
 - A análise consiste de todas as atividades feitas com ou para o conhecimento do cliente. A informação produzida é aquela que o cliente deve discutir e aprovar.
 - O projeto inclui as atividades que resultam em informação que interessa apenas ao programador.
 - Com essa definição, a análise invade um pouco o "lado da solução", pois o cliente deve discutir alguns tipos de interações que ocorrerão na interface do usuário, etc.

Análise vs. Projeto

- Segunda alternativa:



- Portanto não há definição binária que isole "análise" de "projeto".
- Um modelo de análise deve ser aprovado pelo cliente e pode incluir alguma (pequena) discussão da solução, principalmente no que diz respeito à interface com usuário.