

Linguagem de Programação 1

Processo de Desenvolvimento de Software

Prof. Fábio José Rodrigues Pinheiro

FEVEREIRO DE 2022



**INSTITUTO
FEDERAL**

Catarinense

Campus
Videira

Quando um software é bem-sucedido

- Atende às necessidades dos usuários
- Opera perfeitamente durante um longo período
- Fácil de modificar
- Fácil de usar



Quando um software é bem-sucedido

- Atende às necessidades dos usuários
- Opera perfeitamente durante um longo período
- Fácil de modificar
- Fácil de usar



Quando um software falha

- Usuários insatisfeitos
- Propenso a erros
- Difícil de modificar
- Difícil de usar



O desenvolvimento de software é uma atividade complexa

Grande número de projetos que não chegam ao fim, extrapolam recursos financeiros e de tempo



O desenvolvimento de software é uma atividade complexa

Grande número de projetos que não chegam ao fim, extrapolam recursos financeiros e de tempo

Processo de desenvolvimento de software

Compreende todas as atividades necessárias para definir, desenvolver, testar e manter um software

- Define **quais** atividades serão executadas ao longo do projeto, **quando, como** e por **quem** tais atividades serão executadas



Componente humano de um projeto

■ Gerente de projeto

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

■ Analista de negócio

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

■ Projetista

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

■ Desenvolvedor (programador)

- Responsável pela codificação do sistema

■ Cliente

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



A interação entre as partes



Como o cliente explicou...



Como o líder de projeto entendeu...



Como o analista projetou...



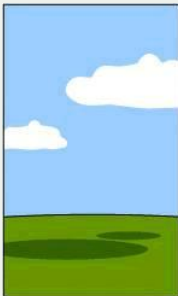
Como o programador construiu...



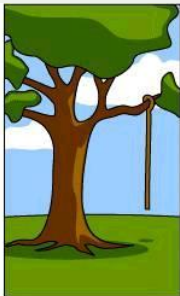
Como o Consultor de Negócios descreveu...



A interação entre as partes



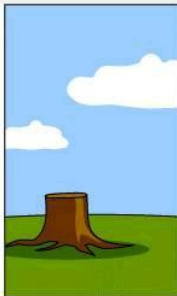
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...



Requisito

Condição ou capacidade que deve ser alcançada por um sistema para atender um contrato ou especificação



Requisito

Condição ou capacidade que deve ser alcançada por um sistema para atender um contrato ou especificação

- Ex: *Alunos poderão fazer matrícula nas disciplinas do semestre letivo por meio de uma interface web*



Requisito

Condição ou capacidade que deve ser alcançada por um sistema para atender um contrato ou especificação

- *Ex: Alunos poderão fazer matrícula nas disciplinas do semestre letivo por meio de uma interface web*

1 Levantamento de requisitos

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



Requisito

Condição ou capacidade que deve ser alcançada por um sistema para atender um contrato ou especificação

- *Ex: Alunos poderão fazer matrícula nas disciplinas do semestre letivo por meio de uma interface web*

1 Levantamento de requisitos

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

2 Análise de requisitos (ou Especificação de requisitos)

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



Requisito

Condição ou capacidade que deve ser alcançada por um sistema para atender um contrato ou especificação

- *Ex: Alunos poderão fazer matrícula nas disciplinas do semestre letivo por meio de uma interface web*

1 Levantamento de requisitos

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

2 Análise de requisitos (ou Especificação de requisitos)

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

3 Projeto

- Determina **como** o sistema funcionará para atender os requisitos, de acordo com os recursos tecnológicos existentes



4 Implementação

- Ocorre a tradução da descrição computacional da fase de projeto em código executável através do uso de linguagens de programação

5 Testes

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

6 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



Levantamento de requisitos

O que o usuário necessita do sistema?

Levantamento de requisitos

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



O que o usuário necessita do sistema?

Levantamento de requisitos

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

- Tarefa relativamente simples, pois o cliente sabe o que deseja e possui um bom conhecimento sobre as características e funções que lhe trarão benefícios



O que o usuário necessita do sistema?

Levantamento de requisitos

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

- Tarefa relativamente simples, pois o cliente sabe o que deseja e possui um bom conhecimento sobre as características e funções que lhe trarão benefícios
 - Geralmente não é verdade!



O que o usuário necessita do sistema?

Levantamento de requisitos

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

- Tarefa relativamente simples, pois o cliente sabe o que deseja e possui um bom conhecimento sobre as características e funções que lhe trarão benefícios
 - Geralmente não é verdade!

Cliente falando para o Analista...

Eu sei que você imagina que entendeu aquilo que eu lhe disse, mas o que você não compreende é que aquilo que eu lhe disse não era exatamente aquilo que eu quis dizer



Entrevista com a parte interessada (*stakeholder*)

- Ajuda identificar requisitos que ainda fazem parte do escopo do projeto ou mesmo requisitos que são contraditórios

Uma forma de fazer

- Pode ser feita no local de trabalho do entrevistado
- Identificar as pessoas chaves
- Não induzir respostas
- Tomar nota de tudo que foi discutido



Entrevista com a parte interessada (*stakeholder*)

- Ajuda identificar requisitos que ainda fazem parte do escopo do projeto ou mesmo requisitos que são contraditórios

Alguns exemplos de perguntas

■ Perguntas iniciais

- Quem irá usar a solução?
- Quais problemas essa solução irá tratar?
- Você poderia descrever o ambiente de negócio em que a solução será usada?

■ Perguntas finais

- Você é a pessoa correta para responder essas perguntas? Suas respostas são oficiais?
- Eu deixei de perguntar algo que você acha importante?
- Alguma outra pessoa poderia me prestar informações adicionais?



Brainstorm – tempestade de ideias

- Explorar ideias de um grupo de pessoas sem fazer qualquer tipo de julgamento ou crítica sobre as ideias expostas (todas ideias são aceitas)
- O objetivo é fazer com que o grupo de pessoas compartilhe seu conhecimento sobre o negócio

Uma forma de fazer

- Apresente um tema e os participantes discutem sobre o mesmo
- Todas ideias precisam ser escritas e expostas (e.g. em um mural)
- Analisar as ideias levantadas e selecionar as melhores



Questionário

- Adequado quando se tem um grande número de pessoas das quais precisa se extrair as mesmas informações
- Questionário encaminhado por escrito e a ideia é colher todas as opiniões sobre um mesmo conjunto de questões

Leitura de documentação

- Manuais de procedimentos, processos mapeados, relatórios, etc



■ Requisitos funcionais

- Definem as funcionalidades do sistema (o que o sistema deve fazer)

■ Requisitos não-funcionais

- Declaram características de qualidade que o sistema deve possuir
- Desempenho, Confiabilidade, Portabilidade, Segurança, Usabilidade

■ Restrições

- Impostas sobre o desenvolvimento do sistema
- Adequação a custos e prazos, aspectos legais (licenciamento), etc.



Requisitos funcionais

Sistema Acadêmico – Exemplo no Livro de Eduardo Bezerra

- R.1 – O sistema deve permitir que alunos visualizem as notas obtidas por semestre letivo
- R.2 – O sistema deve permitir o lançamento das notas das disciplinas lecionadas em um semestre letivo e controlar os prazos e atrasos neste lançamento
- R.3 – O sistema deve permitir que os alunos realizem a inscrição em disciplinas de um semestre letivo
- R.4 – O sistema deve manter informações cadastrais sobre os alunos e sobre seus históricos escolares
- R.n – ...



■ Requisitos não funcionais

- A interface de usuário deve ser simplificada e evitar múltiplos cliques (passos) para chegar em determinada funcionalidade
- Fazer uso de protocolos criptográficos para troca de mensagens
- O sistema deverá ser desenvolvido em uma linguagem para *web* e que não dependa de plugins para o navegador

■ Restrições

- O sistema precisa estar em produção antes do início do próximo semestre letivo
- O banco de dados precisa ser o MySQL
- Regras de negócio...



- São políticas, condições ou restrições que devem ser consideradas na execução dos processos existentes de uma organização
- Regras de negócio tem influência sobre a lógica de execução de um ou mais **casos de uso**
- Regras são documentadas em um **modelo de regras de negócio**, de forma que cada regra deva receber um identificador único
- A descrição do modelo pode ser feita utilizando **texto informal** ou alguma **forma de estruturação**



- Um professor só pode lecionar disciplinas para as quais esteja habilitado
- Um correntista não pode sacar mais de R\$ 1.000,00 por dia de sua conta
- Senhas devem ter no mínimo 6 caracteres, entre números e letras, e devem ser atualizadas a cada três meses



RN01 – Quantidade máxima de inscrições por semestre letivo

| | |
|-----------|--|
| Descrição | Em um semestre letivo, um aluno não pode se inscrever em uma quantidade de disciplinas cuja soma de créditos ultrapasse 20 |
|-----------|--|

RN02 – Quantidade de alunos possíveis

| | |
|-----------|---|
| Descrição | Uma oferta de disciplina não pode ter mais de 40 alunos inscritos |
|-----------|---|

RN03 – Cancelamento de matrícula

| | |
|-----------|---|
| Descrição | Um aluno deve ter a matrícula cancelada se for reprovado mais de duas vezes na mesma disciplina |
|-----------|---|



- Declaração da necessidade e viabilidade
- Declaração do escopo do sistema ou produto
- Lista de clientes, usuários e outros interessados que participaram do levantamento de requisitos
- Descrição do ambiente técnico do sistema
- Lista de requisitos e modelo com regras de negócio
- Conjunto de cenários de uso que esclarecem o uso do sistema sob diferentes condições operacionais



Análise de requisitos

Requisito

Condição ou capacidade que deve ser alcançada por um sistema para atender um contrato ou especificação

Análise

- separação de um todo em seus elementos ou partes componentes
- estudo pormenorizado de cada parte de um todo, para conhecer melhor sua natureza, suas funções, relações, causas, etc.
- processo ou método com que se descreve, caracteriza e compreende algo



Estudo detalhado dos requisitos levantados e a **construção de um conjunto de modelos** de análise para representar o sistema a ser construído

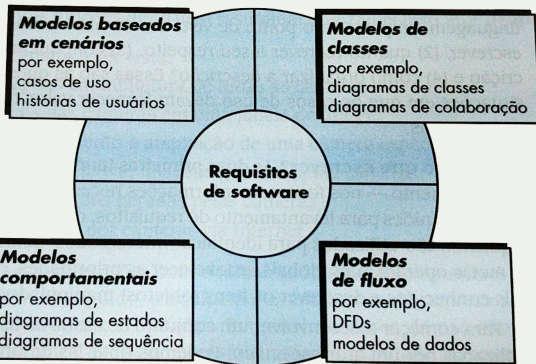
- Ponte entre a “Descrição do Sistema” e o “Modelo de projeto”
 - Documento resultante da análise deve especificar tudo o que deverá ser projetado e construído
-
- **Modelo de requisitos deve**
 - 1 Descrever o que o cliente solicitou
 - 2 Estabelecer uma base para a criação de um projeto de software
 - 3 Definir um conjunto de requisitos que possa ser validado assim que o software for construído



- Modelo deve ter **foco nas necessidades visíveis** no domínio do problema ou do negócio – nível de abstração elevado
- **Cada elemento no modelo** de requisitos deve **contribuir para o entendimento** geral dos requisitos levantados
- Deixe considerações de infraestrutura e outros modelos não funcionais até a fase de projeto
- Certifique-se que o **modelo** de requisitos **agregará valor a todos os interessados**
 - Interessados no negócio devem usar o modelo para validar os requisitos
 - Projetistas devem usar o modelo como base para o projeto
 - Pessoal da Qualidade de Software deve usar o modelo para construção dos testes de aceitação



Validar requisitos – pode ser necessário analisá-los de diferentes pontos de vista



■ Cenários

- Sistema sob o ponto de vista do usuário

■ Classes

- Conjunto de objetos que são manipulados em cada cenário

■ Comportamentais

- Comportamento de um sistema, representação seus estados e eventos

■ Fluxo

- Como as informações são modificadas



Modelagem baseada em cenários

- Consiste em uma **representação das funcionalidades externamente observáveis do sistema** e dos **elementos externos ao sistema** que interagem com o mesmo
- Molda os requisitos funcionais do sistema



- Representa o software do ponto de vista do usuário final
- Um **caso de uso** consiste de uma **descrição narrativa das interações** entre um sistema e seus agentes externos (**atores**)
- Representa **quem faz o que com o sistema**, sem considerar o comportamento interno do sistema



- Ator é qualquer elemento **externo** que **interage** com o sistema
 - Externo, pois não faz parte do sistema
 - Interage através de troca de informações com o sistema (envia e/ou recebe)
- Ator **primário** é aquele que inicia uma sequência de interações de um caso de uso
- Ator **secundário** opera, mantém ou auxilia na utilização do sistema, existindo apenas para permitir que atores primários possam usar o sistema



- Ator é qualquer elemento **externo** que **interage** com o sistema
 - Externo, pois não faz parte do sistema
 - Interage através de troca de informações com o sistema (envia e/ou recebe)
- Ator **primário** é aquele que inicia uma sequência de interações de um caso de uso
- Ator **secundário** opera, mantém ou auxilia na utilização do sistema, existindo apenas para permitir que atores primários possam usar o sistema

Sistema modelado: navegador web

Para que o **usuário** (ator primário) requisiite uma página ao **navegador**, um outro ator (o secundário) está envolvido, no caso o **servidor web**.



- Um ator pode participar de muitos casos de uso e um caso de uso pode envolver vários atores
- Exemplos de atores (papéis que representam em relação ao sistema):
 - Pessoas – Cliente, Gerente, Vendedor, etc.
 - Organizações – Empresa, Fornecedora, Administradora de Cartões, etc.
 - Outros sistemas – Sistema Acadêmico, Sistema de Vendas, etc.
 - Equipamentos – Leitora de código de barras, sensores, câmeras, etc.



- Enumere as funções ou atividades realizadas por um ator específico
 - Obtida de uma lista funções dos requisitos do sistema, conversas com interessados, etc
- Geralmente casos de uso são escritos primeiro na forma de narrativa informal
- Se for necessário uma formalidade maior, então são reescritos usando um modelo estruturado



Cenário: proprietário acessa sistema de vigilância por câmeras via Internet

- Se eu estiver longe da minha residência eu poderia usar qualquer computador com navegador web para entrar no sistema CasaSegura. Devo fornecer meu nome de usuário e dois níveis de senha e, depois de validados, terei acesso a toda funcionalidade para o meu sistema instalado em casa.
- Para acessar a visão de uma câmera específica, seleciono “vigilância” no menu e depois escolho qual câmera desejo visualizar. Como alternativa eu poderia visualizar uma tela com miniatura de todas as câmeras que tenho instaladas clicando no menu “todas as câmeras”



Caso de uso: proprietário acessa sistema de vigilância por câmeras via Internet

Ator: Proprietário

- 1 O proprietário do imóvel faz login no site CasaSegura
- 2 O proprietário do imóvel introduz seu nome de usuário e as duas senhas (cada uma com pelo menos 8 caracteres)
- 3 O sistema mostra um menu de botões com as principais funções
- 4 O proprietário do imóvel seleciona “vigilância” no menu
- 5 O proprietário do imóvel seleciona “escolher câmera”
- 6 O sistema mostra a planta baixa da casa
- 7 O proprietário do imóvel seleciona o ícone da câmera desejada na planta baixa
- 8 ...



- **Caso de uso:** Realiza Saque
- **Ator primário:** Cliente
- **Resumo:** Realizar o saque de dinheiro no terminal de atendimento
- **Fluxo principal:**
 - 1 Cliente insere seu cartão no caixa eletrônico
 - 2 Sistema apresenta solicitação de senha
 - 3 Cliente digita a senha
 - 4 Sistema exibe menu de operações disponíveis
 - 5 Cliente indica que deseja realizar um saque
 - 6 Sistema requisita quantia a ser sacada
 - 7 Cliente retira o dinheiro



Etapas do exemplo do sistema de vigilância...

- 1 O proprietário do imóvel seleciona “escolher câmera”

- 2 O sistema mostra a planta baixa da casa

■ **O ator pode fazer algo diferente nesse ponto?**



Etapas do exemplo do sistema de vigilância...

- 1 O proprietário do imóvel seleciona “escolher câmera”
 - 2 O sistema mostra a planta baixa da casa
- **O ator pode fazer algo diferente nesse ponto?**
 - Poderia escolher “visualizar miniaturas de todas as câmeras”



Etapas do exemplo do sistema de vigilância...

- 1 O proprietário do imóvel seleciona “escolher câmera”
 - 2 O sistema mostra a planta baixa da casa
- **O ator pode fazer algo diferente nesse ponto?**
 - Poderia escolher “visualizar miniaturas de todas as câmeras”
 - **Ator poderia encontrar alguma condição de erro nesse ponto?**



Etapas do exemplo do sistema de vigilância...

- 1 O proprietário do imóvel seleciona “escolher câmera”
 - 2 O sistema mostra a planta baixa da casa
- **O ator pode fazer algo diferente nesse ponto?**
 - Poderia escolher “visualizar miniaturas de todas as câmeras”
 - **Ator poderia encontrar alguma condição de erro nesse ponto?**
 - Caso a planta baixa não tenha sido configurada será disparada uma mensagem de erro



Etapas do exemplo do sistema de vigilância...

1 O proprietário do imóvel seleciona “escolher câmera”

2 O sistema mostra a planta baixa da casa

■ **O ator pode fazer algo diferente nesse ponto?**

■ Poderia escolher “visualizar miniaturas de todas as câmeras”

■ **Ator poderia encontrar alguma condição de erro nesse ponto?**

■ Caso a planta baixa não tenha sido configurada será disparada uma mensagem de erro

■ **O ator pode encontrar algum outro tipo de comportamento nesse ponto?**



Etapas do exemplo do sistema de vigilância...

1 O proprietário do imóvel seleciona “escolher câmera”

2 O sistema mostra a planta baixa da casa

■ O ator pode fazer algo diferente nesse ponto?

- Poderia escolher “visualizar miniaturas de todas as câmeras”

■ Ator poderia encontrar alguma condição de erro nesse ponto?

- Caso a planta baixa não tenha sido configurada será disparada uma mensagem de erro

■ O ator pode encontrar algum outro tipo de comportamento nesse ponto?

- O sistema pode disparar um alarme (alguém entrou na casa) e apresentar para o ator opções para lidar com isso



- **Caso de uso:** Realiza Saque
- **Ator primário:** Cliente
- **Resumo:** Realizar o saque de dinheiro no terminal de atendimento
- **Fluxo principal:**
 - 1 Cliente insere seu cartão no caixa eletrônico
 - 2 Sistema apresenta solicitação de senha
 - 3 Cliente digita a senha
 - 4 Sistema exibe menu de operações disponíveis
 - 5 ...
- **Exceções:**
 - Foi informada senha incorreta. Veja o caso de uso “Validar Senha”



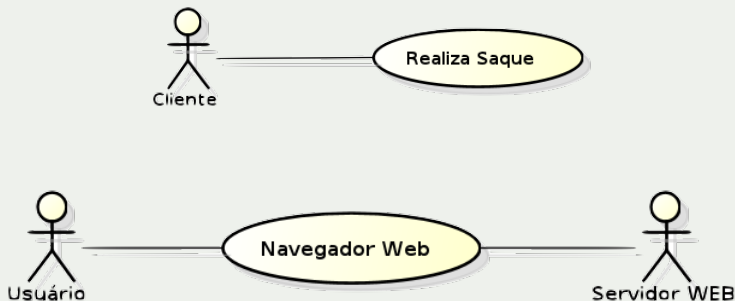
Diagrama de caso de uso UML

- Utilizado normalmente nas fases de levantamento e análise de requisitos
- Base para os demais diagramas da UML
- Linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar



Diagrama de casos de uso

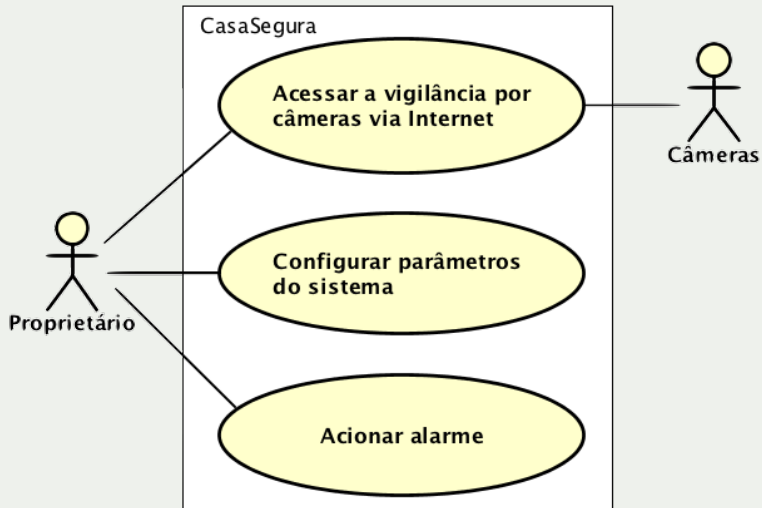
Corresponde a uma visão externa do sistema e representa graficamente atores, casos de uso e seus relacionamentos



Relacionamento de comunicação representa a informação de quais **atores** estão associadas a quais **casos de uso**

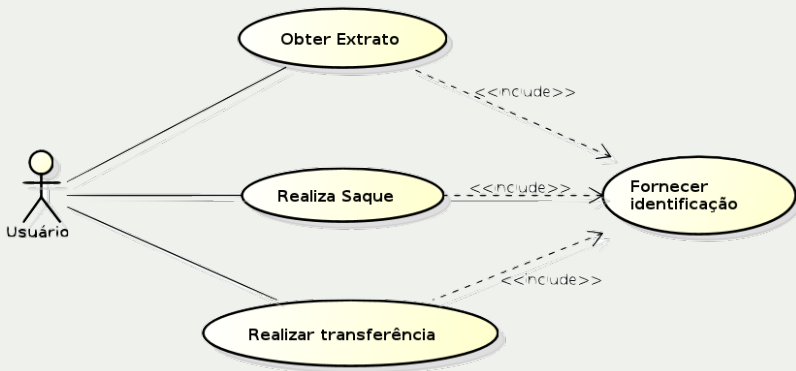


Diagrama de caso de uso: CasaSegura

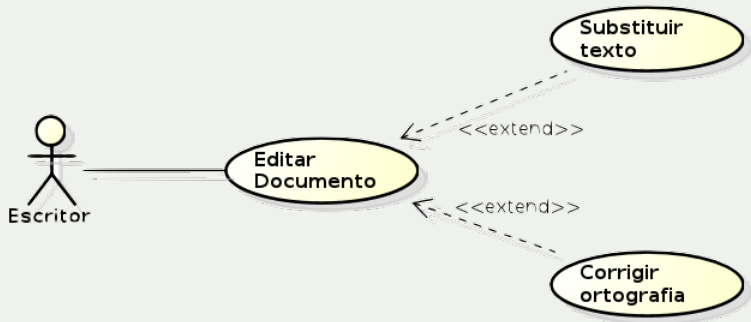


Relacionamentos – Inclusão <<include>>

- Para incluir um caso de uso dentro de outro
 - permitido somente entre casos de uso
- Deve ser usado quando mais de um caso de uso fizer uso de uma mesma sequência de instruções



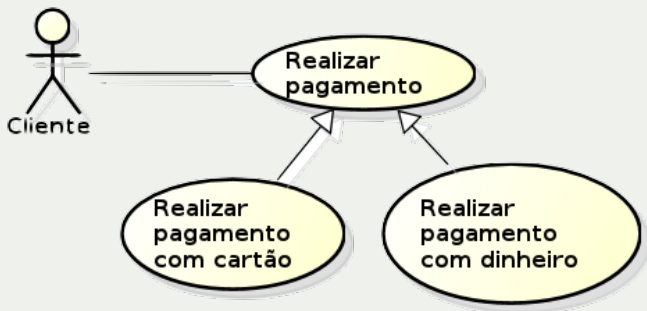
- Modelar situações em que diferentes sequências de interações podem ser inseridas em um caso de uso
- Cada uma dessas **diferentes sequências representa um comportamento opcional**, ou seja, que só ocorre sob certas condições, ou cuja realização depende da escolha do ator



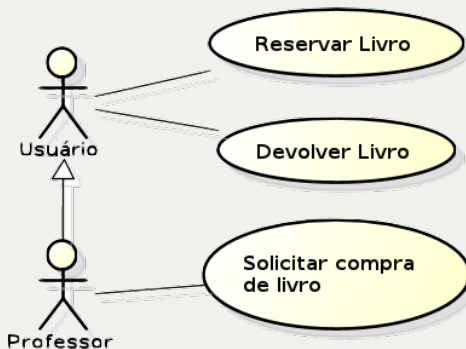
- Permite que um caso de uso (ou ator) herde características de um caso de uso (ou ator) mais genérico



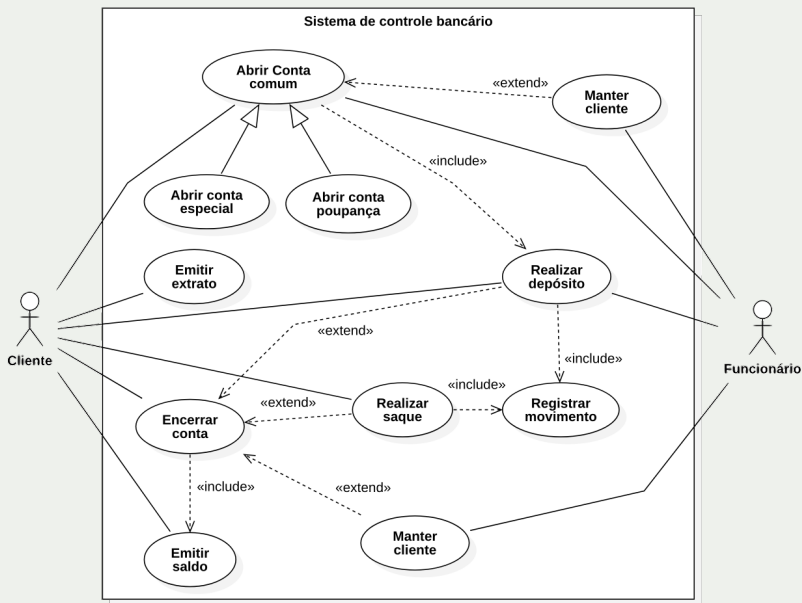
- Permite que um caso de uso (ou ator) herde características de um caso de uso (ou ator) mais genérico
- **Casos de uso:** se B herda de A, então B pode participar de qualquer relacionamento no qual A participa. B também pode redefinir a sequência de comportamento de A



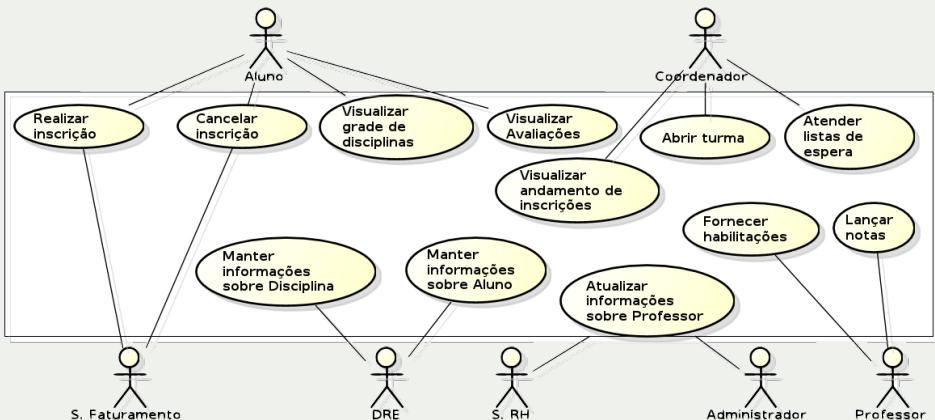
- Permite que um caso de uso (ou ator) herde características de um caso de uso (ou ator) mais genérico
- **Ator:** O ator herdeiro possui o mesmo comportamento do ator do qual ele herda e além disso pode participar de casos de uso em que o ator do qual herda não participa



Exemplo: Sistema de controle bancário



Exemplo: Sistema Acadêmico



Processo Unificado e desenvolvimento iterativo

- **Processo de desenvolvimento de software** descreve abordagem para construção, implantação e possivelmente manutenção de um software
- **Processo Unificado (PU)** é um processo iterativo de desenvolvimento flexível e adequado para POO
 - Práticas da *eXtreming Programming* (XP) e SCRUM podem ser empregadas no PU

Desenvolvimento iterativo

Ciclo de vida organizado em uma série de mini projetos curtos, de duração fixa (em semanas) chamados de **iterações**

- O produto de cada iteração é um sistema parcial, executável, possível de ser testado e integrável



- Enfrentar problemas que envolvem altos riscos e alto valor nas iterações iniciais
- Envolver continuamente os usuários na avaliação, na realimentação e nos requisitos
- Construir arquitetura central coesa nas iterações iniciais
- Verificar continuamente a qualidade, fazendo testes desde o início, com frequência e em situações realísticas



A finalidade da modelagem (diagramas UML) é principalmente entender, não documentar

- Não modele todo ou a maior parte do projeto. Adie problemas simples até a fase de codificação
- Use a ferramenta mais simples possível (e.g. quadro branco)
- Não modele sozinho, prefira modelar em pares ou trios
- Use uma notação simples para rascunhar. Detalhes exatos de UML não são importantes
- Saiba que todos modelos ficarão imprecisos e que o código final ficará diferente





GILLEANES T.A. GUEDES

UML2: UMA ABORDAGEM PRÁTICA

Editora Novatec, 2011

- Capítulo 3



EDUARDO BEZERRA

PRINCÍPIOS DE ANÁLISE E PROJETOS DE SISTEMAS COM UML

Editora Campus, 2003

- Fluxo Principal, Alternativos e de Exceção (página 66)
- Regras de negócio (página 70)
- Casos de Uso (páginas 81 até 88)



ROGER S. PRESSMAN

ENGENHARIA DE SOFTWARE: UMA ABORDAGEM PROFESSIONAL

Editora Bookman, 2011

- Capítulos 2, 5, 6 e 8





EDUARDO BEZERRA

PRINCÍPIOS DE ANÁLISE E PROJETOS DE SISTEMAS COM UML

Editora Campus, 2003



ROGER S. PRESSMAN

ENGENHARIA DE SOFTWARE: UMA ABORDAGEM PROFESSIONAL

Editora Bookman, 2011



CRAIG LARMAN

UTILIZANDO UML E PADRÕES

Editora Bookman, 2007

