

REQUISITOS DE SISTEMAS

CASOS DE USO

Olá!

Nesta aula, você irá: 1. Identificar a utilidade de casos de uso para a engenharia de software.

2. Reconhecer como o caso de uso contribuiu para a área de requisitos de sistemas.

3. Conhecer a estrutura dos casos de usos.

4. Obter uma introdução sobre a linguagem UML.

1 Introdução

Uma modelagem de um determinado sistema é um processo que consiste na representação de uma visão (ou perspectiva) do que se espera do sistema, no tocante ao seu funcionamento e resultado(s). É um consenso que ter uma representação visual de seu sistema antes que ele entre na etapa de implementação é de fundamental importância.

Desde meados da década de 80, o CASO DE USO estabelece uma metodologia que institui regras para a modelagem de sistemas. É como se fosse a planta baixa de uma residência. Antes que venhamos a construí-la, precisamos observar o desenho e demais informações, para evitarmos desagradáveis surpresas no o produto depois de terminado, ou seja, de nossa casa, por exemplo.

Representados por diagramas, os Casos de Uso tem o objetivo de auxiliar a comunicação entre os analistas e o cliente. Ele então expõe uma espécie de cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. Enfim, o cliente deve ter acesso através do diagrama de Casos de Uso a identificação das principais funcionalidades de seu sistema.

A partir dessa análise, conseguimos então perceber se estamos no caminho correto, portanto iremos atender os requisitos do sistema. Isso porque não estamos falando de uma conversa técnica ("bits e bytes"), mas com uma linguagem entendível por todos os integrantes da equipe, principalmente do solicitante, facilitando grandemente a comunicação.

Podemos também citar sobre uma segunda característica dessa modelagem, é que ela independe do tipo de plataforma tecnológica; ou seja, qual a linguagem de programação, qual o banco de dados etc. Lembra-se que falamos sobre isso no início da disciplina. Pois então, o Caso de Uso é uma estratégia muito peculiar a engenharia de requisitos.

Vamos agora aprender sobre sua estrutura.

O diagrama de Caso de Uso é compostos basicamente por 3 elementos. São eles:

- Atores; (Um ator é representado por um boneco e um rótulo com o nome do ator. Um ator identifica um usuário do sistema, seja ele humano ou outro sistema.)
- Casos de uso; (Um ator é representado por um boneco e um rótulo com o nome do ator. Um ator identifica um usuário do sistema, seja ele humano ou outro sistema.)
- Relacionamentos entre estes elementos.

Vamos a exemplos:

• **Representação do requisito “CADASTRAR CLIENTE”:**

Lembre-se que usamos a figura de “ator” para representar quaisquer entidades que interagem com o sistema.

Um ator representa um papel no sistema, mas um papel pode ser representando por vários atores.

Mediante o que sugere o diagrama, temos as seguintes ações do atores:

• **Ator “Cliente” poderá executar os casos de uso**

“realizar saque” e “consultar saldo”;

• **Ator “Gerente” poderá executar os casos de uso**

“abrir conta” e “vender seguro”.



2 Relacionamentos entre casos de uso

Mediante aspectos inerente a necessidade de fazer uso de casos de uso por outro caso de uso, estes podem se relacionar de duas formas:

- **Include**
- **Extends**

Include

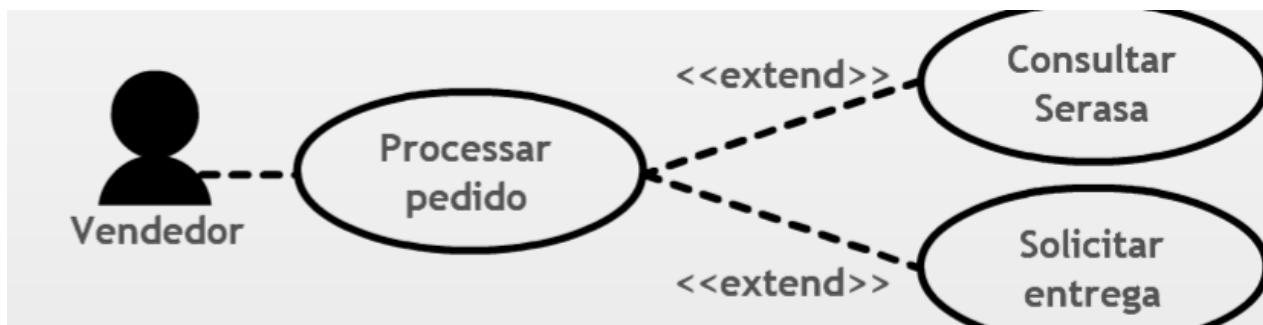
Quando um caso de uso “A” inclui (include) outro caso de uso “B”. Isto implica que ao executar o caso de uso “A” executa-se também o caso de uso “B”.



Neste exemplo, o ator “Vendedor” pode executar o caso de uso “Processar Pedido”, e também o caso de uso “Emitir nota Fiscal”, visto que os casos de uso possuem relacionamento. Contudo, o ator não pode executar o caso de uso “Emitir nota Fiscal” sem executar “Processar Pedido”.

Extends

Quando um caso de uso “A” tem um relacionamento do tipo extends com outro caso de uso “B”. Implica que ao executar o caso de uso “A” não necessariamente “B” será executado.



Neste cenário, o vendedor pode fazer uso de quaisquer um dos casos de uso de maneira independente. Ele executa o caso de uso “Consultar Serasa” e/ou “Solicitar Entrega”.

Atenção: Portanto, a diferença é que no include existe uma dependência do uso do caso de uso, o que não acontece quando o relacionamento é extend.

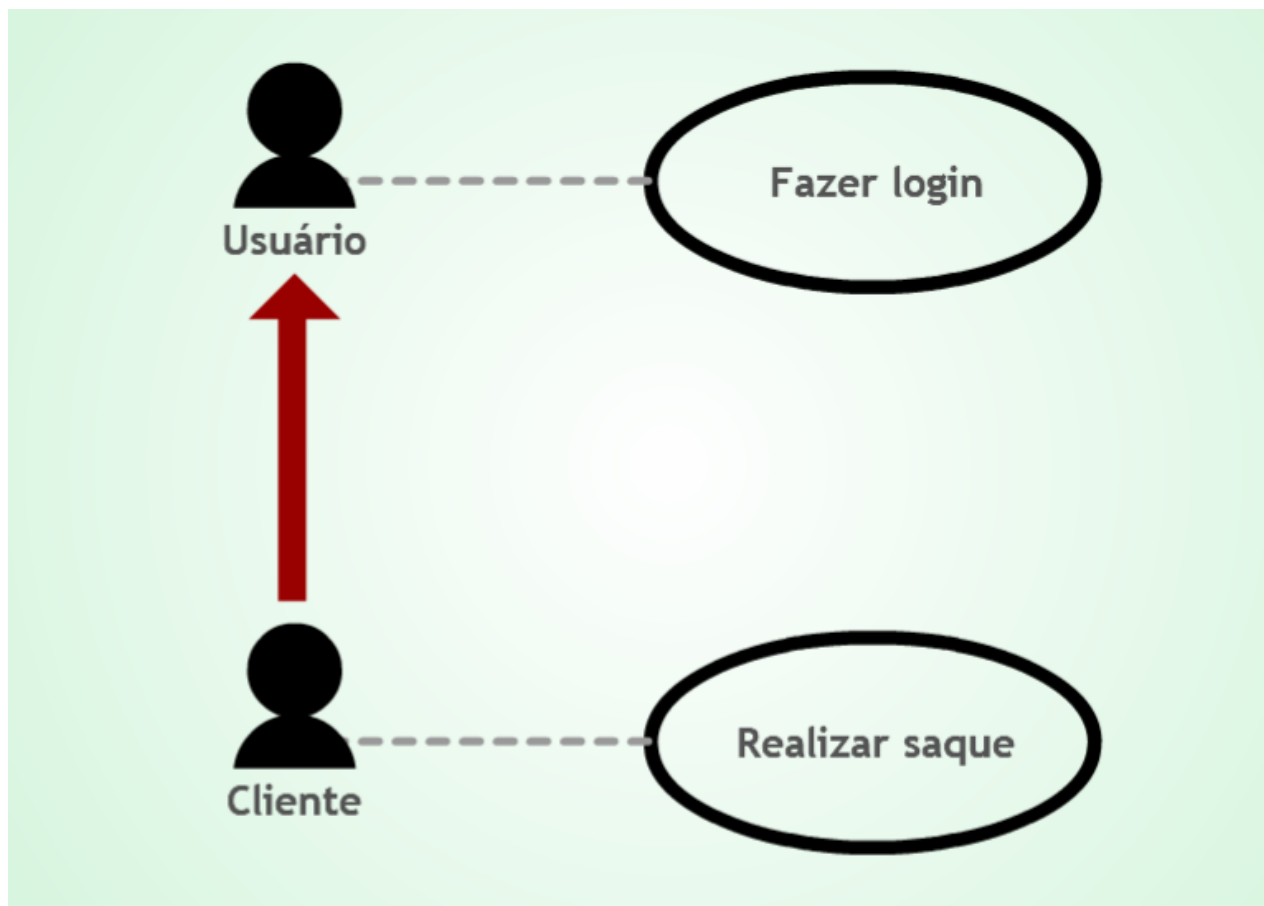
3 Relacionamento entre atores

O ator pode herdar as funcionalidades (casos de uso) de outro ator.

Aqui temos os Atores “usuário” e “Cliente”.

Ao ator “usuário” está liberado o acesso ao caso de uso “Fazer Login”. No caso do ator “Cliente”, ele pode executar o caso de uso “Realizar saque”, bem como “Fazer Login”, visto que ele possui também o privilégio de executar todos os casos de uso disponíveis ao ator “usuário”.

Neste exemplo, é importante verificar o sentido da seta, a qual indica que o sentido é do ator “Cliente” para o ator “usuário”.



4 Representação do sistema

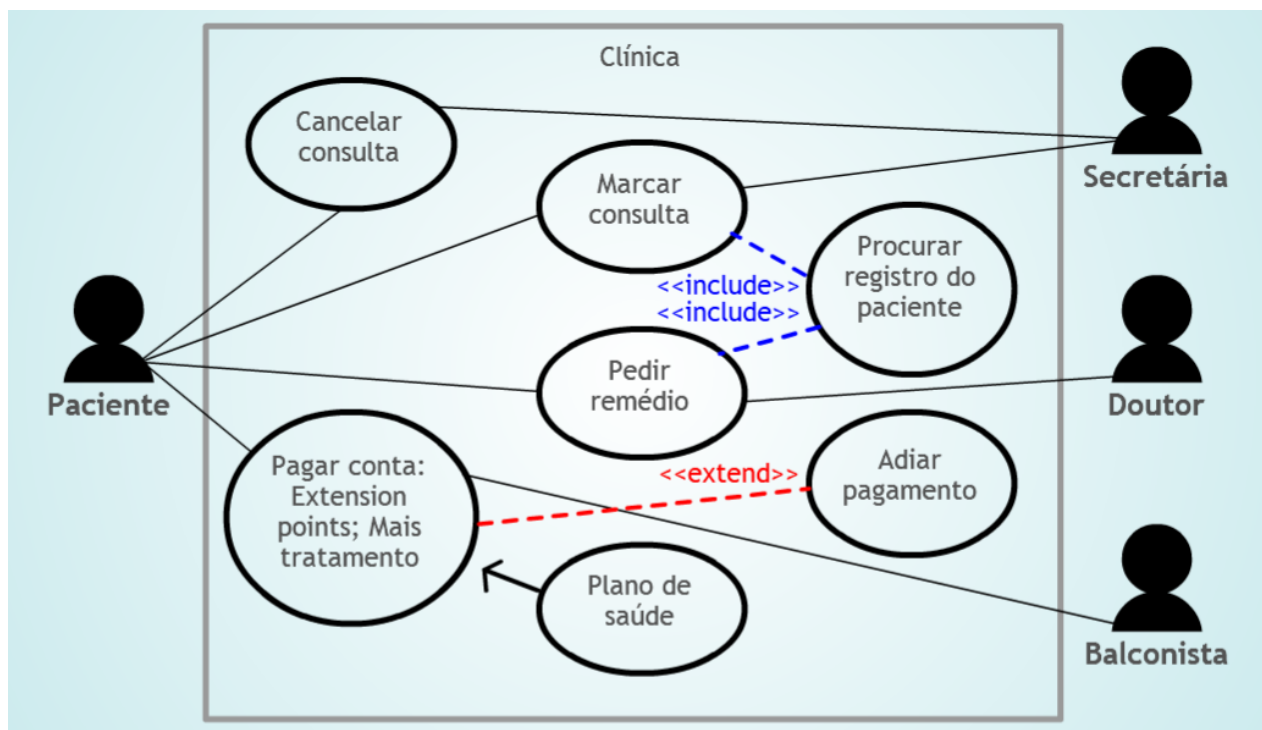
No tocante ao sistema como um todo, ou seja, a representatividade global do funcionamento é feito através de mais dois elementos:

Nome do sistema: Localizado dentro do retângulo.

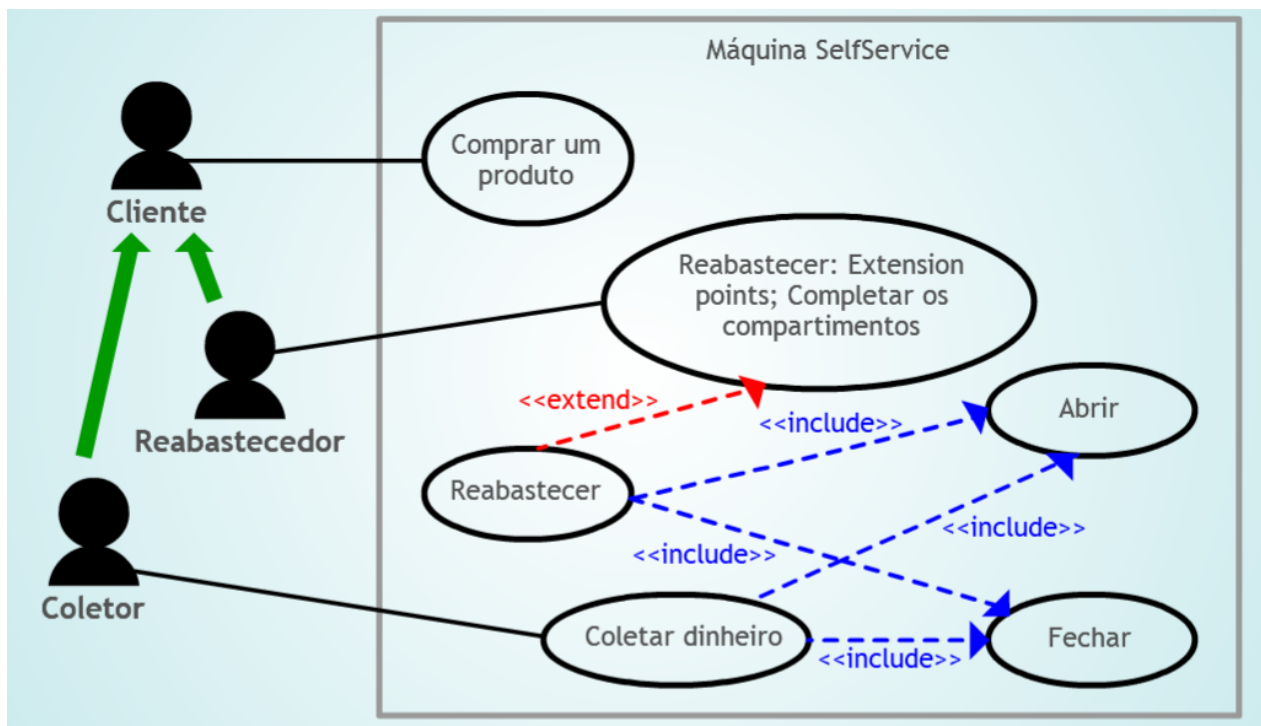
Limites do sistema: representado por um retângulo envolvendo os casos de uso que compõem o sistema.

Acompanhe os dois exemplos a seguir (visto do conhecimento já adquirido, descreva sua interpretação para cada cenário apresentado e remeta ao seu tutor).

Exemplo 1



Exemplo 2



Fonte: Fonte das imagens (todas): <http://celodemelo.wordpress.com/2007/03/17/entendendo-o-diagrama-de-casos-de-uso/>

5 A UML – Unified Modeling Language

A UML surgiu a partir de um incentivo (inclusive financeiro) da Rational Software na união entre outras três metodologias de modelagem. Foram eles: (a) o método do americano Grady Booch; (b) o método OMT (Object Modeling Technique) do sueco Ivar Jacobson; e (c) o método OOSE (Object-Oriented Software Engineering) do americano James Rumbaugh.

O esforço inicial do projeto começou com a união do método de Booch com o método OMT de Jacobson, o que resultou no lançamento do Método Unificado no final de 1995. Logo em seguida, Rumbaugh juntou-se a Booch e Jacobson na Rational Software e seu método OOSE começou a ser incorporado à nova metodologia. O trabalho de Booch, Jacobson e Rumbaugh conhecidos popularmente como "Os Três Amigos", resultou no lançamento, em 1996, da primeira versão da UML propriamente dita.

Assim que a primeira versão foi lançada, diversas grandes empresas atuantes na área de software passaram a contribuir com o projeto, fornecendo sugestões para melhorar e ampliar a linguagem. Finalmente a UML foi adotada pela OMG (Object Management Group) em 1997, como a linguagem padrão de modelagem. Hoje, em 2007, a UML está na versão 2.0.

Com uma modelagem, seu propósito é para permitir o entendimento e não para documentação. Ela então é parte do processo do desenvolvimento do sistema. Portanto, podemos utilizar os Casos de Usos para disseminação do conhecimento referente ao software a ser entregue.

Além do Caso de Uso que estudamos anteriormente, os seguintes tipos de diagramas são suportados pelo Umbrello UML Modeller:

- **Diagrama de Classe mostra classes e os relacionamentos entre elas**
- **Diagrama de Sequência mostra objetos e uma sequência das chamadas do método feitas para outros objetos.**
- **Diagrama de Colaboração mostra objetos e seus relacionamentos, colocando ênfase nos objetos que participam na troca de mensagens**
- **Diagrama de Estado mostra estados, mudanças de estado e eventos num objeto ou uma parte do sistema**
- **Diagrama de Atividade mostra atividades e as mudanças de uma atividade para outra com os eventos ocorridos em alguma parte do sistema**
- **Diagrama de Componente mostra os componentes de programação de alto nível (como KParts ou Java Beans).**
- **Diagrama de Distribuição mostra as instâncias dos componentes e seus relacionamentos.**
- **Os Diagramas de Entidade-Associação mostram os dados e as relações e as restrições entre os dados.**

Atenção: Vale uma consulta ao livro base dessa disciplina, e fazer uma verificação dos detalhes sobre cada um deles, bem como identificar suas diferenças, principalmente no tocante ao Caso de Uso - o qual foi detalhado em nossa aula de hoje.

CONCLUSÃO

Nesta aula, você:

- Compreendeu como os casos de uso contribuem para sucesso na identificação de requisitos.
- Aprendeu sobre a modelagem de sistemas através de casos de uso.
- Analisou o modelo para desenvolvimento de casos de uso.