

IDENTIFICAÇÃO DA NECESSIDADE

Análise de Objetivos de Negócio

Analisar o objetivo é extremamente crucial ao sucesso do projeto

O projeto final de sistemas não é analisado em termos de sua beleza ou elegância técnica, mas em termos de benefícios para o negócio. Embora seja tentador para o técnico não se meter em assuntos não técnicos, não pode pular essa fase.

O que se deve descobrir sobre o cliente

1. Conhecendo o negócio do cliente

Antes de discutir objetivos de negócio com o cliente, é bom entender o negócio:

- O cliente participa de que indústria ou área de serviços?
- Qual é o mercado do cliente?
- Quem são os fornecedores e parceiros do cliente?
- Que produtos e serviços o cliente produz?
- Que produtos e serviço o cliente utiliza?
- Quais as vantagens competitivas do cliente?

NOTA: Seu projeto poderá ajudar a melhorar a posição competitiva do cliente?

2. Conhecendo a estrutura organizacional do cliente

Nas primeiras reuniões com o cliente, descubra a estrutura organizacional:

- Quais são os departamentos?
- Quais são as linhas de negócio?
- Quais são os parceiros?
- Onde estão as filiais?

Seu projeto refletirá provavelmente a estrutura corporativa

- Identifique os maiores grupos de usuários, pois isso afetará o fluxo de tráfego na rede.

Descubra que são os responsáveis técnicos e financeiros pelo projeto do novo sistema

- Quem tem poder de aceitar ou rejeitar sua proposta de projeto?

3. Identificando o objetivo maior da rede

Obtenha, em uma única frase, o objetivo maior do sistema, do ponto de vista do negócio:

- Por que o cliente quer um novo sistema?
- Para quê o sistema será usado?
- Como o sistema deve ajudar o cliente no seu negócio?

Algumas possibilidades do objetivo de negócio para o sistema são:

- Aumentar faturamento e lucro
- Melhorar a comunicação corporativa
- Diminuir o time-to-market (ciclo de vida do produto), aumentando a produtividade dos empregados
- Construir parcerias com outras empresas
- Expandir a operação da empresa para mercados globais

- Mudar o modelo de negócio para se basear num sistema de alcance global
- Modernizar tecnologias obsoletas

*NOTA: Isso quase nunca é o objetivo do negócio;
a tecnologia não muda em função da tecnologia, mas do negócio.*

- Reduzir custos de telecomunicações e de processos
- Fornecer mais informações a mais gente para que tomem decisões melhores de negócios, mais rapidamente
- Melhorar a segurança e confiabilidade de aplicações e dados de missão crítica
- Melhorar o suporte ao cliente (do cliente)
- Oferecer novos serviços ao cliente (do cliente)

4. Identificando os critérios de sucesso

Quais são os critérios de sucesso do projeto, do ponto de vista do cliente?

A luz de quê o cliente vai dizer que o novo sistema é bem sucedido?

- Diminuir os custos operacionais
- Aumentou o faturamento
- Construiu parcerias

Pode haver resposta diferente para pessoas diferentes

- Diretoria
- Gerentes operacionais
- Usuários finais
- Engenheiros de suporte

Lembre de formar alianças e comprometimento interno para melhor as chances de sucesso do projeto

5. Identificando os critérios de sucesso

O que ocorre se o projeto for um fracasso (não for feito, tiver desempenho inadequado)?

Qual é a visibilidade do projeto à alta direção da empresa?

Quais são os efeitos de uma má operação nos aspectos operacionais do negócio?

6. Identificando o novo escopo do projeto

Trata-se do projeto de um novo sistema ou de uma ampliação/ajuste de um sistema existente?

Qual tipo de sistema está sendo projetado?

- Sistema de Processamento de Transações – atendem ao nível operacional
- Sistema de Trabalhadores de Conhecimento – criam informações e manipulam esses dados
- Sistema de Informações Gerenciais – suporte aos níveis mais altos através de relatórios e processos correntes
- Sistema de Apoio a Decisão – analisam e armazenam dados para auxiliar em decisões críticas
- Sistemas de Apoio ao Executivo – incorporam dados externos como leis e dados de concorrentes

7. Identificando as aplicações do que cliente

Descobrir aplicações atuais e futuras

Uma tabela como mostrada abaixo pode ser preenchida:

Nome da Aplicação	Tipo de Aplicação	Aplicação Nova?	Criticalidade	Comentários

Nome da Aplicação: dado pelo usuário

Tipo de Aplicação: Aplicações do Usuário/Aplicações de Sistema

Criticalidade: usar um número, até poder levantar o *downtime* (tempo de indisponibilidade) aceitável

- 1 – Extremamente crítico
- 2 – Mais ou menos crítico
- 3 – Crítico

Comentários: qualquer informação relevante

- Quando a aplicação deixará de ser usada
- Quando a aplicação será implantada
- Planos de uso regional de certas aplicações

Análise de Restrições de Negócio

1. Política e politicagem

Não entender certos aspectos políticos da situação do cliente pode comprometer o projeto.

Escute o que acontece nas reuniões para identificar os aspectos políticos.

- Relações entre grupos
- Gerentes mais comprometidos
- Tolerância à falhas

Se informe sobre políticas internas da empresa

2. Aspectos técnicos de recursos humanos

Se informe sobre as habilidades dos técnicos da empresa

- Certas empresas não estão prontas para certos tipos de sistemas complexos

3. Restrições orçamentárias

Se informe sobre o orçamento disponível, incluindo:

- Aquisição de equipamentos
- Contratos de manutenção
- Contratos de suporte
- Consultoria
- Ajudar gerente a elaborar uma análise ROI (Return On Investment)

4. Cronograma

Não se controla cronograma, mas deve se adequar a ele

Opinar enfaticamente se achar o cronograma inviável

4. Checklist de Objetivos de Negócio

Estará pronto para execução se:

- Pesquisei a área de negócio e os concorrentes
- Entendo a estrutura corporativa do cliente
- Lista dos objetivos do cliente
- Entendeu o escopo do sistema
- Conheço o orçamento e concordo com o cronograma
- Discuti as necessidades de treinamento

ESTUDO DE VIABILIDADE

Um estudo de viabilidade vem buscar elementos que visem validar financeiro/administrativamente a execução de um projeto. É necessário o levantamento das seguintes considerações, sempre levando em considerações suas vantagens e desvantagens sobre cada especificação:

1. Viabilidade Operacional

- Performance
- Eficiência
- Informação
- Serviços
- Personalização
- Interface com o usuário / Facilidade de uso / Praticidade

2. Viabilidade Técnica

- Linguagens de Programação
- Sistema de Gerenciamento de Banco de Dados
- Interfaces gráficas

ANÁLISE CUSTO/BENEFÍCIO

Uma boa análise de custo/benefício para uma determinada ação deve incluir uma dimensão temporal e outras características de um bom cenário de negócios. Deve-se considerar as expectativas de retorno e dispêndio ao longo de um tempo pré-estabelecido bem como os modelos de custo e benefícios que determinam o que deve ser incluído neste cenário ou não.

Uma análise de custo/benefício busca quantificar cada benefício e cada custo a ser incluído em uma análise financeira, mesmo aqueles custos e/ou benefícios denominados intangíveis ou ocultos. Ela também preocupa-se em não omitir nenhum custo ou benefício não quantificável. É importante tentar quantificar o máximo possível: se nenhum valor financeiro é associado a determinado custo ou benefício, estes não contribuirão em nada à análise financeira. Uma organização pode investir em tecnologia na tentativa de “melhorar sua imagem”, ampliar a satisfação de seus clientes ou criar um “ambiente de trabalho mais profissional”. Mas qual valor monetário seria creditado a estes benefícios?

CASOS DE USO

Um caso de uso representa uma unidade discreta da interação entre um usuário (humano ou máquina) e o sistema. Um caso de uso é uma unidade de um trabalho significativo. Por exemplo: o "login para o sistema", "registrar no sistema" e "criar pedidos" são todos casos de uso. Cada caso de uso tem uma descrição o qual descreve a funcionalidade que irá ser construída no sistema proposto. Um caso de uso pode "incluir" outra funcionalidade de caso de uso ou "estender" outro caso de uso com seu próprio comportamento.

Casos de uso são tipicamente relacionados a "atores". Um ator é um humano ou entidade máquina que interage com o sistema para executar um significativo trabalho.

É importante notar que não descreve como o software deverá ser construído, mas sim como ele deverá se comportar quando estiver pronto. Um software frequentemente é um produto complexo, e sua descrição envolve a

identificação e documentação de vários casos de uso, cada um deles descrevendo uma "fatia" do que o software ou uma de suas partes deverá oferecer.

Normalmente evitam o uso de termos técnicos, preferindo a linguagem do utilizador final, são empregados tanto por quem desenvolve o software quanto pelos utilizadores do software.

Definições

Segundo Ivan Jacobson, podemos dizer que um caso de uso é um *"documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo"*.

Um caso de uso é uma técnica de modelagem usada para descrever o que um novo sistema deve fazer. Ele é construído através de um processo interativo no qual as discussões entre o cliente e os desenvolvedores do sistema conduzem a uma especificação do sistema da qual todos estão de acordo.

Os casos de uso tem por objetivo:

- Decidir e descrever os requisitos funcionais do sistema.
- Fornecer uma descrição clara e consistente do que o sistema deve fazer.
- Permitir descobrir os requisitos funcionais das classes e operações do sistema. (*Casos de uso NÃO são requisitos*)

Podemos dizer que os componentes de um modelo de casos de uso são :

Ator - é um papel que tipicamente estimula/solicita ações/eventos do sistema e recebe reações. Cada ator pode participar de vários casos de uso

Casos de uso - documento narrativo que descreve a sequência de eventos feitos por um ator no uso do sistema.

Sistema - O sistema a ser modelado

Na UML o modelo de casos de uso consiste de diagramas de casos de uso que mostram os atores, os casos de uso e seus relacionamentos. Nos primeiros contatos com os modelos de casos de uso surgem com frequência três perguntas para as quais não existe uma resposta absoluta, são elas:

01. Como identificar atores?

Para identificar os atores que vão participar do modelo devemos fazer as seguintes perguntas

- Quem usa o sistema?
- Quem inicializa o sistema?
- Quem fornece os dados?
- Quem usa as informações?

02. Como descrever atores?

- Nome do caso de uso
- Tipo de uso (frequente, ocasional, etc...)
- Descrição de seu papel no sistema

03. Como Identificar casos de uso?

Os casos de uso são interações entre os atores e o sistema. Temos então ações do ator e ações do sistema. Sendo que os atores sempre iniciam a ação.

Vamos dar um exemplo prático para que tudo fique mais claro. Vamos supor, por questão de simplicidade, que temos que modelar usando casos de uso a compra de item em uma loja com um terminal de ponto de venda.

Quais são os atores?

Quem usa o sistema é o cliente e ele usa um terminal de caixa.

Como podemos identificar o caso de uso?

Podemos chamar este caso de uso de: Comprar Item.

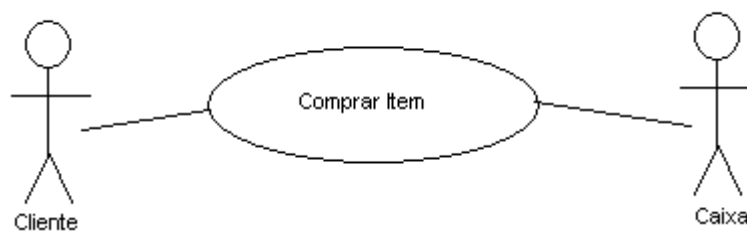
Agora vamos a um descrição textual do caso de uso **Comprar Item** onde atual os atores **cliente e caixa**.

Use case - Comprar Item

Atores - Cliente, Caixa

Descrição - Este caso de uso começa quando um cliente chega ao terminal com itens que deseja comprar. O caixa registra os itens , recebe o pagamento e emite uma nota fiscal. O Cliente recebe os itens comprados.

Na **UML** temos o diagrama de caso de uso que pode ser representado para o caso acima da seguinte forma:



Algumas considerações:

- Nomeie um caso de uso começando com um verbo, para enfatizar que ele é um processo. *Ex: Cadastrar Cliente, Comprar Item, etc.*
- Para identificar claramente um ator iniciador e um evento, comece a descrição da sequência de um caso de uso usando o seguinte esquema:
 - Este caso de uso começa quando o <Ator> <Evento que inicia o caso de uso>
 - Ex: Este caso de uso começa quando um cliente chega com vários itens para comprar

Vamos a um outro exemplo:

Suponha que você tenha um almoxarifado de peças onde clientes façam pedido e onde um operador receba tarefas do sistema para buscar peças para os pedidos dos clientes e distribuir peças do setor de compras para o almoxarifado. (O exemplo é bem simples para facilitar o entendimento do conceito)

Como poderíamos identificar os atores e os casos de uso?

Vamos identificar os atores. Eles são: Cliente, Operador, Sistema e Setor de Compras.

Certo? Não, errado! Sistema não pode ser um ator pois ele não se ajusta ao conceito dado a um ator : Um agente externo ao sistema.

Lembre-se um ator é um papel que interage com o sistema então no lugar de Sistema poderíamos sugerir um administrador ou gerente. Então os atores seriam: Cliente, Operador, Administrador e Compras.

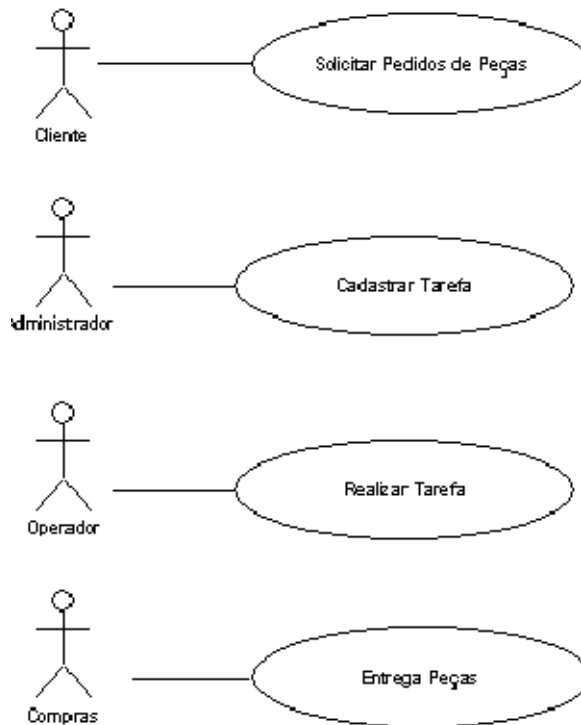
E os casos de usos, quais seriam?

- solicita peças (ator Cliente)
- entrega peças (ator Compras)
- buscar pedidos (ator operador)
- distribuir pedidos (ator operador)
- cadastrar Tarefas (administrador)

Certo? Errado! No caso do ator operador ele não atua sobre o sistema nos casos de uso buscar pedidos e distribuir pedidos, ele atua sobre o sistema realizando Tarefa. Então o correto seria.

- solicita peças (ator Cliente)
- entrega peças (ator Compras)
- realiza Tarefa (ator operador)
- cadastrar Tarefas (administrador)

Usando a representação **UML** para os diagramas de casos de uso teríamos :



Este seria o caso de uso preliminar pois não temos muito detalhamento nesta etapa do modelo. A próxima etapa seria realizar um refinamento do modelo a fim de obter o relacionamento entre os casos de uso através da **generalização, inclusão ou extensão**.

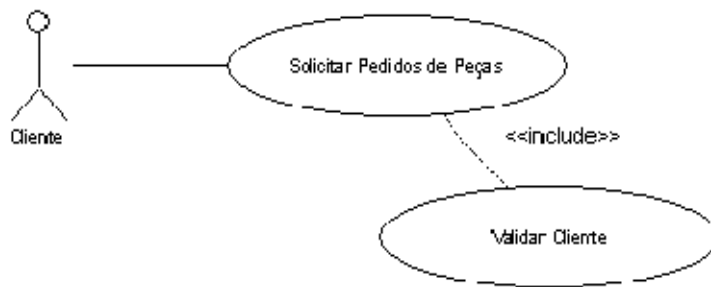
A partir do diagrama de casos de uso preliminar muitas vezes temos que definir casos de usos adicionais separadamente pois as operações se encontram duplicadas em outros casos de uso ou são complexas e longas e a separação nos ajuda a compreendê-las.

Os relacionamentos possíveis são :

01. Inclusão: Se um caso de uso inicia ou inclui o comportamento de outro, dizemos que ele usa o outro.

Ex: No caso de uso **Comprar Item** se o pagamento for feito com dinheiro podemos ter a inclusão **PagarComDinheiro**

O relacionamento de inclusão em UML é ilustrado com uma linha de generalização com o rótulo **<<include>>**. Então para o exemplo do cliente com o use case Solicitar Pedidos de peças teríamos:

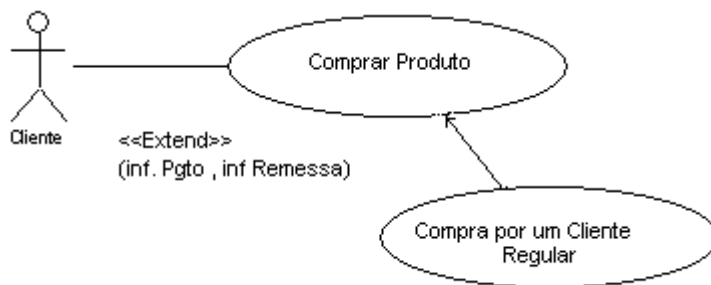


As propriedades básicas da inclusão são :

- realizar um decomposição funcional;
- reduzir a complexidade de um caso de uso;
- O caso de uso básico não pode executar sem a inclusão;
- Comportamento comum.

02. Extensão - Define pontos de extensão que adicionam comportamento a um caso de uso base descrevendo uma variação do comportamento normal. O caso de uso base pode ser executado mesmo sem a extensão.

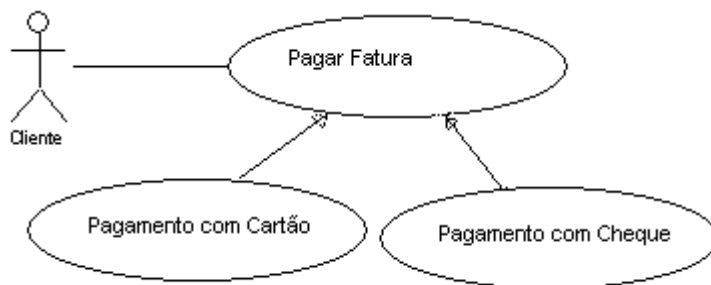
Ex: O caso de uso Comprar Produto pode apresentar a extensão Compra por um Cliente Regular. Abaixo temos o diagrama UML.



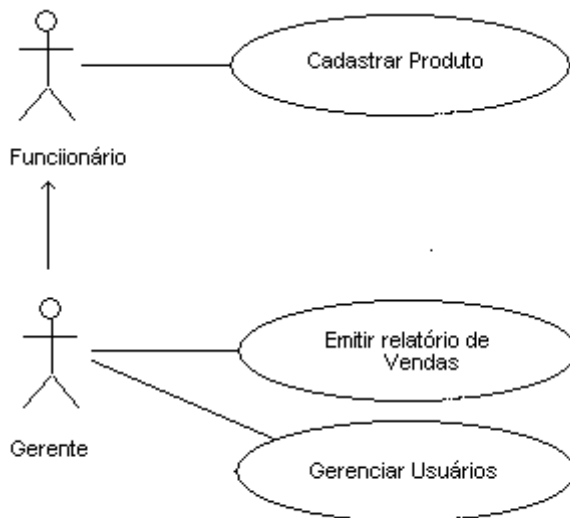
Os pontos de extensão são indicados na linha entre os casos de uso do sistema.

03. Generalização - Indica um caso de base que possui diferentes especializações e inclui comportamento ou sobrescreve o caso de uso base.

O caso de uso Pagar fatura apresenta as generalizações: Pagamento com cartão e Pagamento com Cheque, conforme o diagrama abaixo:



Além disto temos também os relacionamentos entre atores onde um ator especializado pode acessar os casos de uso de um Ator base. Abaixo temos um exemplo onde o Ator gerente acessa os casos de uso do ator funcionário.



Exemplo de um Use Case de alto nível: Comprar item

Segue uma breve descrição do processo de comprar um item numa loja quando um TPDV é utilizado

Use case:	Comprar item
Atores:	Cliente, Caixa
Tipo:	primário (a ser explicado logo)
Descrição:	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.

- UML não força o formato exato de um Use Case. A clareza na descrição é o essencial.
- Iniciamos acima com um Use Case de alto nível, fornecendo poucos detalhes
 - Útil para entender rapidamente os processos principais envolvidos

Exemplo de um Use Case expandido: Comprar item com dinheiro vivo

- Ignoramos a questão de dar baixa no inventário aqui

Use Case:	Comprar item com dinheiro
Atores:	Cliente (iniciador), Caixa
Propósito:	Capturar uma venda e seu pagamento em dinheiro
Resumo:	m 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.
Tipo:	primário e essencial
Referência cruzada:	R1.1, R1.2, R1.3, R1.7, R1.9, R2.1 (pode fazer referência a outros use Cases)

Sequência típica de eventos

Ação do ator	Resposta do sistema
1. O Use Case inicia quando um cliente chega a um caixa munido de TPDV com itens a comprar	
2. O caixa registra a identificação de cada item	3. Determina o preço do item e adiciona a informação ao total da transação de venda

Se houver mais itens, o caixa pode informar a quantidade também	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.

Use Cases Primários, Secundários e Opcionais

- Use Cases primários: representam processos importantes e/ou comuns (comprar itens)
- Use Cases secundários: representam processos menos importantes ou raros (Pedido de reposição de estoque)
- Use Cases opcionais: representam processos que talvez não sejam considerados

Use Cases Essenciais versus Reais

- Use Cases essenciais: possui uma descrição breve, muito abstrata, sem detalhes, sem mencionar tecnologias empregadas (poderia incluir uma frase: "cliente se identifica" sem maiores detalhes porque a forma de identificação pode mudar com tempo e com a disponibilidade de tecnologia)
- Use Cases reais: muito mais concretos, mencionando tecnologias correntemente usadas ("cliente se identifica passando seu Smart Card no leitor")
 - Use Cases essenciais são usados mais cedo no processo (antes do design) e Use Cases podem ser transformados em Use Cases reais durante o design
- Durante o processo de desenvolvimento:
 - Durante o levantamento de requisitos
 - Crie Use Cases em formato de alto nível
 - Crie um diagrama de Use Cases, indicando os relacionamentos
 - Selecione os Use Cases mais críticos, que deverão influenciar o sistema mais ou que envolvam mais risco e escreva uma versão Expandida Essencial
 - Priorize os Use Cases (próximo capítulo)
 - Durante a fase de análise

- Escreva uma versão Expandida Essencial dos Use Cases sendo atacados na iteração corrente (se já não estiver pronta)
- Durante a fase de projeto
 - Escreva uma versão Expandida Real dos Use Cases sendo atacados na iteração corrente

Estudo de Caso: O sistema TPDV

Identificar o sistema, atores e Use Cases

- O sistema será a combinação de hardware (o terminal) e software executando nele
- Os atores relevantes e alguns processos que eles iniciam são:

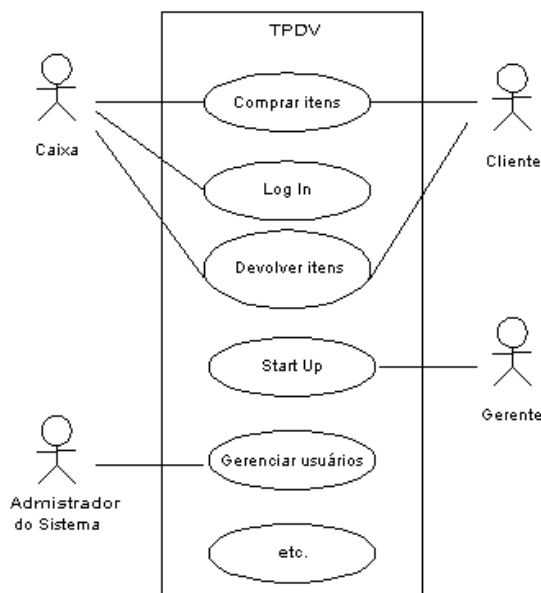
Caixa	Fecha caixa
Cliente	Compra itens; Devolve itens
Gerente	Start; Shut Down
Administrador do sistema	Adiciona novos usuários

Escrever Use Cases em formato de alto nível

Use case: Comprar item
 Atores: Cliente (iniciador), Caixa
 Tipo: primário
 Descrição: 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.

Use case: Start Up
 Atores: Gerente
 Tipo: primário
 Descrição: Um gerente liga um TPDV para o preparar para uso pelos Caixas.
 O gerente verifica que a data e horas estão corretas.
 O sistema está pronto para uso pelos Caixas

Elaborar um diagrama de Use Cases



Priorização de Use Cases

Objetivos

- Priorizar Use Cases
- Quando necessário, criar versões simplificadas de Use Cases
- Alocar os Use Cases às iterações de desenvolvimento

Introdução

De forma geral, cada iteração da fase de construção vai tratar de um ou mais Use Cases identificados. Às vezes, um Use Case pode ser tratado em duas ou mais iterações, com versões simplificadas tratadas primeiro

Priorização de Use Cases

- Normalmente, quem prioriza os Use Cases é o cliente (ou Domain Expert)
 - Os técnicos podem ajudá-lo fornecendo estimativas de esforço de desenvolvimento para os vários Use Cases
 - Mas lembre que é o cliente que sabe o *Business Value* de cada Use Case
- Alguns parâmetros que podem aumentar a prioridade de um Use Case
 - Afetam muito a arquitetura
 - O primeiro incremento deveria ser um teste arquitetural (uma validação da viabilidade da arquitetura)
 - O processo de desenvolvimento deve ser "architecture-centric"
 - Pouco esforço dando muito insight sobre o projeto do sistema
 - Use Cases que envolvem muito risco, são críticos no tempo
 - O processo de desenvolvimento deve ser "risk-confronting"
 - Envolvem business processes fundamentais
 - Deixe o cliente falar!
 - Podem aumentar significativamente o lucro ou diminuir as despesas/custos
 - Deixe o cliente falar!

O estudo de caso

- Uma possibilidade de priorização

Prioridade	Use Case	Justificativa
Alta	Comprar itens	Satisfaz quase todos os critérios de aumento de prioridade
Média	Adicionar novos usuários login Devolver itens	Afeta a segurança Afeta a segurança Processo importante; afeta a contabilidade
Baixa	Fechamento de caixa Start Up Shut Down	Efeito mínimo na arquitetura Definição depende de outros Use Cases Efeito mínimo na arquitetura

- O Use Case Start Up poderia ser feito numa versão mínima se ele for necessário ao funcionamento do resto
- Escalonamento final dos Use Cases no estudo de caso

- Teremos que atacar Comprar Itens na primeira iteração além de uma versão reduzida de Start Up
- Como Comprar Itens é complexo, pode-se quebrá-lo em várias versões incrementais:
 - Comprar Itens versão 1:
 - Pagamento em dinheiro apenas
 - Sem atualização de estoque
 - Loja stand-alone, sem integração a uma organização maior
 - Entrada manual de UPC, sem leitor de código de barras
 - Sem cálculo de impostos
 - Sem descontos especiais
 - Sem log in
 - Sem manutenção de perfil de usuário (preferências, ...)
 - Sem controle contábil do valor em caixa
 - Recibo não contém informação completa (sem ID do Caixa, ID do TPDV)
 - Comprar Itens versão 2: (todas as formas de pagamentos)
 - Comprar Itens versão 3: (versão completa)
- As três versões (além dos outros Use Cases) são distribuídas ao longo dos incrementos

CURSO ONLINE

<http://www.aspercom.com.br/ead/>

Entendendo Casos de Uso – Vale 5,00 na nota da 1ª AP.