



# Tecnologia em Análise e Desenvolvimento de Sistemas

## Engenharia de Software II

### UML – parte 2

Prof. Claudemir Santos Pinto  
profdemir@yahoo.com.br

# ***UML – Unified Modeling Language***

## ■ Diagrama de Caso de Uso:

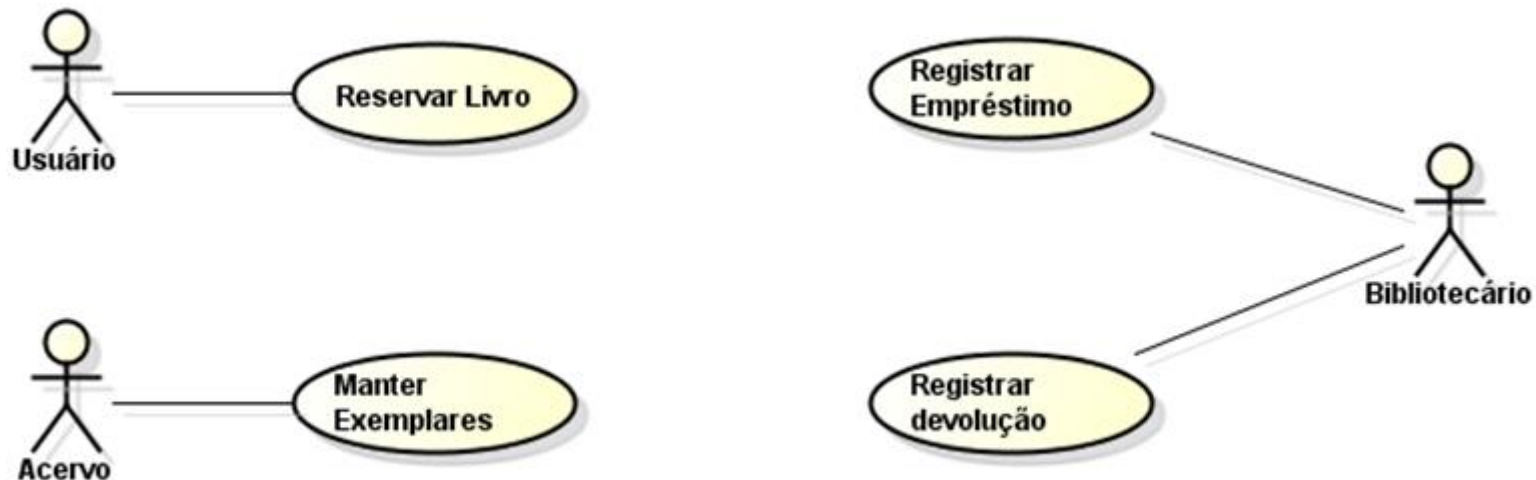
Permite dar uma visão global e de alto nível do sistema.  
É composto basicamente por quatro partes:

- **Cenário:** Sequência de eventos que acontecem quando um usuário interage com o sistema.
- **Ator:** Usuário do sistema, ou melhor, um tipo de usuário.
- **Caso de Uso (Use Case):** É uma tarefa ou uma funcionalidade realizada pelo ator (usuário)
- **Associação:** é o que liga um ator com um caso de uso

# ***UML – Unified Modeling Language***

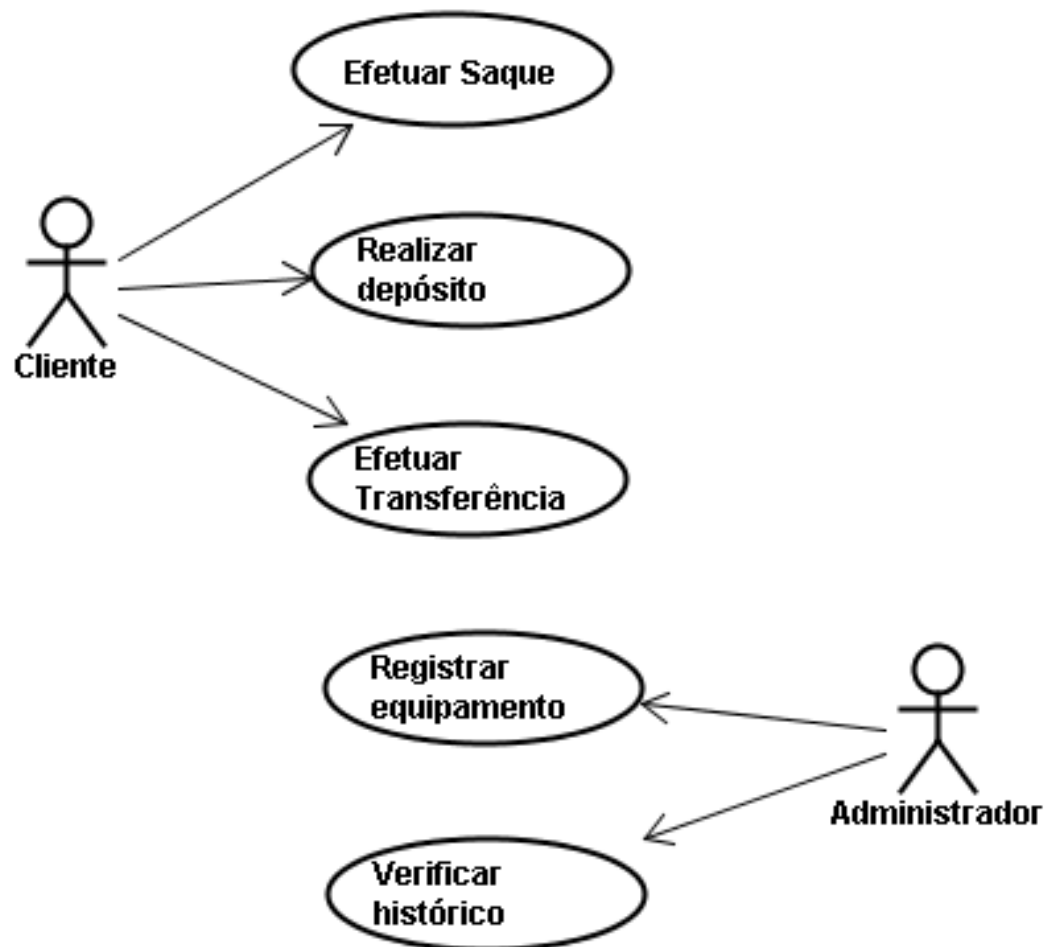
- Diagrama de Caso de Uso – Exemplos

## Sistema de biblioteca



# ***UML – Unified Modeling Language***

## Sistema de Caixa Eletrônico



# ***UML – Unified Modeling Language***

Vamos criar um **cenário** de exemplo para um diagrama de caso de uso:

“A clínica médica Saúde Perfeita precisa de um sistema de agendamento de consultas e exames. Quando acontece uma solicitação de consulta, a secretária procura data e hora disponível mais próxima na agenda do médico e marca a consulta. Eventualmente ela pode cancelar uma consulta. Durante a realização da consulta, o médico pode prescrever medicações e exames, caso necessário, e isso tudo deve ficar gravado no computador”.

# ***UML – Unified Modeling Language***

- Com esse cenário simples podemos começar a criar nosso diagrama. Inicialmente vamos definir nossos atores:
  - a) Secretária
  - b) Médico

# ***UML – Unified Modeling Language***

## ■ Encontrando Casos de Uso

Faça as seguintes perguntas para cada ator:

- Que funções o ator requer do sistema? O que o ator precisa fazer?
- O ator precisa ler, criar, apagar, modificar, ou armazenar alguns tipos de informações no sistema?
- O ator tem que ser notificado sobre eventos no sistema? Ou o ator precisa notificar o sistema sobre alguma coisa? O que estes eventos representam em termos de funcionalidade?
- O trabalho diário do ator poderia ser simplificado ou feito com mais eficiência através de novas funções no sistema?

# ***UML – Unified Modeling Language***

## ■ Encontrando Casos de Uso

Sem considerar os atores atuais:

- Quais entradas/saídas o sistema precisa ? De onde as entradas vêm e para onde as saídas vão?
- Quais são os maiores problemas com a implementação atual do sistema?



# ***UML – Unified Modeling Language***

■ Agora vamos definir algumas ações de cada usuário:

a) **Secretária**

Consultar Agenda

Marcar Consulta

Cancelar Consulta

b) **Médico**

Realizar Consulta

Prescrever Medicação

Solicitar Realização de Exames

# ***UML – Unified Modeling Language***

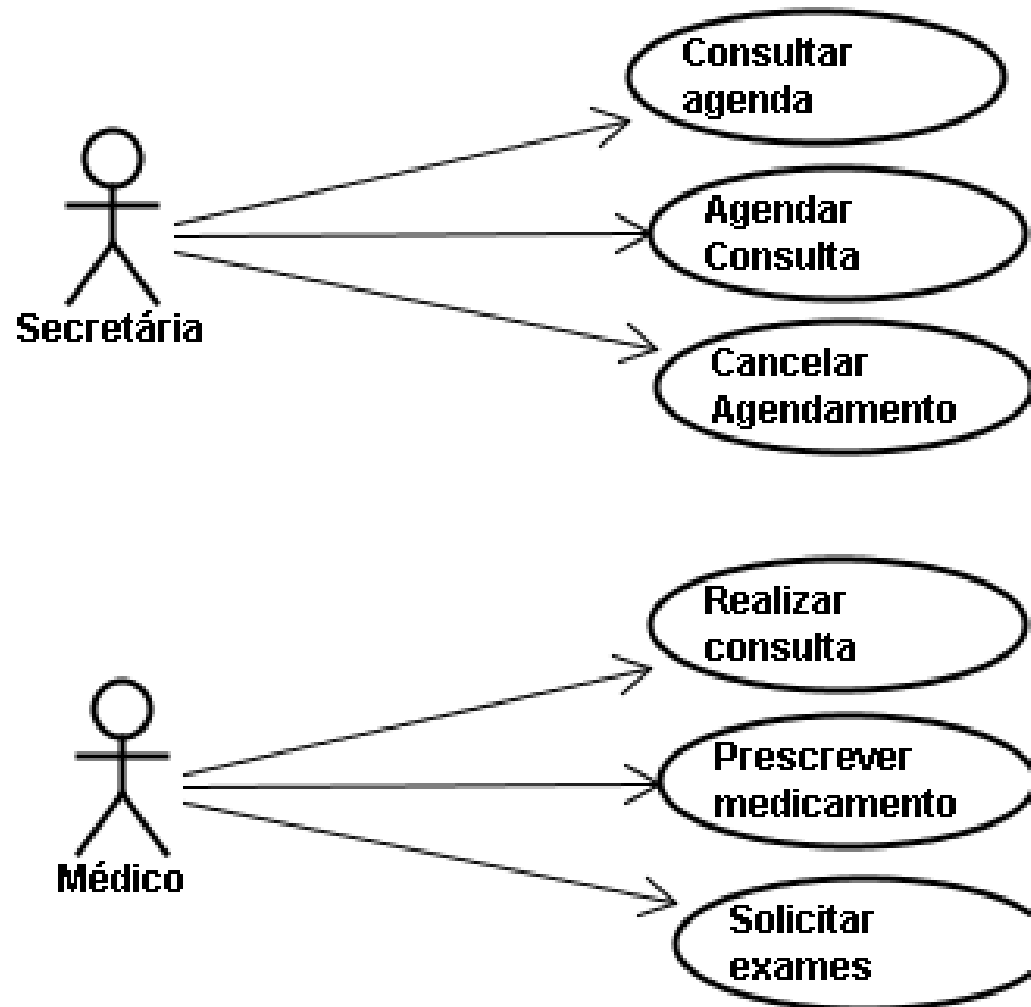
Agora já temos uma relação de atores e ações relacionadas a esses atores. Poderíamos criar um documento textual para registrar nossos atores e funcionalidades.

Mas, alguém um dia disse que “uma imagem vale mais que mil palavras”. Pois bem, **nesta fase inicial** podemos expressar tudo o que definimos em um desenho simples utilizando os padrões da UML para documentação de casos de uso. Vamos fazer um DIAGRAMA DE CASO DE USO.

# ***UML – Unified Modeling Language***

Uma possível interpretação para nosso problema.

Como podemos observar esse diagrama composto por desenhos simples descrevem de maneira bem objetiva a situação proposta. Nele vemos as funcionalidades do sistema e as interações dos usuários com elas.



# ***UML – Unified Modeling Language***

- Organizando Casos de Uso
  - Generalização / Herança
  - Inclusão
  - Extensão

# ***UML – Unified Modeling Language***

## ■ **Generalização / Herança**

- Relaciona um caso de uso especializado a um mais geral
- O filho herda os atributos, operações e sequência de comportamento dos pais
- O filho pode adicionar e redefinir o comportamento do pai
- O filho pode substituir o pai em qualquer lugar que ele aparece

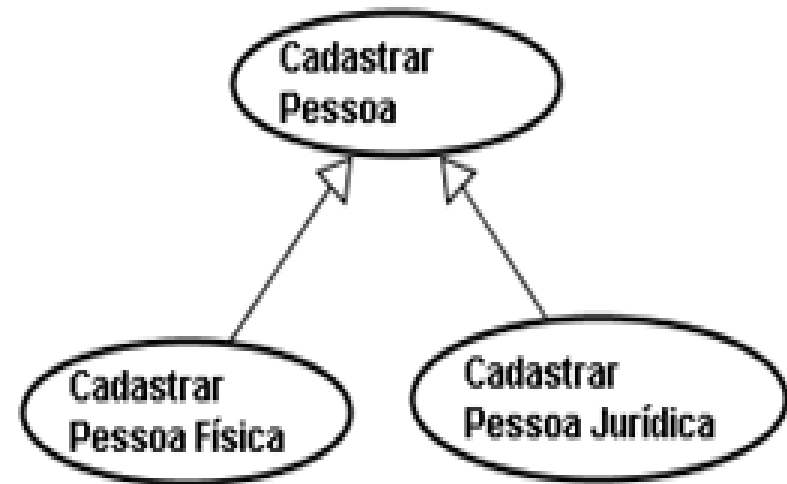
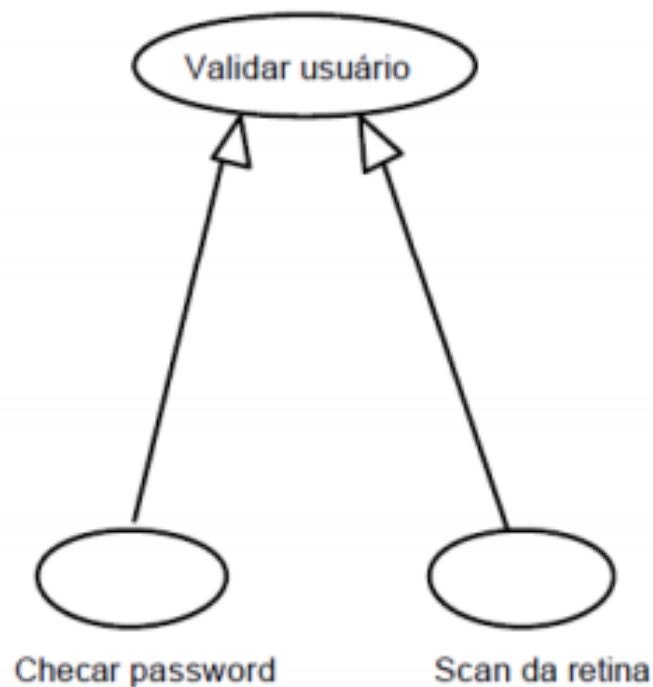
# ***UML – Unified Modeling Language***

## ■ **Generalização / Herança**

- É possível abstrair comportamentos de casos de uso.
- Normalmente a similaridade entre casos de uso é identificada após a construção do caso de uso.
- Os casos de uso Checar password e Scanear retina ambos servem para validar o usuário.
- Identificar um caso de uso abstrato Validar usuário para realizar esta validação.

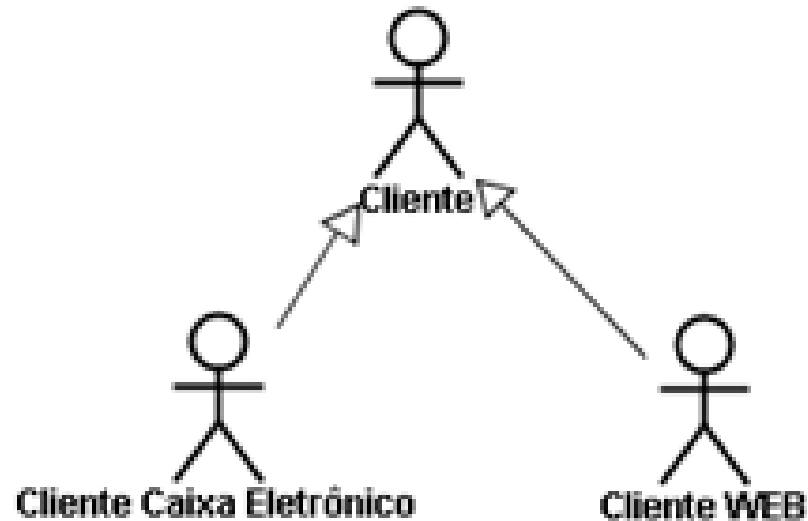
# ***UML – Unified Modeling Language***

- **Generalização / Herança – casos de uso**



# ***UML – Unified Modeling Language***

- Generalização / Herança – atores





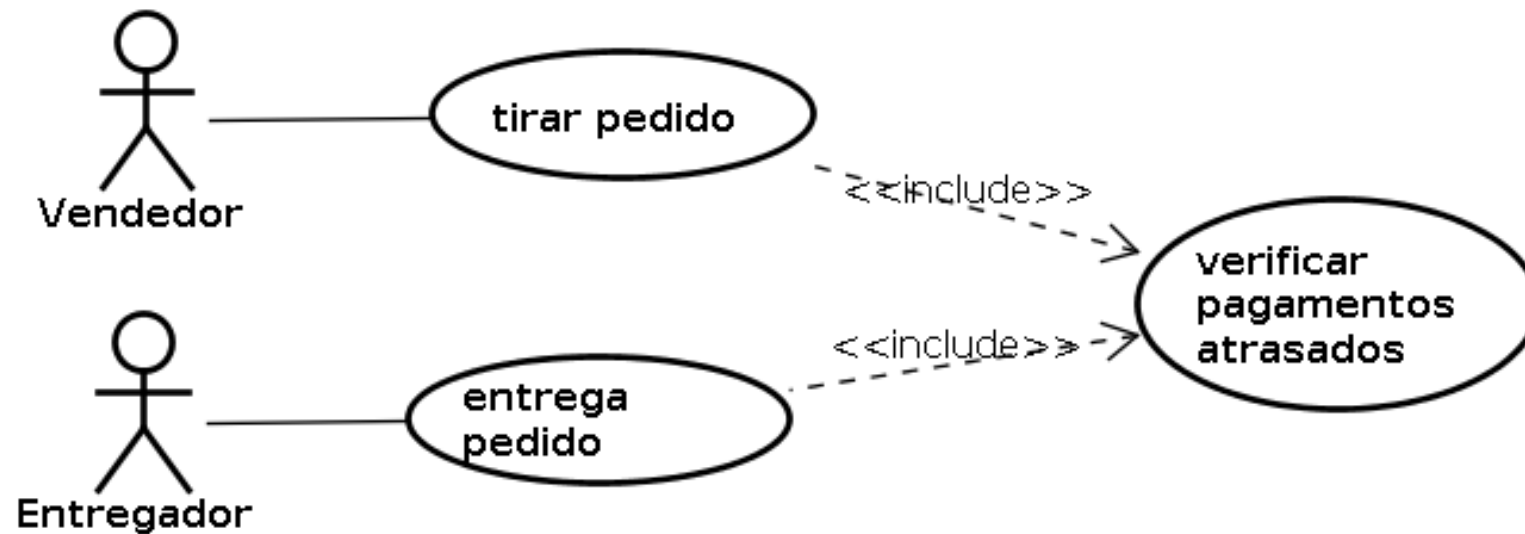
# ***UML – Unified Modeling Language***

## **Inclusão - Include**

- Em uma das formas de interação, um determinado caso de uso pode *incluir* outro. Incluir é uma relação direta entre dois casos de usos, implicando que o comportamento do caso de uso incluído é inserido no comportamento do caso de uso inclusor.
- Esta relação indica uma **obrigatoriedade** do caso de uso incluir a funcionalidade do caso de uso incluído. Assim, sempre que o primeiro ocorrer obrigatoriamente o incluído ocorrerá.
- A notação é uma seta pontilhada para o caso de uso incluído com o estereótipo <<include>>.

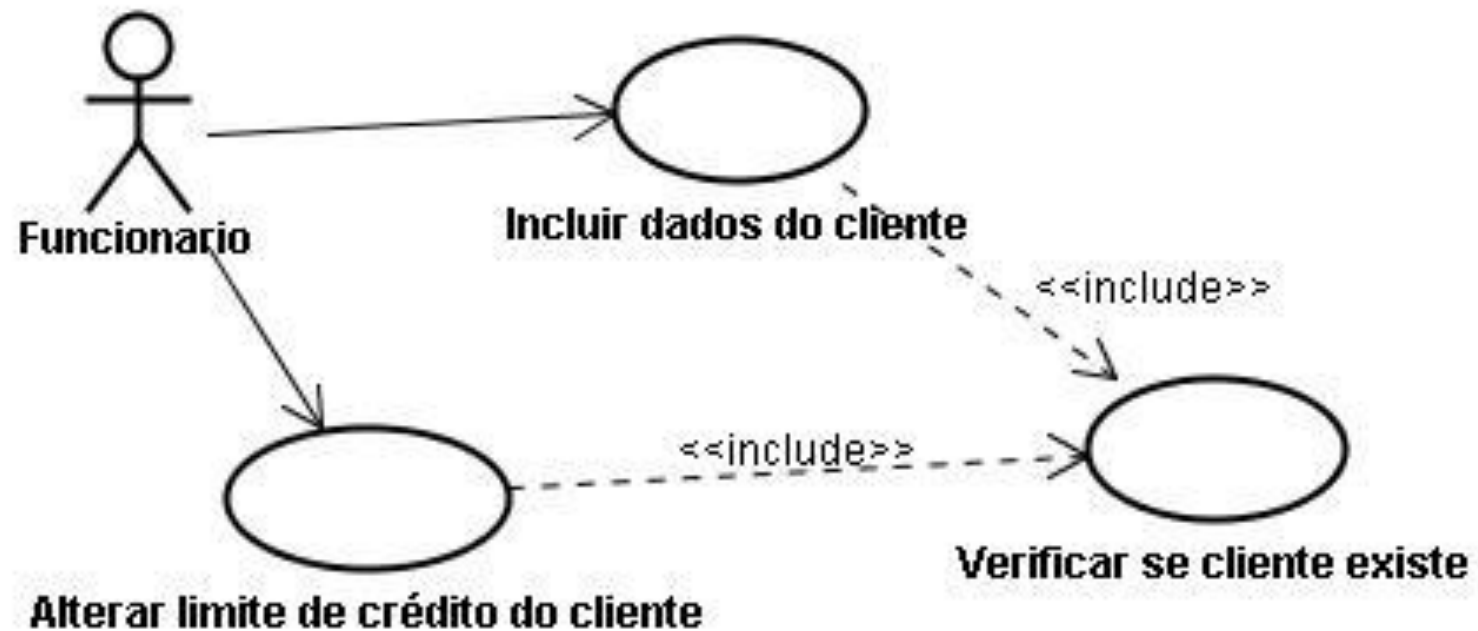
# ***UML – Unified Modeling Language***

## **Include - exemplo**



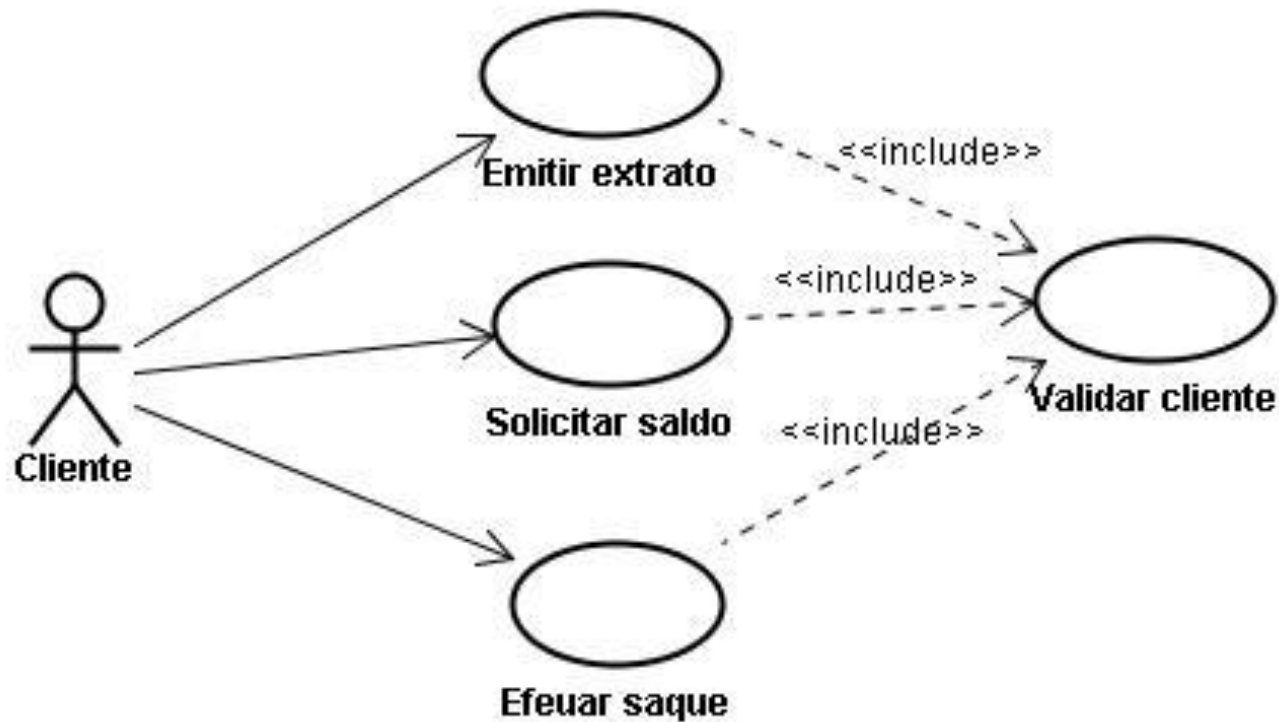
# ***UML – Unified Modeling Language***

## **Include - exemplo**



# ***UML – Unified Modeling Language***

## **Include - exemplo**



# ***UML – Unified Modeling Language***

## ■ **Extensão - Extend**

- Outra forma de interação, um caso de uso pode *estender* outro. Esta relação indica que o comportamento do caso de uso estendido **pode ser ou não** inserida no caso de uso extensor. Notas ou restrições podem ser associadas ao relacionamento para ilustrar as condições em que este comportamento será executado.
- A notação é uma seta pontilhada da extensão para o caso de uso estendido com a etiqueta <<extend>>.

# ***UML – Unified Modeling Language***

## ■ **Extend - exemplo**

