

# **ANALISE E PROJETOS DE SISTEMAS**

**Aula: Modelagem de  
sistemas de software.**



Prof. Anderson Augusto Bosing

# Objetivo

**Compreender como os modelos gráficos podem ser usados para representar sistemas de software;**

**Compreender por que diferentes tipos de modelo são necessários e as perspectivas fundamentais de modelagem de sistema de contexto, interação, estrutura e comportamento;**

**Terá sido apresentado a alguns dos tipos de diagramas da Unified Modeling Language (UML), e como eles podem ser usados na modelagem de sistema.**



# Introdução a Modelagem de Sistemas

**Modelagem de sistema** é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema.

A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica, que, atualmente, quase sempre é baseada em notações de UML (linguagem de modelagem unificada, do inglês Unified Modeling Language).

No entanto, também é possível desenvolver modelos (matemáticos) formais de um sistema, normalmente como uma especificação detalhada do sistema.



# O que é modelagem de Sistemas ?

A modelagem de sistemas de software consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema, considerando-se várias perspectivas diferentes e complementares.



# Introdução a Modelagem de Sistemas

Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema; durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implementam; e, após isso, são usados para documentar a estrutura e a operação do sistema. Você pode desenvolver modelos do sistema existente e do sistema a ser desenvolvido.

- **Modelos do sistema existente** são usados durante a engenharia de requisitos. Eles ajudam a esclarecer o que o sistema existente faz e podem ser usados como ponto de partida para discutir seus pontos fortes e fracos. Levam, então, os requisitos para o novo sistema.



# Introdução a Modelagem de Sistemas

- Modelos do novo sistema são usados durante a engenharia de requisitos para ajudar a explicar os requisitos propostos para outros stakeholders do sistema. Os engenheiros usam esses modelos para discutir propostas de projeto e documentar o sistema para a implementação. Em um processo de engenharia dirigida a modelos, é possível gerar uma implementação completa ou parcial do sistema a partir do modelo de sistema.



# **Previsão do Comportamento Futuro do Sistema**

O comportamento do sistema pode ser discutido mediante uma análise dos seus modelos.

Os modelos servem como um “laboratório”, em que diferentes soluções para um problema relacionado à construção do sistema podem ser experimentadas.



# Aspecto Importante

O aspecto mais importante de um modelo de sistema é que ele deixa de fora os detalhes. Um modelo é uma abstração do sistema a ser estudado, e não uma representação alternativa dele. Idealmente, uma representação de um sistema deve manter todas as informações sobre a entidade representada. Uma abstração deliberadamente simplifica e seleciona as características mais salientes.



# Diferentes Perspectivas

A partir de perspectivas diferentes, você pode desenvolver diversos modelos para representar o sistema. Por exemplo:

1. Uma perspectiva externa, em que você modela o contexto ou o ambiente do sistema.
2. Uma perspectiva de interação, em que você modela as interações entre um sistema e seu ambiente, ou entre os componentes de um sistema.
3. Uma perspectiva estrutural, em que você modela a organização de um sistema ou a estrutura dos dados processados pelo sistema.
4. Uma perspectiva comportamental, em que você modela o comportamento dinâmico do sistema e como ele reage aos eventos.



# **E o que essas perspectivas tem em comum ?**

## **Atividade - Visões de Arquitetura.**

**Quais visões ou perspectivas são úteis durante o projeto e documentação de Arquitetura de um Sistema ?**

**Quais notações devem ser utilizadas para descrever os modelos de arquitetura ?**

**É possível representar em um único diagrama todas as informações relevantes a respeito da arquitetura de um sistema ?**

**Detalhe a visão 4 + 1, de Kruchten, da arquitetura do sistema (KRUCHTEN, 1995).**

**Leitura Obrigatória :**

**Capítulo 6.2 Visões de Arquitetura**

**SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo: Pearson, 2018. E-book.**

**Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 04 ago. 2023.**



# **Atividade - Visões de Arquitetura.**

**Quais visões ou perspectivas são úteis durante o projeto e documentação de Arquitetura de um Sistema ?**

**Quais notações devem ser utilizadas para descrever os modelos de arquitetura ?**

**É possível representar em um único diagrama todas as informações relevantes a respeito da arquitetura de um sistema ?**

**Detalhe a visão 4 + 1, de Kruchten, da arquitetura do sistema (KRUCHTEN, 1995).**

**Leitura Obrigatória :**

**Capítulo 6.2 Visões de Arquitetura**

**SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo: Pearson, 2018. E-book.**

**Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 04 ago. 2023.**



# **ANALISE E PROJETOS DE SISTEMAS**

**Aula: Atividades Típicas  
de um Processo de  
Desenvolvimento.**



Prof. Anderson Augusto Bosing

# Levantamento dos Requisitos

Compreensão do problema, visando permitir que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido.

# Análise de requisitos

Estudo detalhado dos requisitos levantados e a construção de modelos para representar o sistema a ser construído.



# Levantamento dos Requisitos e Análise de requisitos

- O escopo do software é refinado;
- Nessa fase, a interação entre quem desenvolverá e o cliente é muito grande, contudo não se discute como será feito o software, mas sim o que ele deve fazer.
- Devem ser analisados o domínio do problema e o domínio da solução.
- São usadas as histórias de usuário (User Stories) ou especificação tradicional de requisitos, que são pequenas histórias escritas para auxiliar na definição do requisito entre quem desenvolve e o cliente.
- Todo esse material é compilado e gera um relatório que servirá para os tomadores de decisão definirem, ou não, a continuidade e o desenvolvimento do software.



# Projeto

Determina como o sistema funcionará para atender os requisitos, de acordo com os recursos tecnológicos existentes. A modelagem do software pode ser realizada por um conjunto de diagramas, por exemplo, pela UML.

- Utiliza a fase anterior como insumo.
- Essa fase é voltada ao programador do software.
- Definição de como seria a interface que o usuário opera, quais cores seriam utilizadas na interface, como seria o fluxo de funcionamento de cada botão existente na interface, em qual banco de dados ficariam os dados gerados.
- O projeto, diferente da fase anterior, define como as coisas acontecerão.



# Implementação

Ocorre a tradução da descrição computacional da fase de projeto em código executável através do uso de linguagens de programação. É nessa fase que os diagramas/modelos criados “ganham vida”.

- Essa fase deve traduzir o projeto em um software, utilizando ferramentas e linguagens adequadas.
- Dadas as inúmeras metodologias de desenvolvimento, cada programador pode desenvolver o código do sistema de uma forma diferente.



# Testes

Para verificar a corretude do sistema, levando-se em conta a especificação feita na fase de projeto.

- **Teste de unidade:** cada componente é testado individualmente, sem conexão com outros componentes.  
✓ Exemplo: testar apenas o método que faz a soma dos valores das vendas, dados dois números, ele deve retornar a soma dos dois.
- **Teste de módulo:** um módulo é um agrupamento de pequenos componentes, que tem uma função específica.  
✓ Exemplo: testar a geração do relatório de vendas de um produto.
- **Teste de subsistemas:** são testados módulos integrados que controlam as interfaces do sistema.  
✓ Exemplo: testar as interfaces do sistema de compra e venda.



# Testes

- **Teste de sistema:** testa a integração dos subsistemas que formam o sistema principal.
  - ✓ Exemplo: realizar login, operações e geração de relatórios em um sistema.
  
- **Teste de aceitação:** testes realizados com dados reais fornecidos pelos clientes. Último teste antes de colocar o sistema em operação.
  - ✓ Exemplo: quando há muitos usuários na base e o login não acontece de forma instantânea.



# Implantação

O sistema é empacotado, distribuído e instalado no ambiente do usuário. São entregues os manuais do sistema e os usuários são treinados para utilizar o sistema.

➤ O software deve ser instalado em ambiente produção.

➤ Envolve:

- ✓ Treinamento de usuários;
- ✓ Configuração do ambiente de produção;
- ✓ Conversão bases de dados (se necessário).



# **ANALISE E PROJETOS DE SISTEMAS**

**Aula: O componente  
humano (participantes do  
processo)**



Prof. Anderson Augusto Bosing

# O componente humano (participantes do processo)

## Gerente de projeto

- Responsável pela gerência e coordenação das atividades necessárias para a construção do sistema, alem de estimar tempo e custo.

## Analista

- Possui conhecimento sobre o domínio do negócio para que possa levantar os requisitos.

## Projetista

- Avalia as alternativas de solução e gera uma especificação detalhada da solução computacional (Ex: projetista de rede, de banco de dados, etc.).

## Programador

- Responsável pela implementação do sistema.

## Cliente

- O cliente usuário e especialista no domínio do negócio e interage diretamente com o Analista para levantar os requisitos do sistema.



# **ANALISE E PROJETOS DE SISTEMAS**

**Aula: Utilização da UML no  
processo iterativo e  
incremental**



Prof. Anderson Augusto Bosing

# Utilização da UML no processo iterativo e incremental

A UML é independente de processo de desenvolvimento.

- Vários processos podem utilizar a UML para modelagem de sistemas OO.

Os artefatos de software construídos através da UML evoluem à medida que as iterações são realizadas.

- A cada iteração, novos detalhes são adicionados a esses artefatos.
- Além disso, a construção de um artefato fornece informações para adicionar detalhes a outros.



# Modelo Incremental

- Segundo Pressman (2011) “modelos evolucionários são iterativos. Apresentam características que possibilitam desenvolver versões cada vez mais completas do software”.
- Dessa forma, o primeiro modelo evolucionário que surgiu é o modelo incremental de desenvolvimento de software.
- Para Sommerville (2014) o sistema incremental tem o objetivo de reduzir o retrabalho custoso do modelo cascata, possibilitando ao cliente postergar decisões e requisitos conforme necessidade, mas mantendo o poder de gerenciamento que o modelo oferece.



Prof. Anderson Augusto Bosing

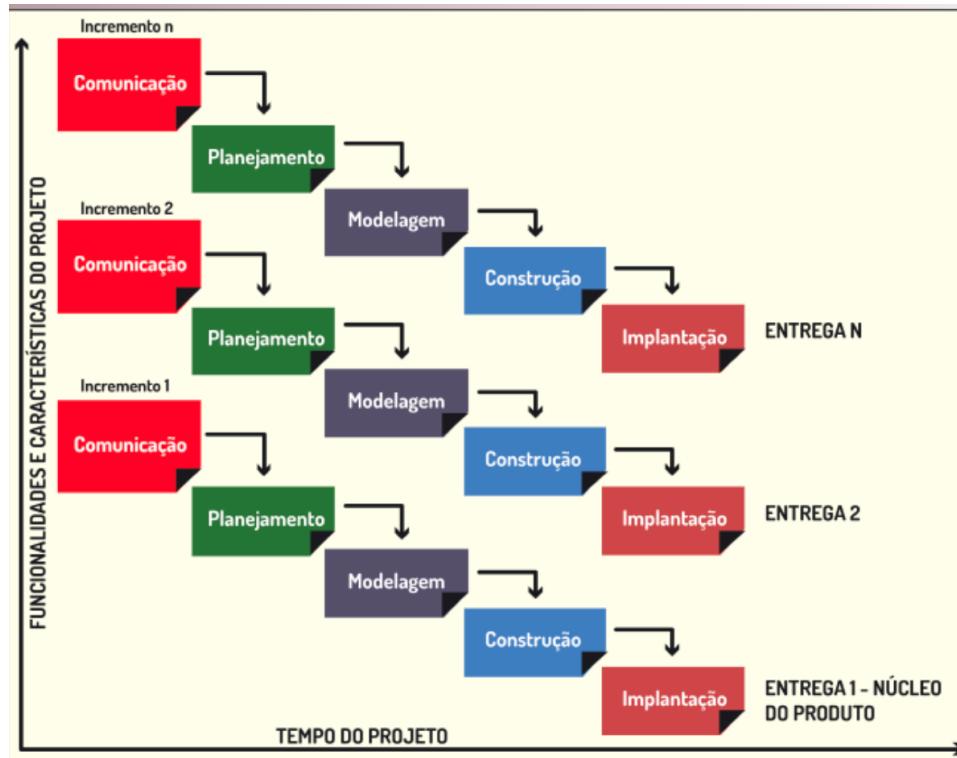
# Modelo Incremental

- Essa metodologia divide a fase de desenvolvimento do software em ciclos completos, passando por todas as fases existentes no modelo clássico, levantamento e análise de requisitos, projeto, implementação e testes.
  
- O modelo trabalha com 5 (cinco) fases distintas: comunicação, planejamento, modelagem, construção e implantação.



Prof. Anderson Augusto Bosing

# Modelo Incremental



1. **Comunicação:** trata das informações do negócio, como são geradas, processadas e quem utilizará. Define os requisitos.
2. **Planejamento:** define os objetos, suas características e como se relacionarão entre si.
3. **Modelagem:** desenha o processo, visando atender à função do negócio, e define os procedimentos necessários para a manipulação dos objetos de dados definidos na etapa anterior.
4. **Construção:** ocorre a geração da aplicação, normalmente por meio de reutilização de componentes que serão integrados posteriormente.
5. **Implantação:** acontece alguns pequenos testes e a entrega de parte do produto.

# EXERCICIO FERRAMENTAS CASE

- Definição de Conceito.
- Qual sua aplicação?
- Como elas podem ser classificadas ?
- Elas podem ser Classificadas de acordo com os serviços que oferecem ?
- Identifique exemplos de ferramentas case e para que são aplicadas.
- Escolha duas delas que possuem aplicação para o mesmo propósito e realize um benchmark entre elas identificando os pontos positivos e negativos de cada uma delas.



# FERRAMENTAS CASE

A sigla CASE significa “Computer Aided Software Engineering”, em português: “Engenharia de Software Auxiliada por Computador”.

**Ferramentas CASE são ferramentas utilizadas como suporte para desenvolver um software. Essas ferramentas oferecem um conjunto de serviços, fortemente relacionados, para apoiar uma ou mais atividades do processo de desenvolvimento de software e podem minimizar o tempo de desenvolvimento do programa, mantendo o alto nível de qualidade.**

As ferramentas CASE podem auxiliar no desenvolvimento de software desde a análise de requisitos e modelagem até programação e testes. Podem ser consideradas como ferramentas automatizadas que tem como objetivo auxiliar o desenvolvedor de sistemas em uma ou várias etapas do ciclo de desenvolvimento de software.



# FERRAMENTAS CASE

As ferramentas CASE podem ser classificadas em:

- Horizontais: oferecem serviços utilizados durante todo o processo de software;
- Verticais: utilizadas em fases específicas do processo de software.

Elas também podem ser classificadas de acordo com os serviços que oferecem, dentre as quais, cita-se.

- Documentação;
- Planejamento e gerenciamento de projetos;
- Especificações formais;
- Comunicação;
- Análise e projeto de software;
- Projeto e desenvolvimento de interfaces;
- Suporte a Programação;
- Gerenciamento de Configuração;
- Controle de Qualidade;
- Testes de software;
- Depuração;
- Reengenharia



# Engenharia de Software II

**Aula: Modelagem de  
Caso de Uso**



Prof. Anderson Augusto Bosing

# **Modelagem de Caso de Uso**

O modelo de casos de uso (MCU) é uma representação das funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema que interagem com ele. O MCU é um modelo de análise que representa um refinamento dos requisitos funcionais do sistema em desenvolvimento.

A ferramenta da UML utilizada na modelagem de casos de uso é o diagrama de casos de uso.



# **Modelo de Casos de Uso(MCU)**

**O MCU representa os possíveis usos de um sistema da maneira como são percebidos por um observador externo a este sistema.**

**Cada um desses usos está associado a um ou mais requisitos funcionais identificados para o sistema.**

**É um modelo de análise que representa um refinamento dos requisitos funcionais.**

**Possui diversos componentes: casos de uso, atores e relacionamentos entre eles.**



# O que é Modelagem de Caso de Uso?

- Faz a ligação entre as necessidades dos envolvidos para os requisitos de software.
- Define limites claro para um sistema.
- Captura e comunica o comportamento desejado do sistema.
- Identifica quem ou o que interage com o sistema.
- Valida/Verifica os requisitos.
- Um instrumento de planejamento.



# O que é um Caso de Uso ?

**Um caso de uso define uma sequência de ações a serem executadas pelo sistema para produzir um resultado de valor observável para um ator.(RUP)**

**Usamos casos de uso para captar os comportamentos pretendidos de um sistema, sem especificar como esse comportamento é implementado.**



# **Casos de Uso são comumente identificados por nomes ou identificadores.**

**Todo caso de uso possui um nome que o identifica e diferencia dos demais casos de uso do sistema.**

**O nome é uma sequência de caracteres de texto e deve ser único no pacote que o contém.**

**No geral, os nomes são expressões verbais ativas, que nomeiam um comportamento específico do sistema.**

**Exemplos: Registrar Venda, Fazer Pedido, Manter Usuários, Manter Produtos.**



# **Casos de Uso são comumente identificados por nomes ou identificadores.**

**Os identificadores representam através de um código o caso de uso a que se referenciam.**

**Exemplo:**

**UC.001 - Registrar Venda**

**UC.002 - Fazer Pedido**

**UC.003 - Manter Usuários**

**UC.004 - Manter Produtos**



# O que é um Caso de Uso ?

Um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema.

Representa um relato de uso de certa funcionalidade do sistema em questão, sem revelar a estrutura e o comportamento internos desse sistema.



# O que é um Caso de Uso ?

Cada caso de uso de um sistema se define pela descrição narrativa das interações que ocorrem entre o elemento externo e o sistema.

A UML não define uma estrutura textual a ser utilizada na descrição de um caso de uso.

Há vários estilos de descrição propostos para definir casos de uso.



**Podemos dizer que há três dimensões em que o estilo de descrição de um caso de uso pode variar:**

- ✓ **Formato**
- ✓ **Grau de detalhamento**
- ✓ **Grau de abstração**



# **Formato**

**O formato de uma descrição de caso de uso diz respeito à estrutura utilizada para organizar a sua narrativa textual.**

**Os formatos comumente utilizados são:**

- ✓ Contínuo**
- ✓ Numerado**
- ✓ Tabular**



# **Formato Contínuo – Exemplo Realizar Saque**

**Este caso de uso inicia quando o cliente chega ao caixa eletrônico e insere seu cartão. O sistema requisita a senha do cliente. Após o cliente fornecer sua senha e esta ser validada, o sistema exibe as opções de operações possíveis. O cliente opta por realizar um saque. Então o sistema requisita o total a ser sacado. O cliente fornece o valor da quantidade que deseja sacar. O sistema fornece a quantia desejada e imprime o recibo para o cliente. O cliente retira a quantia e o recibo, e o caso de uso termina.**



# **Formato Numerado - Exemplo Realizar Saque**

- 1) Cliente insere seu cartão no caixa eletrônico.**
- 2) Sistema apresenta solicitação de senha.**
- 3) Cliente digita senha.**
- 4) Sistema valida a senha e exibe menu de operações disponíveis.**
- 5) Cliente indica que deseja realizar um saque.**
- 6) Sistema requisita o valor da quantia a ser sacada.**
- 7) Cliente fornece o valor da quantia que deseja sacar.**
- 8) Sistema fornece a quantia desejada e imprime o recibo para o cliente.**
- 9) Cliente retira a quantia e o recibo, e o caso de uso termina.**



# Formato Tabular - Exemplo Realizar Saque

Cliente	Sistema
Insere seu cartão no caixa eletrônico.	Apresenta solicitação de senha.
Digita senha.	Valida senha e exibe menu de operações disponíveis.
Solicita realização de saque.	Requisita a quantia a ser sacada.
Fornece o valor da quantia que deseja sacar.	Fornece a quantia desejada e imprime o recibo para o cliente
Retira a quantia e o recibo.	



# Grau de Detalhamento

O grau de detalhamento a ser utilizado na descrição de um caso de uso pode variar desde o mais sucinto até a descrição com vários detalhes (expandido).

Um caso de uso sucinto descreve as interações entre ator e sistema sem muitos detalhes.

Um caso de uso expandido descreve as interações em detalhes.



# Grau de Abstração

O grau de abstração de um caso de uso diz respeito à existência ou não de menção a aspectos relativos à tecnologia durante a descrição desse caso de uso.

**Essencial:** é completamente desprovido de características tecnológicas.

**Real:** a descrição das interações cita detalhes da tecnologia a ser utilizada na interação entre o ator e o sistema.



# **Descrição Essencial – Exemplo (numerada)**

- 1) Cliente fornece sua identificação.**
- 2) Sistema identifica o usuário.**
- 3) Sistema fornece opções disponíveis para movimentação da conta.**
- 4) Cliente solicita o saque de determinada quantia.**
- 5) Sistema requisita o valor da quantia a ser sacada.**
- 6) Cliente fornece o valor da quantia que deseja sacar.**
- 7) Sistema fornece a quantia desejada.**
- 8) Cliente retira dinheiro e recibo, e o caso de uso termina.**



# **Descrição Real – Exemplo já utilizado (numerada)**

- 1) Cliente insere seu cartão no caixa eletrônico.**
- 2) Sistema apresenta solicitação de senha.**
- 3) Cliente digita senha.**
- 4) Sistema valida a senha e exibe menu de operações disponíveis.**
- 5) Cliente indica que deseja realizar um saque.**
- 6) Sistema requisita o valor da quantia a ser sacada.**
- 7) Cliente fornece o valor da quantia que deseja sacar.**
- 8) Sistema fornece a quantia desejada e imprime o recibo para o cliente.**
- 9) Cliente retira a quantia e o recibo, e o caso de uso termina.**



# Casos de Uso contém Requisitos de Software

**Cada caso de uso:**

- Descreve ações que o sistema faz para entregar algo de valor para um agente.
- Mostra a funcionalidade do sistema que o agente usa.
- Modela um diálogo entre o sistema e os agentes.
- É um completo e significativo fluxo de eventos da perspectiva de um agente em particular.



# Benefícios dos Casos de Uso

- Dá um contexto para os requisitos

Coloca os requisitos do sistema em sequências lógicas.

Ilustra o porque o sistema é necessário.

Ajuda a verificar se todos os requisitos foram capturados.

- São fáceis de entender.

Usam terminologia que clientes e usuários entendem.

Descreve a história concreta de uso do sistema.

Verifica o entendimento dos envolvidos.

- Facilita o acordo com os clientes.

- Facilita o reuso: teste, documentação e design.



# **Engenharia de Software II**

**Aula: Especificação de  
Caso de Uso e Regras de  
Negócios.**



Prof. Anderson Augusto Bosing

# **Revisando a Ultima Aula**

- **Estudo de Caso de Diagrama de Caso de Uso e Modelo de Caso de Uso.**
- **Sentido das ligações.**
- **Nomes dos Casos de uso em verbos do infinitivo.**



# **Regras de Negócios**

**São padrões que condicionam o funcionamento do negócio, sendo comumente aplicadas no contexto da arquitetura de softwares.**

**No entanto, não ficam restritas a isso. Você pode e deve adotá-las como parâmetro para todas as atividades produtivas da empresa.**



# **Regras de Negócios**

**Em essência são regras da organização que está construindo o sistema e que refletem na forma como o sistema deve funcionar.**



# Regras de Negócios

**Um negócio funciona por processos que, por sua vez, são formados por atividades relacionadas entre si. As funções das áreas de compras, estoque, logística, finanças, vendas e marketing, por exemplo, compõem um processo de fornecimento de um produto ao cliente.**



# Regras de Negócios

Dentro destes processos, existem regras que devem ser seguidas durante a execução das atividades, que ajudam a definir **COMO** as operações devem ser realizadas e gerenciadas, por **QUEM**, **QUANDO**, **ONDE** e **POR QUÊ**, de acordo com a definição do BPM CBOK. Podemos dizer que as regras de negócio são limites impostos às operações, de forma que elas sigam corretamente em direção às políticas e aos objetivos da instituição.



# **Exemplos de Regras de Negócios**

**Em um controle de qualidade de granja, pode-se dizer que a cada 100 ovos impróprios para consumo, o lote será descartado.**

**Em um banco, clientes com faturamento mensal de mais de R\$ 25 mil e CPF sem restrições, serão atendidos pelo gerente Premium pessoa física.**



# **Exemplos de Regras de Negócios**

**Para conclusão de licitações, devem ser feitos três orçamentos e o vencedor será sempre o de menor preço final.**

**Em um processo de seleção de RH, o candidato só pode ser aprovado se tiver mais de 5 anos de experiência na área, diploma de pós-graduação, espanhol fluente e pretensão salarial abaixo de R\$ 8.000,00.**



# Especificação de Casos de Uso

Como vimos, caso de uso é definido através da descrição das interações que ocorrem entre as entidades externas e o sistema.

A descrição ou especificação é uma forma de validar se o entendimento de analistas de sistemas e usuários são os mesmos; e serve de base para a criação dos planos de testes do sistema.

As especificações dos casos de uso são feitas normalmente em linguagem natural e existem diversões padrões estruturais, mas não um único padronizado pela UML.



# Especificação de Casos de Uso

**Cada caso de uso possui um conjunto de ações que precisam ser executadas para que o objetivo da funcionalidade seja alcançado;**

**No caso de efetuar venda: identificar o vendedor, identificar o produto, a quantidade vendida, etc;**

**Essas ações constituem os fluxos que são realizados pelos casos de uso para disponibilizar o resultado desejado.**



# **Pré-Condições**

**Pré-condições:**

**São condições que devem ser verdadeiras para um caso de uso se iniciar.**

**Normalmente envolvem um estado de sistema, uma condição temporal ou estão relacionados a permissões de usuário;**

**Exemplos:**

- 1. Solicitação de mudança deve estar no estado “Pendente Aprovação”**
- 2. Deve ser 01:00**
- 3. O usuário logado deve ter perfil “Administrador”**



# Fluxos de Eventos

- Provavelmente a parte mais importante de um caso de uso.
- Define a sequencia de ações entre o ator e o sistema.
- É uma sequencia de comandos declarativos que descreve as etapas de execução de um caso de uso.



# Fluxos de Eventos

O fluxo de eventos de um caso de uso é composto por:

- **Fluxo Principal:** descreve a funcionalidade principal do caso de uso, quando nenhum desvio é tomado.
- **Fluxos Alternativos:** descrevem desvios pré-definidos ao fluxo principal.
- **Fluxos Excepcionais:** descrevem desvios do fluxo principal que de alguma forma geram exceções na aplicação como erros não previstos ou até mesmo validações.



# Exemplo de Fluxo Principal

1. [EV] O Cliente insere o cartão no caixa.
2. [RS] O sistema realiza a leitura do cartão e solicita a senha ao cliente.
3. [EV] O Cliente entra com a senha.
4. [RS] O sistema valida a senha e exibe as opções de menu.
5. [EV] O cliente escolhe a opção “Sacar”.
6. [RS] O sistema solicita ao cliente a quantidade a ser sacada.
7. [EV] O cliente informa a quantia desejada e clica na opção “Confirmar”.



# Exemplo de Fluxo Principal

8. [RS] O sistema verifica se o cliente possui saldo suficiente para sacar a quantia informada e entrega a quantia sacada.
9. [EV] Cliente retira a quantidade solicitada.
10. [RS] Sistema solicita a remoção do cartão do cliente e a operação é finalizada.



# **Exemplo de Fluxo Alternativo**

**8a.1 [RS]** O sistema verifica que o cliente não possui saldo suficiente e exibe um menu para realização de uma saque de adiantamento ao salário do cliente.

**8a.2 [EV]** O cliente aciona a opção de “Aceitar”.

**8a.3** Retorna ao fluxo principal no passo 9.



# **Exemplo de Fluxo Excepcional**

**A senha informada é incorreta.**

**4a.1 [RS] O sistema apresenta a mensagem de erro “A senha informada está incorreta. Verifique!”**

**4a.2 Retorna ao fluxo principal no passo 3.**



# **Mas o que é [EV] e [RS] ?**

**WAZLAWICK (2004) sugeriu uma forma de descrição dos casos de uso, nomeado como passos obrigatórios. Estes passos obrigatórios incluem os passos que indicam de alguma forma uma troca de informação entre os atores e o sistema.**

**Existem dois tipos de passos obrigatórios:**

- a) Eventos de Sistema[EV]: passos que mostram alguma informação indo de um ator para o sistema.**
- b) Respostas do Sistema[RS]: passos que mostram alguma informação vindo do sistema para o(s) ator(es)**



# **Descrição Essencial e Real do Caso de Uso**

A descrição essencial é descrita por COCKBURN (2005) como “deixe de fora a interface do usuário; focalize a intenção”.

WAZLAWICK (2004, p.63) apresenta como “o que” acontece entre o usuário e o sistema e não “como” isto ocorre.



# **Descrição Essencial e Real do Caso de Uso**

**Na descrição real os casos de uso são descritos em estilo real, ou seja, “como” realmente irá acontecer, sua realização concreta (WAZLAWICK, 2004, p.63-67).**

**Nesse caso descrevemos o processo em termos de projeto real, comprometido com tecnologias de desenvolvimento, interfaces de entrada e saída. Fluxos de eventos representados diretamente no sistema.**

