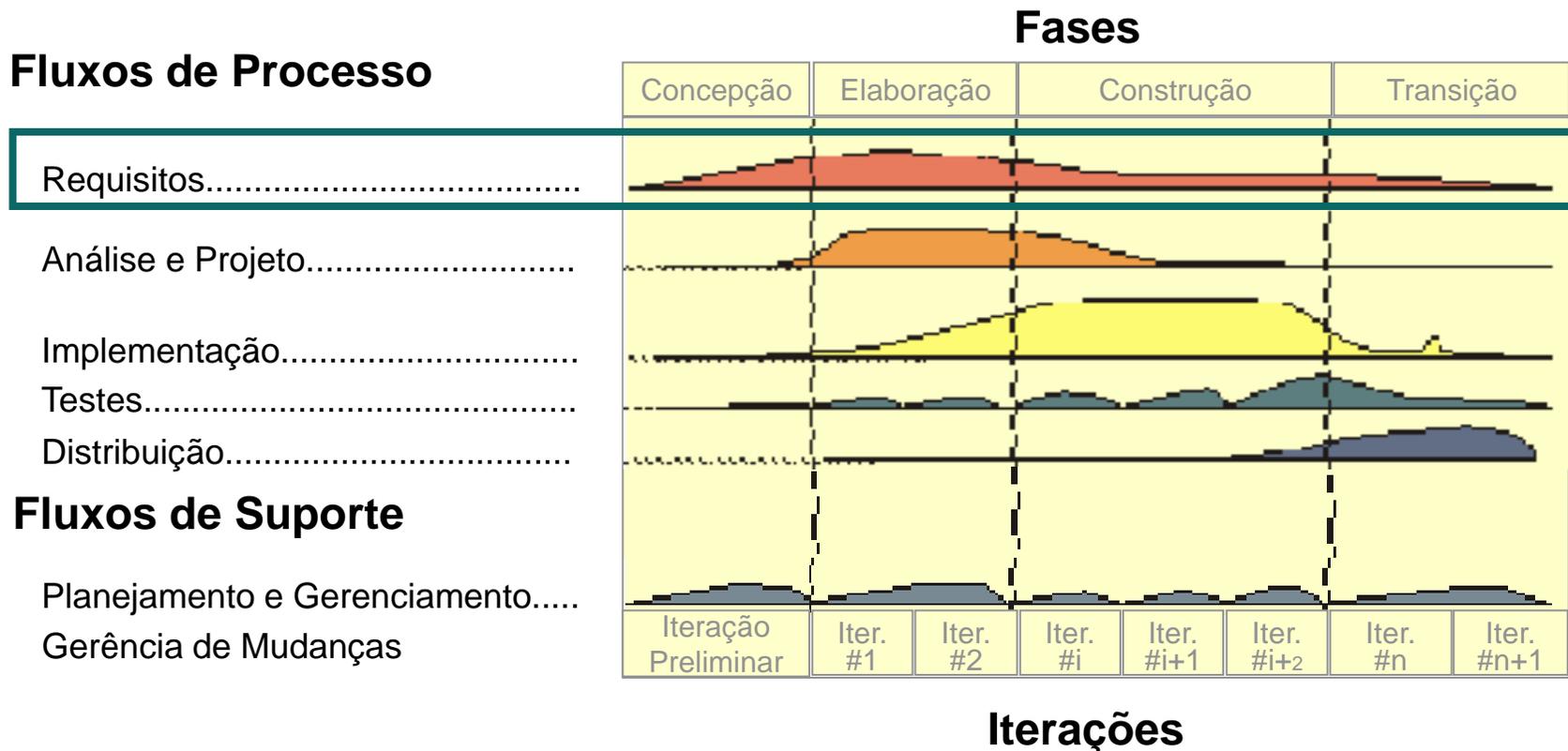


Requisitos Funcionais

Fluxo de Requisitos (RUP):
Atividades, Artefatos e
Responsáveis

O Fluxo de Requisitos



Objetivos do fluxo de requisitos

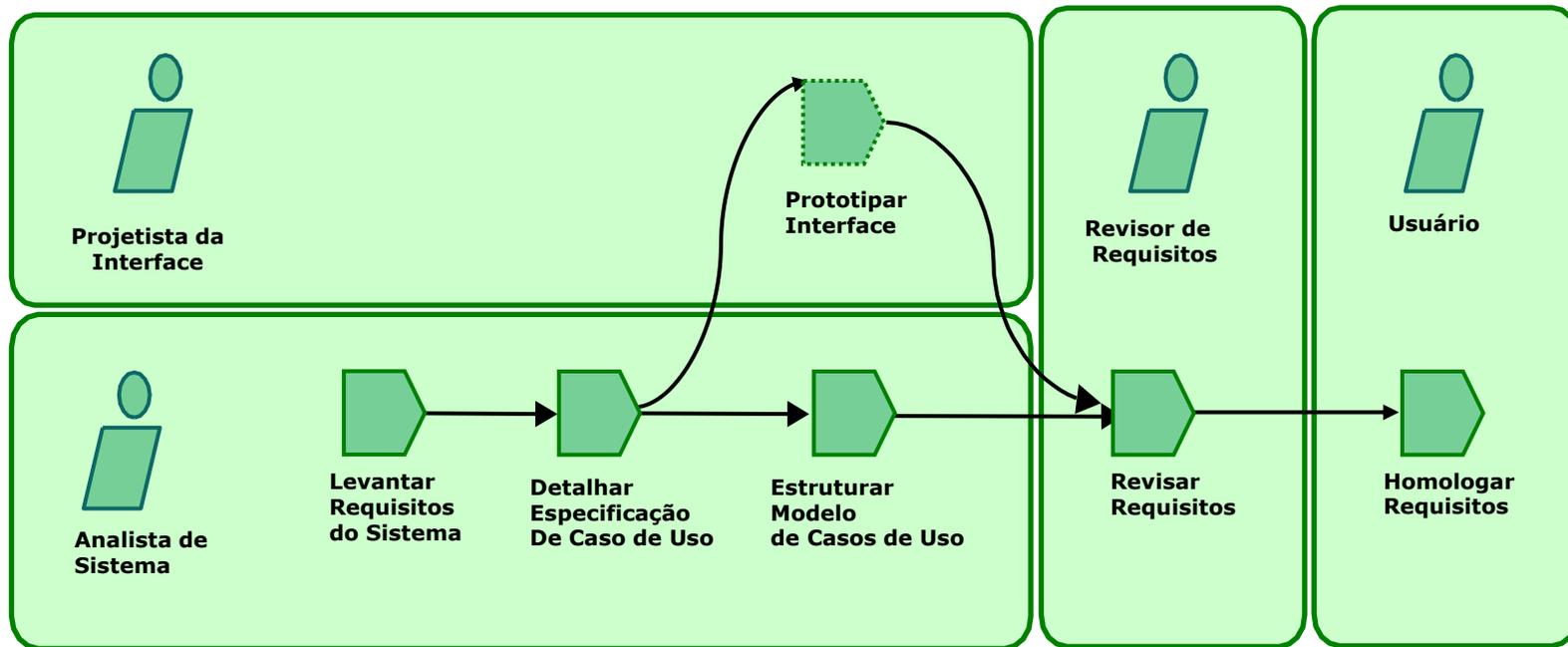
- ◆ Descrever o que o sistema deve fazer, em acordo com o cliente e usuários
- ◆ Descrever como gerenciar escopo e mudanças de requisitos
- ◆ Delimitar o sistema e prover uma base para o planejamento das iterações
- ◆ Definir a interface com o usuário

Devemos ter em mente...

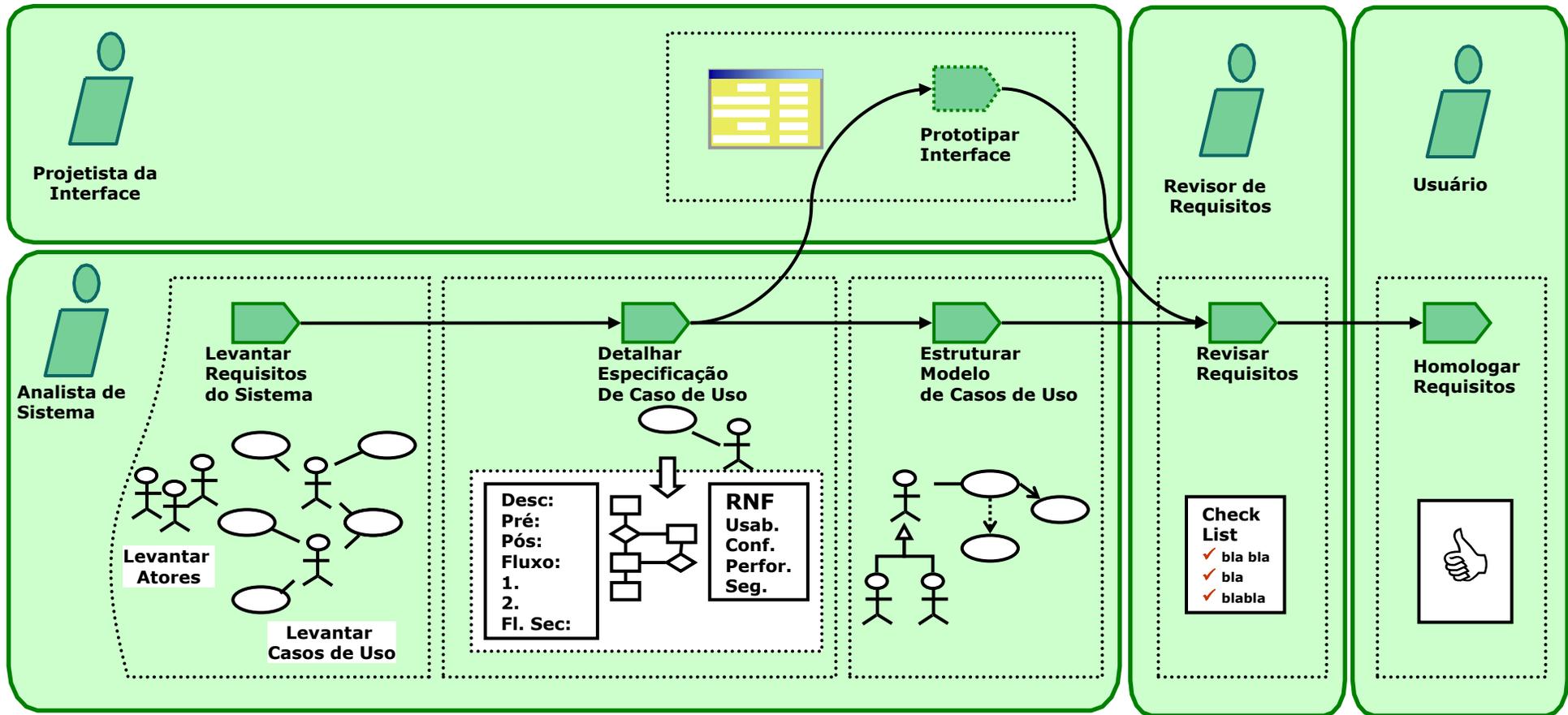
- ◆ O sistema deve prover valor ao cliente e usuários
- ◆ Os requisitos precisam ser definidos na direção correta
- ◆ Os clientes precisam entender o resultado da captura de requisitos



Atividades do fluxo de requisitos



Atividades do fluxo de requisitos



Artefatos gerados

- ◆ Glossário
- ◆ Documento de Requisitos
- ◆ Diagrama de Casos de Uso
- ◆ Protótipo da interface com o usuário (opcional)
- ◆ Termo de Homologação de Requisitos



Glossário

- ◆ Define termos importantes usados no projeto
- ◆ É importante para garantir que os conceitos envolvidos são interpretados da mesma forma por todos os membros da equipe



Glossário

Glossário: estrutura

- ◆ Introdução
 - Objetivos do documento
 - Público ao qual se destina
- ◆ Definições
 - Termos, definições e sinônimos
- ◆ Referências

Documento de requisitos: estrutura

- ◆ Introdução
 - Objetivos do documento
 - Público ao qual se destina
 - Termos e acrônimos
 - Referências
- ◆ Descrição geral do sistema
 - Abrangência e sistemas relacionados
 - Descrição dos usuários
- ◆ Casos de uso
- ◆ Requisitos não funcionais
- ◆ Diagrama de casos de uso

Documento de requisitos

◆ Casos de uso

- Identificador do caso de uso
- Breve descrição
- Ator (pode não ser inserido)
- Prioridade
- Requisitos não funcionais associados
- Pré condições
- Pós condições
- Fluxo de eventos principal
- Fluxos secundários: alternativos e de exceção
- Interfaces associadas (opcional)

Homologação de requisitos: estrutura

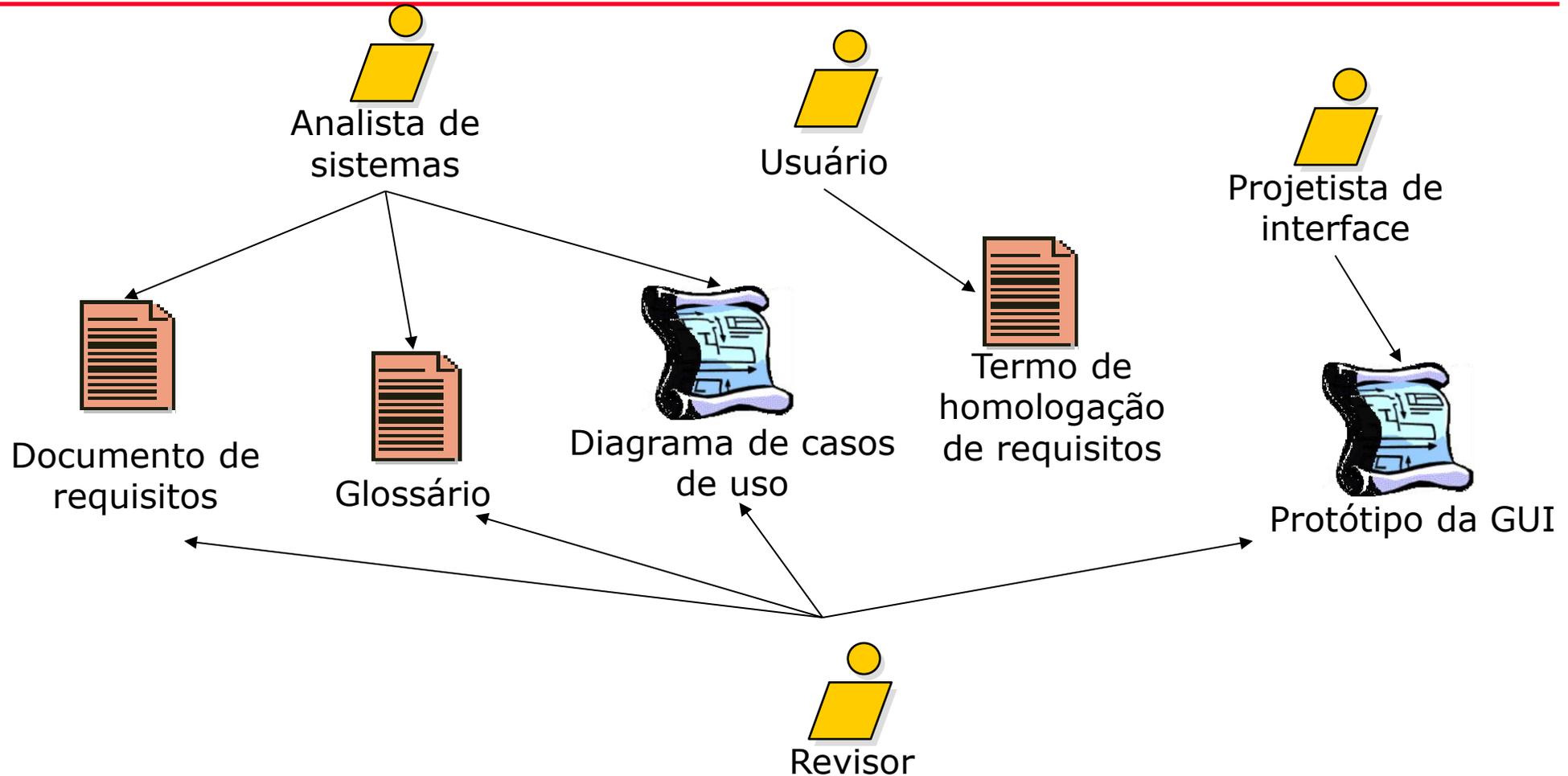
◆ Introdução

- Objetivos do documento
- Organização do documento

◆ Casos de uso homologados

- Para cada caso de uso
 - » Identificador
 - » Resultado da homologação
 - ◆ Homologado, não homologado, homologado com restrições
 - » Comentários

Responsáveis no fluxo de requisitos



Caso de Uso

Conceitos Básicos

Caso de uso

- ◆ É uma forma específica de uso do sistema através da execução de alguma de suas funcionalidades.
- ◆ Uma unidade coerente de funcionalidade provida por um sistema, manifestada por uma seqüência de mensagens trocadas entre o sistema e um ou mais usuários externos (representados como atores), junto com as ações executadas pelo sistema.

Caso de uso: continuação

- ◆ Descrevem o que acontece dentro do sistema.
- ◆ Ajudam muito na comunicação entre clientes e desenvolvedores.
- ◆ Mostram apenas o que o sistema faz, e não como.
 - Capturam o comportamento pretendido para um sistema, sem a necessidade de especificar como esse comportamento será implementado.

Caso de uso: representação gráfica

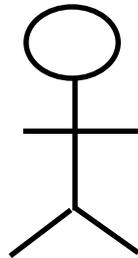
**Solicitar
extrato**

Solicitar saldo

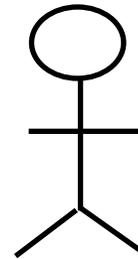
Atores

- ◆ Constituem as entidades que interagem com o ambiente do sistema
 - Pessoas ou outros sistemas (de hardware ou software) que interagem com o sistema em desenvolvimento
- ◆ Definem um papel particular
- ◆ São sempre externos ao sistema
- ◆ O sistema será descrito através de vários casos de uso que são executados por um número de atores

Ator: representação gráfica



Cliente



Caixa

Atores x usuários do sistema

- ◆ Uma mesma pessoa pode desempenhar diferentes papéis



Carlos

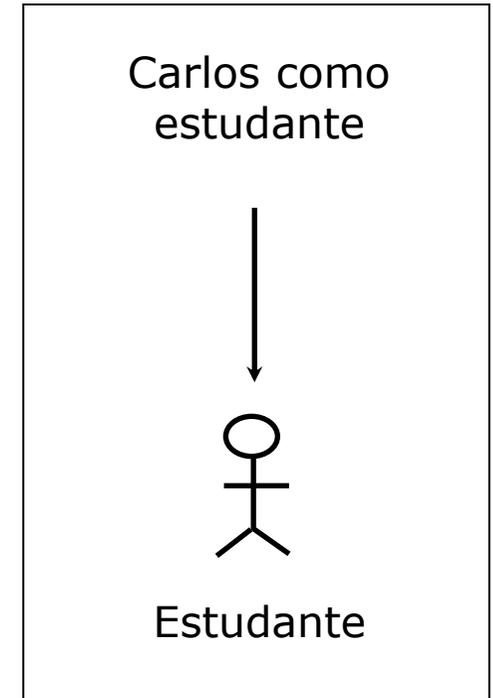
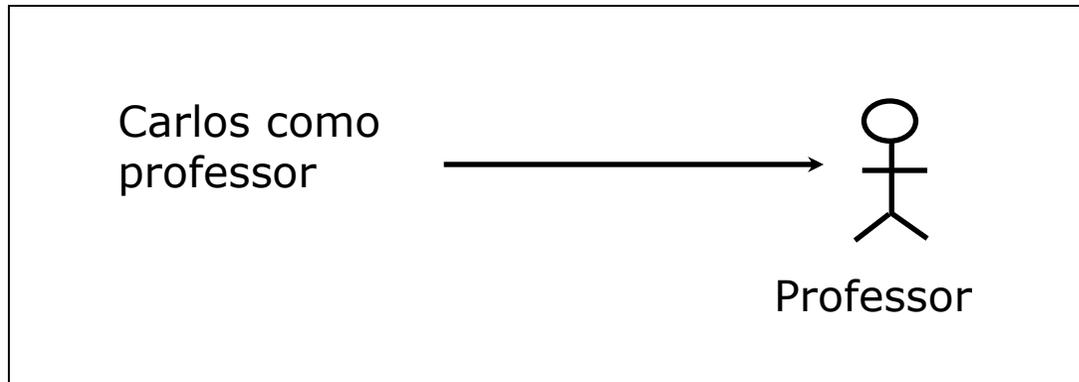


Diagrama de casos de uso

- ◆ Diagrama com os casos de uso do sistema e atores relacionados;
- ◆ Facilitam o entendimento do sistema mostrando a sua “visão externa”;
- ◆ A coleção de casos de uso deve especificar todas as formas existentes de uso do sistema.

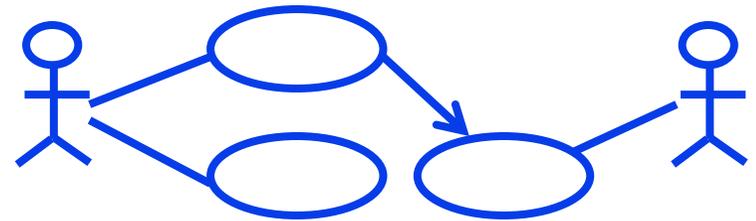
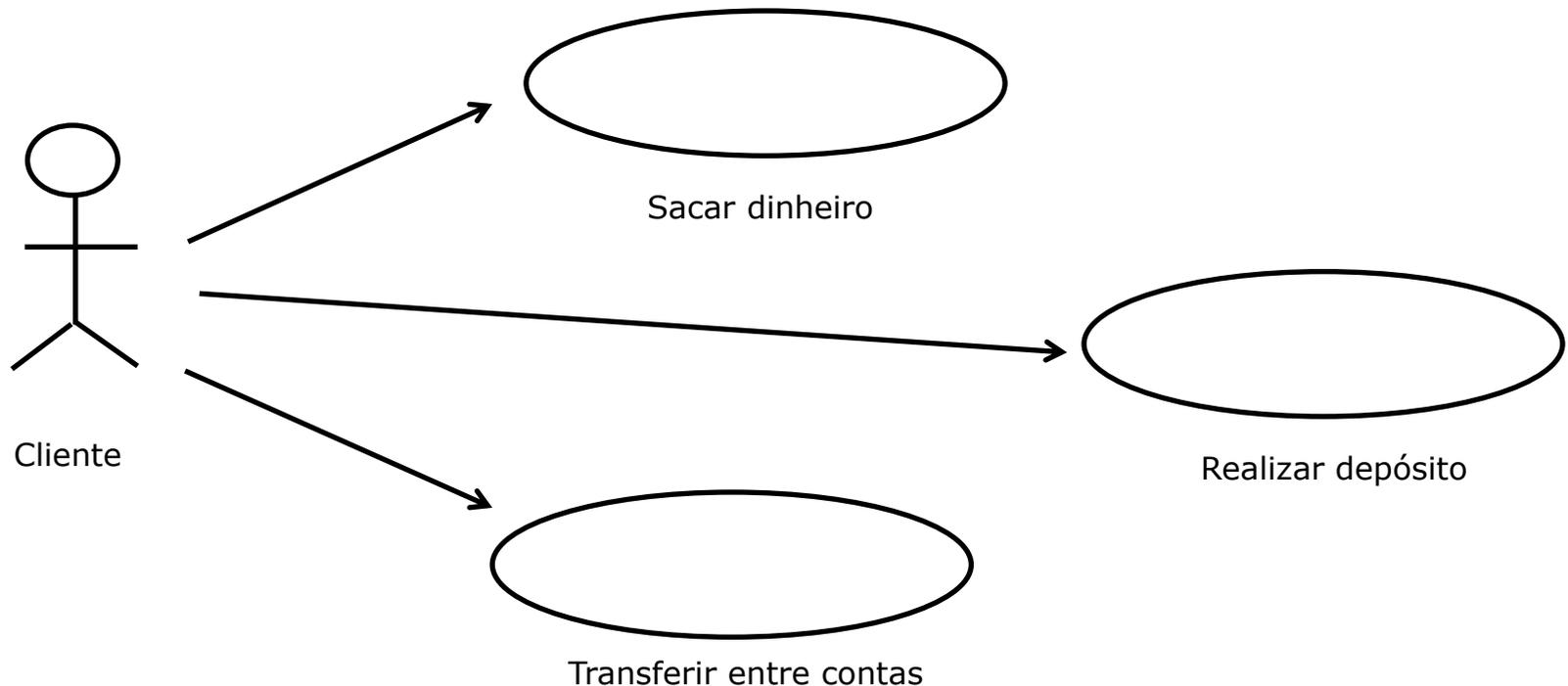


Diagrama de casos de uso: representação gráfica



Uma associação entre um *ator* e um *caso de uso* indica que há uma comunicação, possivelmente com envio e recepção de mensagens.

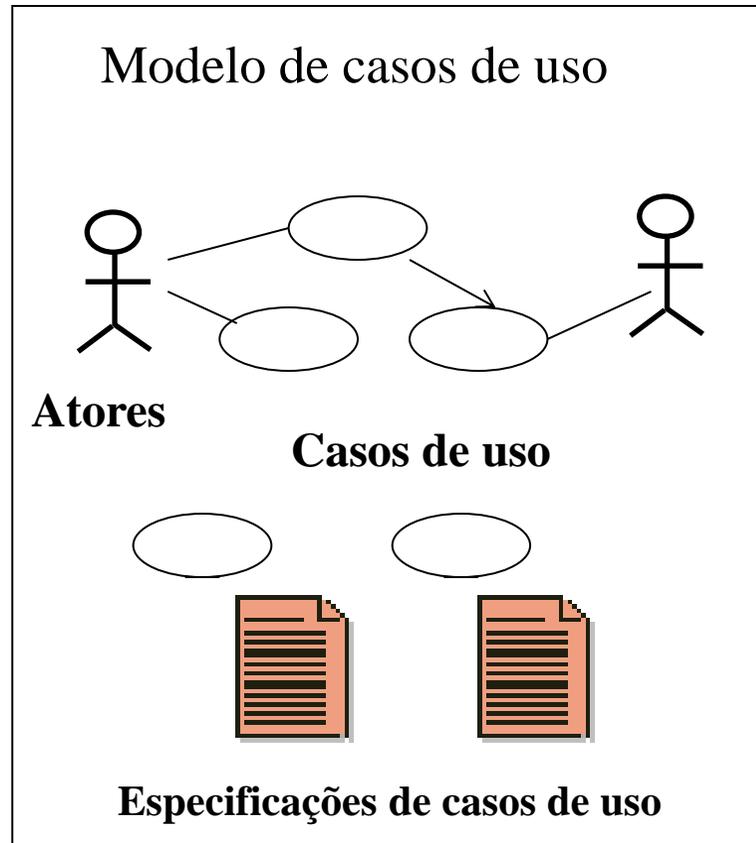
Requisitos x casos de uso

- ◆ Um requisito funcional pode ser mapeado em um ou mais casos de uso
- ◆ Requisitos não funcionais podem ser:
 - Específicos: associados a um caso de uso específico
 - Genéricos: associados a vários casos de uso ou ao sistema com um todo
 - Para serem atendidos podem gerar novos casos de uso

Especificação de Caso de Uso

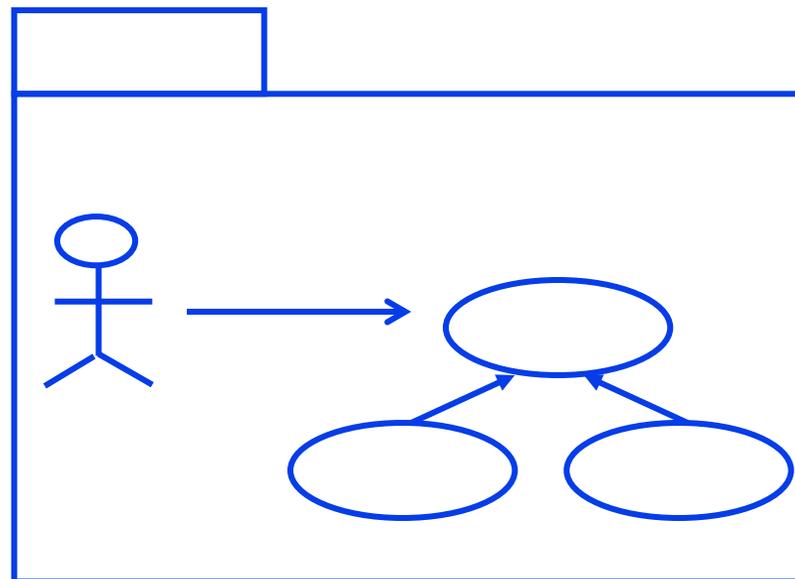
- ◆ Identificador do caso de uso
- ◆ Breve Descrição
- ◆ Ator (opcional)
- ◆ Prioridade
- ◆ Pré condições
- ◆ Pós condições
- ◆ Fluxos de eventos:
 - Fluxo de eventos principal
 - Fluxos secundários: alternativos e de exceção
- ◆ Requisitos Não-Funcionais Específicos
- ◆ Interface gráfica associada

Modelo de casos de uso



Pacotes de Casos de Uso

- ◆ Servem para agrupar casos de uso relacionados
- ◆ Critérios para agrupamento:
 - ator
 - funcionalidades correlatas
 - etc

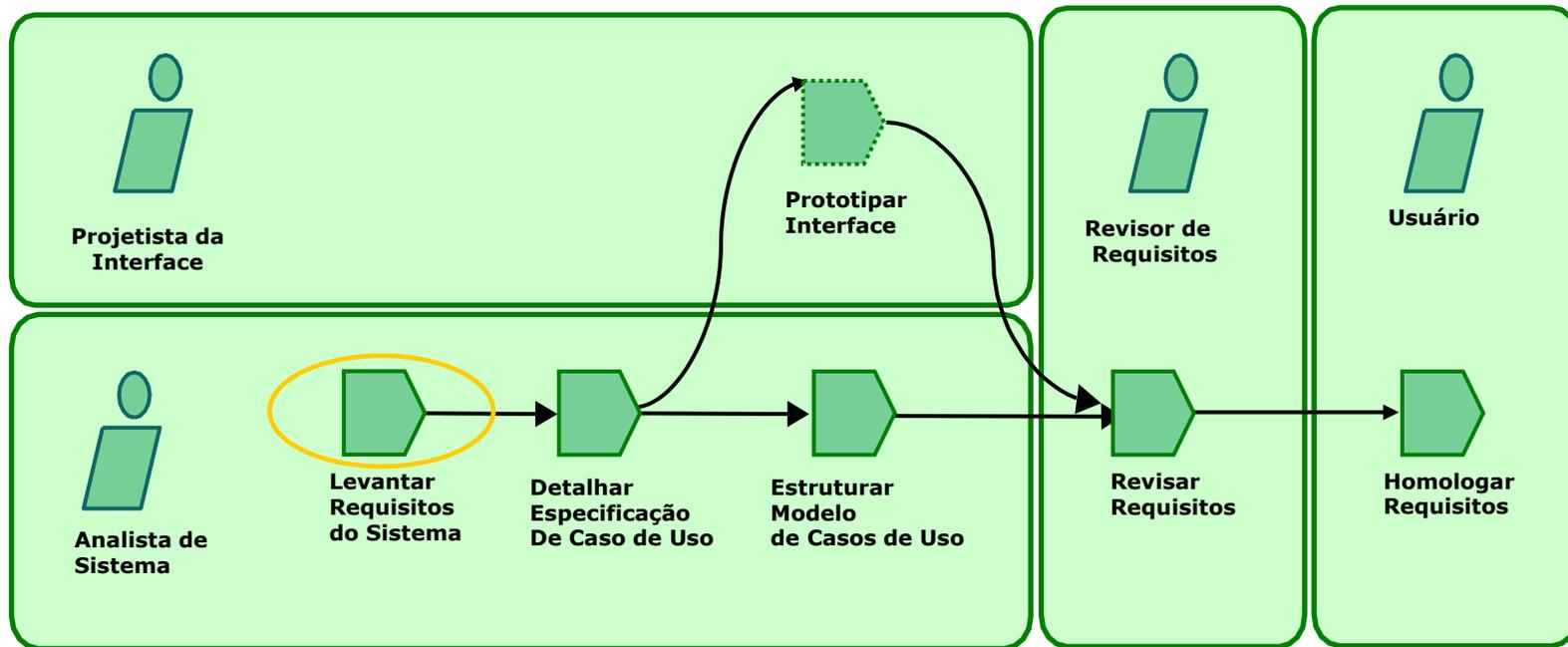


Levantamento de Casos de Uso

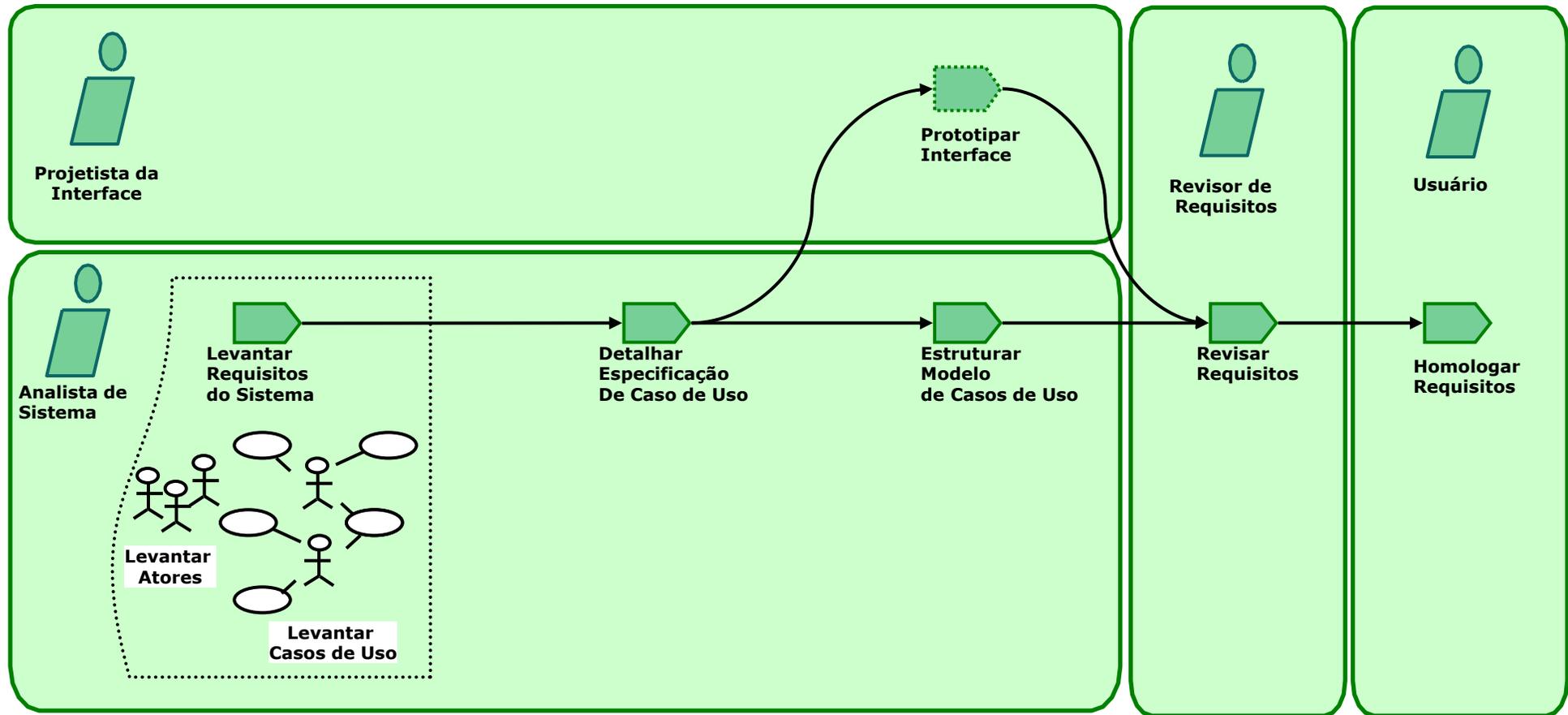
Objetivos

- ◆ Discutir como encontrar atores e casos de uso
- ◆ Apresentar o diagrama de casos de uso e o diagrama de atividades de UML
- ◆ Discutir como especificar os casos de uso

Levantar Requisitos do Sistema



Levantar Requisitos do Sistema



Como encontrar atores?

- ◆ Quem usa o sistema?
- ◆ Quem instala/mantém o sistema?
- ◆ Quem inicia/desliga o sistema?
- ◆ Que outros sistemas usam o sistema?
- ◆ Quem recebe informação do sistema?
- ◆ Quem provê informação ao sistema?

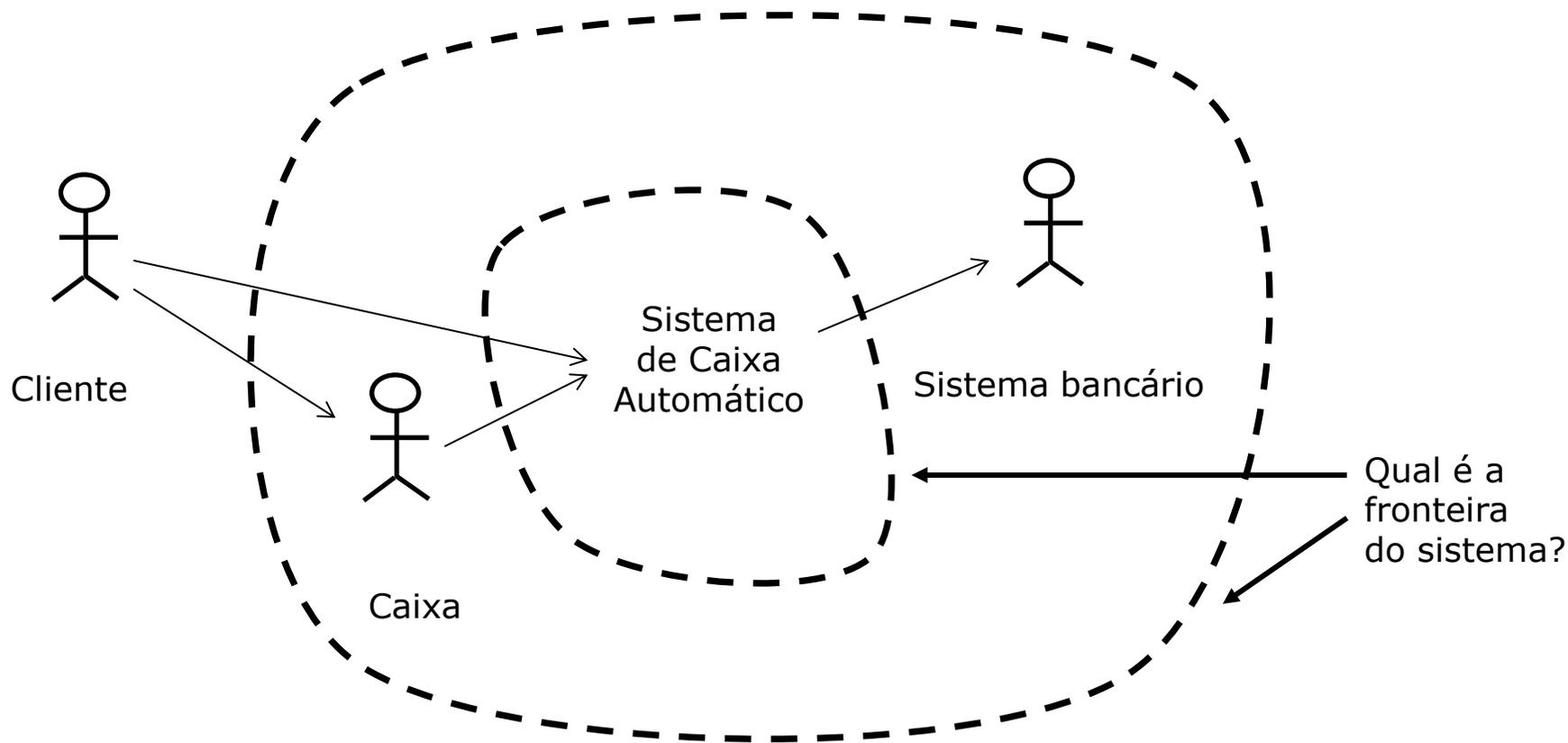


Como encontrar casos de uso?

- ◆ Atores são fundamentais para a descoberta dos casos de uso
- ◆ Pergunte:
 - Que funções o ator vai querer do sistema?
 - O sistema armazena informações? Que informações atores irão criar, ler, atualizar ou apagar?
 - O sistema precisa notificar o ator sobre mudanças no seu estado interno?
 - Existe algum evento externo que o sistema precisa saber? Que ator informa o sistema desses eventos?

Escopo do sistema

- ◆ É preciso delimitar as fronteiras do sistema



Técnicas para levantar casos de uso

◆ Use-Case Workshop

- não pode ter muita gente
- pessoas com diferentes perfis
- presença de um facilitador
- aceitar todo tipo de sugestão, filtrar depois!
- evite pensar em detalhes
- os casos de uso levantados devem estar claros para todos!
 - » Principalmente o valor que este agrega ao usuário
- consulte todos!
- dê sugestões



Técnicas para levantar casos de uso

◆ Reuniões

- conversas com usuários

◆ Storyboarding

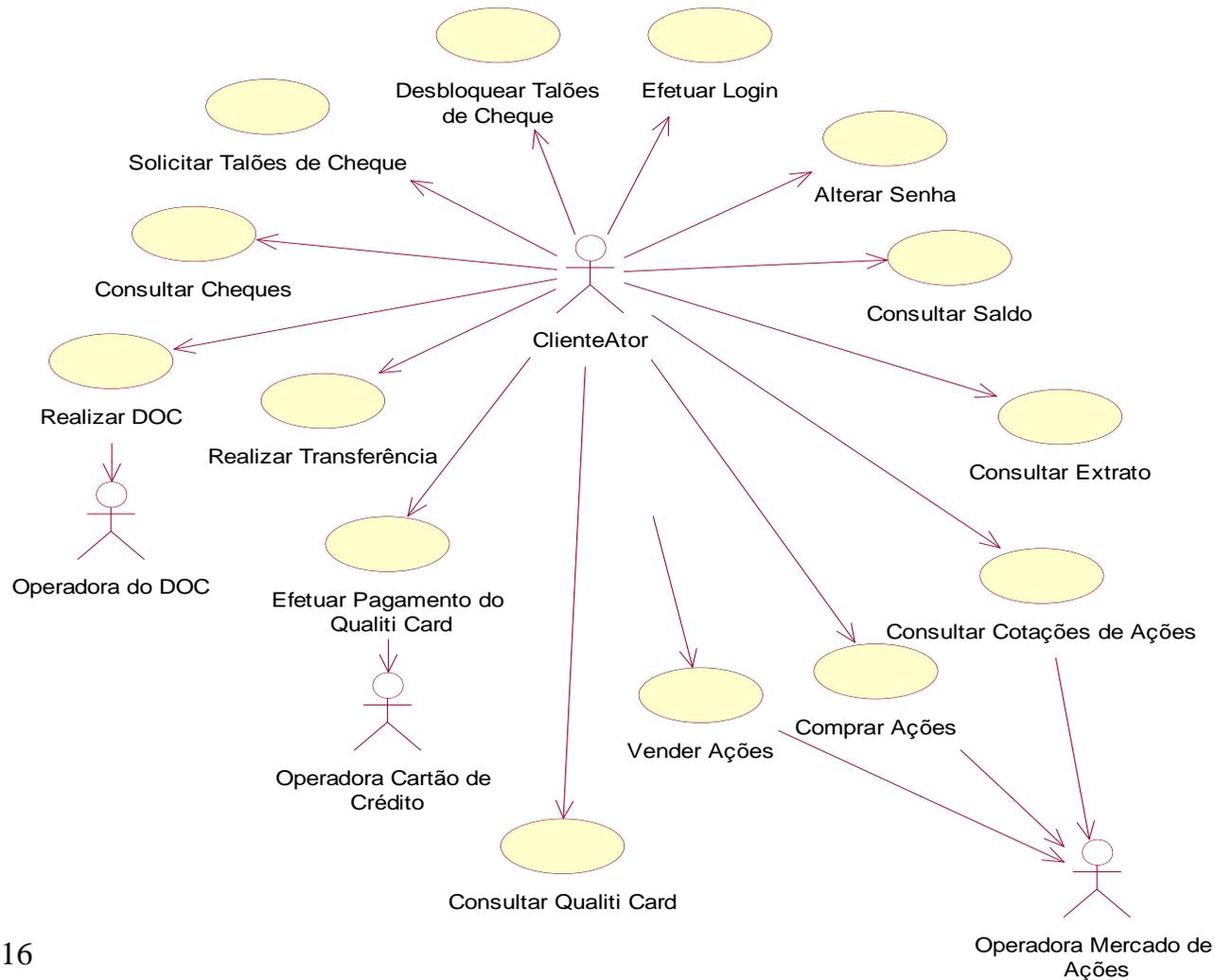
- simulação através de desenhos das interfaces

Exercício

- ◆ Dada uma descrição preliminar do QIB, observe seu diagrama de casos de uso no próximo slide.
- ◆ Em seguida, com base na descrição inicial do Amazônia, crie um diagrama de casos de uso.

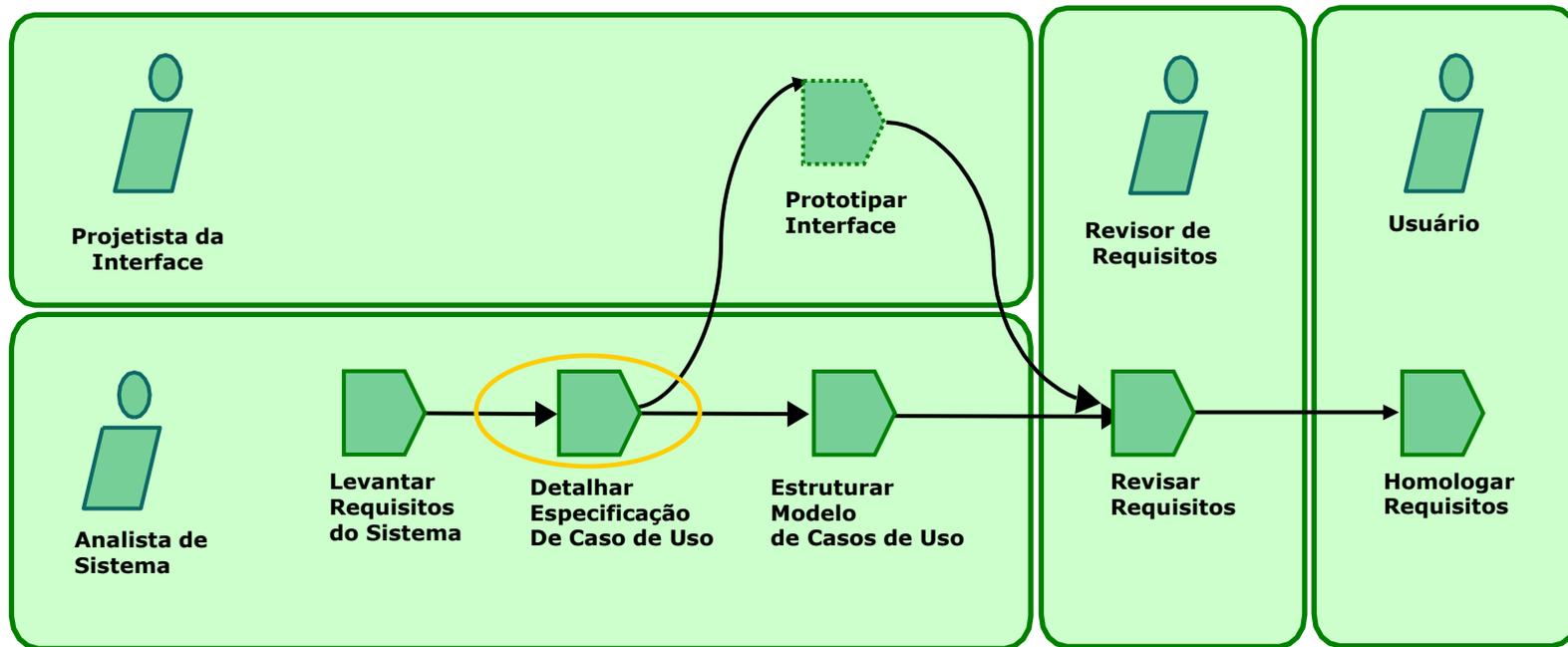


Diagrama de casos de uso do QIB

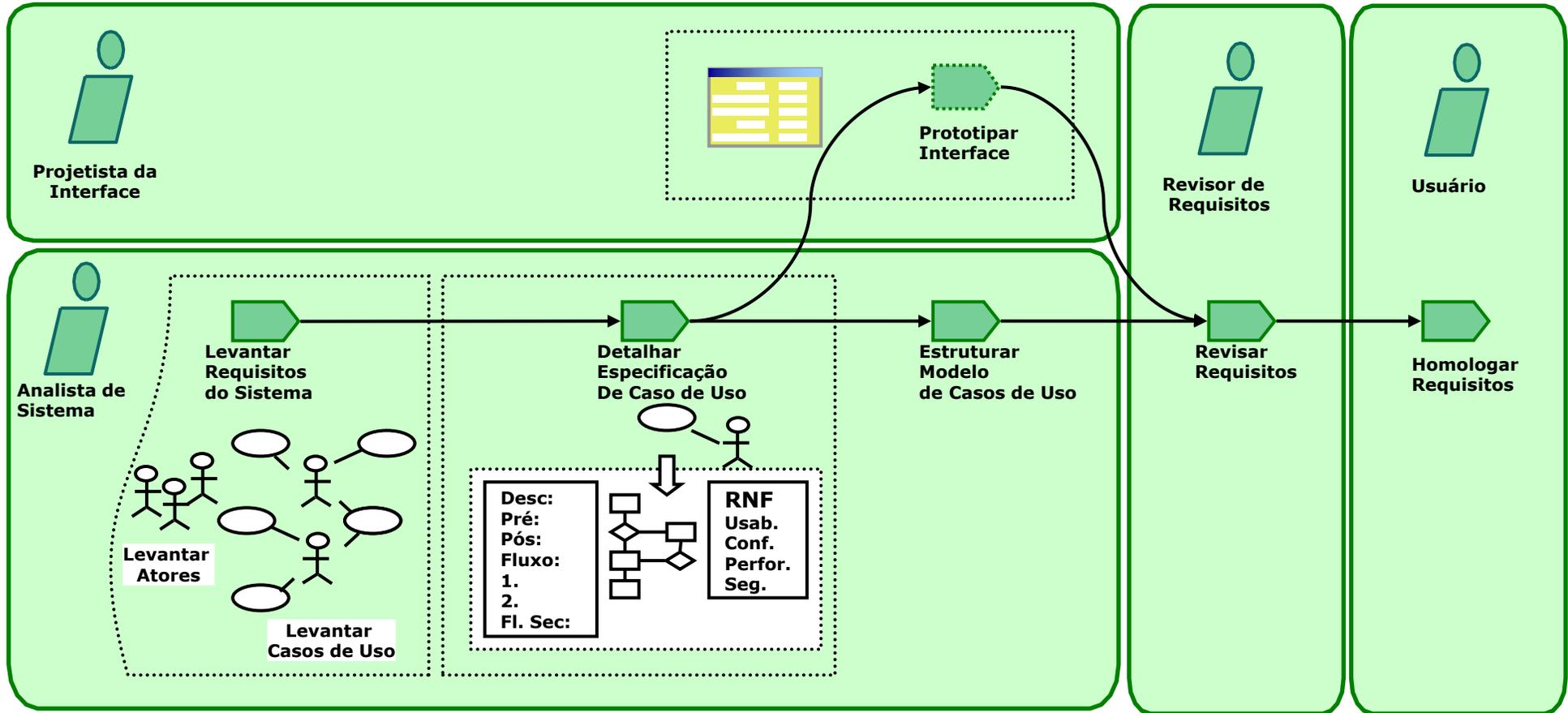


Especificação Detalhada dos Casos de Uso

Detalhar Especificação de Caso de Uso



Detalhar Especificação de Caso de Uso



Quando e por que realizá-las?

◆ Quando?

- após fazer levantamento dos principais casos de uso do sistema

◆ Por que?

- descrever detalhes dos casos de uso
- descrever fluxos de eventos e outras propriedades
- uniformizar entendimento entre clientes, usuários e equipe de desenvolvimento

Especificando casos de uso

- ◆ Casos de uso não precisam ser especificados todos de uma vez
 - o processo é iterativo!
- ◆ Casos de uso devem ser priorizados por iteração
 - Prioridade técnica
 - Prioridade do usuário

Especificando um caso de uso

- ◆ Identificador
- ◆ Breve descrição
- ◆ Ator
- ◆ Prioridade
- ◆ Requisitos não-funcionais associados
- ◆ Pré-condições
- ◆ Pós-condições
- ◆ Fluxo de eventos principal
- ◆ Fluxos secundários: alternativos e de exceção
- ◆ Interfaces associadas (opcional)

Identificação do caso de uso

- ◆ Deve ser única!
- ◆ Não deve mudar nunca
- ◆ Pois casos de uso podem ser referenciados por seu identificador

Breve descrição do caso de uso

- ◆ Dar uma idéia do propósito do caso de uso, do seu objetivo
- ◆ Deve ser feita ao se identificar o caso de uso, para evitar mal-entendidos
- ◆ 2 ou 3 linhas!

Prioridade de casos de uso

- ◆ Essencial para gerenciar os requisitos
- ◆ É preciso definir prioridade de todos os casos de uso
- ◆ Exemplos de prioridade:
 - Essencial
 - Importante
 - Desejável

Pré e pós condições

- ◆ O que deve ser verdade antes e depois da realização do caso de uso!

Pré e pós condições: exemplos

- ◆ Caso de uso “Entregar pedido”
 - Pré condição: os itens do pedido devem existir em estoque
 - Pós condição: os itens enviados devem ser abatidos do estoque

- ◆ Caso de uso “Recadastrar CPF”
 - Pré condição: o usuário deve possuir um CPF
 - Pós condição: a situação do contribuinte é atualizada

Fluxo de eventos básico/principal

- ◆ Série de passos que compõem um caso de uso
- ◆ Concentre-se inicialmente na funcionalidade básica/central do caso de uso
- ◆ Pense nos fluxos secundários depois!

Exemplo de um fluxo básico

◆ Caso de uso “Sacar dinheiro”

1. O cliente passa o seu cartão
2. Digita sua senha
3. Digita o valor do saque
4. O sistema verifica se há saldo suficiente
5. O saldo é debitado da conta do cliente
6. O dinheiro é entregue ao cliente

Exemplo de um fluxo básico

- ◆ Caso de uso “Sacar dinheiro”
- ◆ MAS...
 - E se a senha não conferir?
 - E se não houver saldo?
 - E se não houver dinheiro suficiente na máquina?

Calma, vamos deixar esses detalhes para depois!

Subfluxos

- ◆ Às vezes, o fluxo principal possui várias alternativas igualmente prováveis de ocorrer
- ◆ Nestes casos, pode-se usar o conceito de subfluxos!
- ◆ Cada subfluxo representa um dos possíveis caminhos do fluxo principal

Subfluxos - Exemplo

- ◆ Considere um sistema que realiza compra e vendas de produtos.
- ◆ Caso de uso Cadastrar Produtos
 - Fluxo básico
 1. O funcionário seleciona a opção de cadastro, iniciando o caso de uso.
 2. O sistema requisita que o funcionário forneça a operação que quer efetuar: inclusão, atualização ou remoção de produtos.
 3. De acordo com a opção fornecida pelo funcionário, um dos subfluxos abaixo é executado.

Subfluxos - Exemplo

- Subfluxo Incluir produto
 - 1.O sistema requisita o nome, descrição e preço do novo produto.
 - 2.Quando o usuário fornece os dados requisitados, o sistema gera um identificador único para o novo produto e o armazena no cadastro de produtos.

Subfluxos - Exemplo

- Subfluxo Alterar informações do produto
 - 1.O sistema requisita o nome ou identificador do produto a ser alterado.
 - 2.O funcionário fornece o identificador do produto.
 - 3.O sistema recupera e apresenta os dados do produto para alteração (os mesmos dados requisitados no subfluxo “Incluir produto”).
 - 4.O funcionário atualiza os dados do produto e o sistema armazena os novos dados.
- Subfluxo Remover produto
 - ...

Fluxos secundários

- ◆ Só devem ser analisados e descritos após a descrição dos fluxos básicos.
- ◆ Fluxos alternativos
 - situações especiais (desconto para um cliente)
- ◆ Fluxos de erro
 - situações de erro

Reuso de fluxos secundários

- ◆ Fluxos secundários, principalmente de erros, podem ser referenciados por diferentes casos de uso
- ◆ Evitar duplicação de informação!

Descrição da interface com o usuário

Interfaces associadas (opcional)

- ◆ Ferramenta para compreensão do caso de uso
 - nível de detalhes adequado
- ◆ Facilidade para a descrição de críticas básicas
 - tamanho e tipo dos campos
 - máscaras

Exemplo – QIB

- ◆ Observe os fluxos secundários (alternativos e de exceção) dos casos de uso Atualizar Cotações e Comprar Ações.



Exercício

- ◆ Observe a especificação detalhada dos casos de uso Atualizar Cotações e Comprar Ações do QIB.
- ◆ Em seguida, com base na descrição preliminar e no diagrama de casos de uso do Amazônia, especifique o fluxo principal de dois casos de uso.

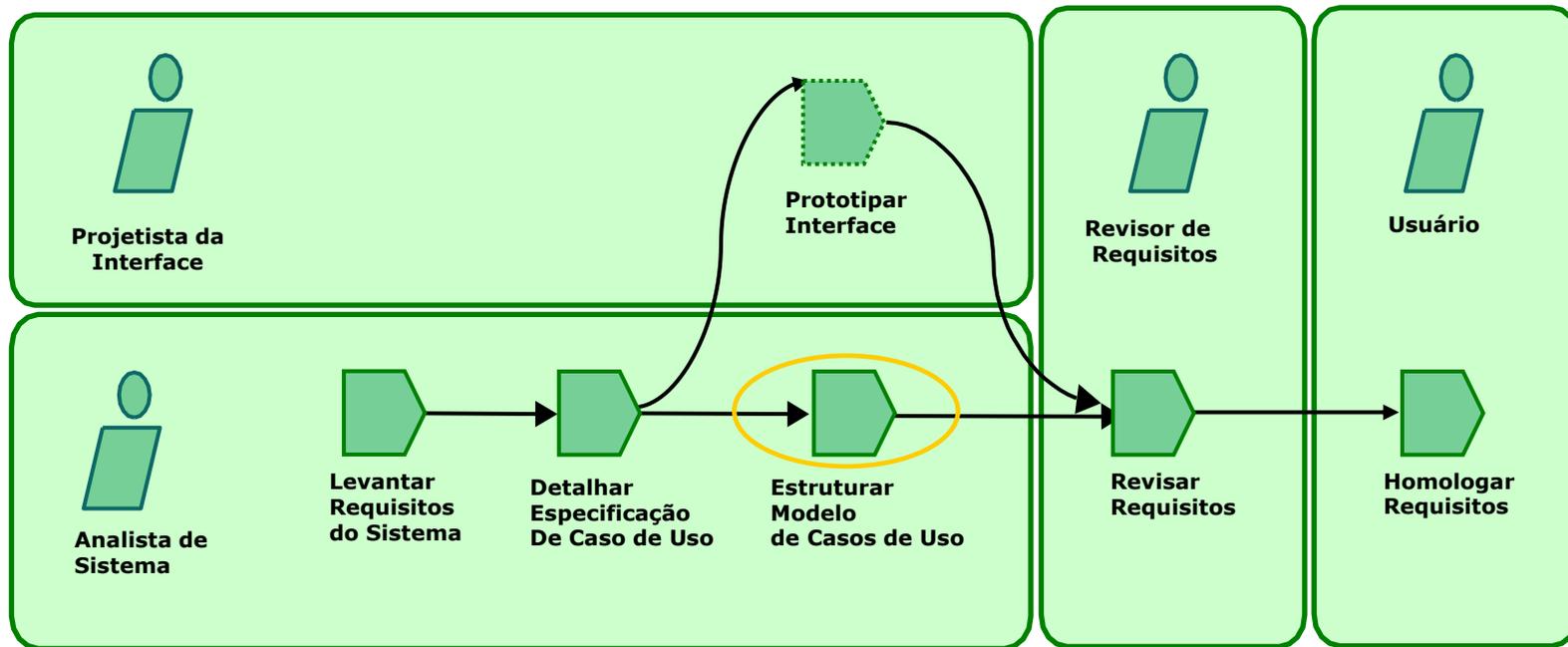


Estruturação do Modelo de Casos de Uso

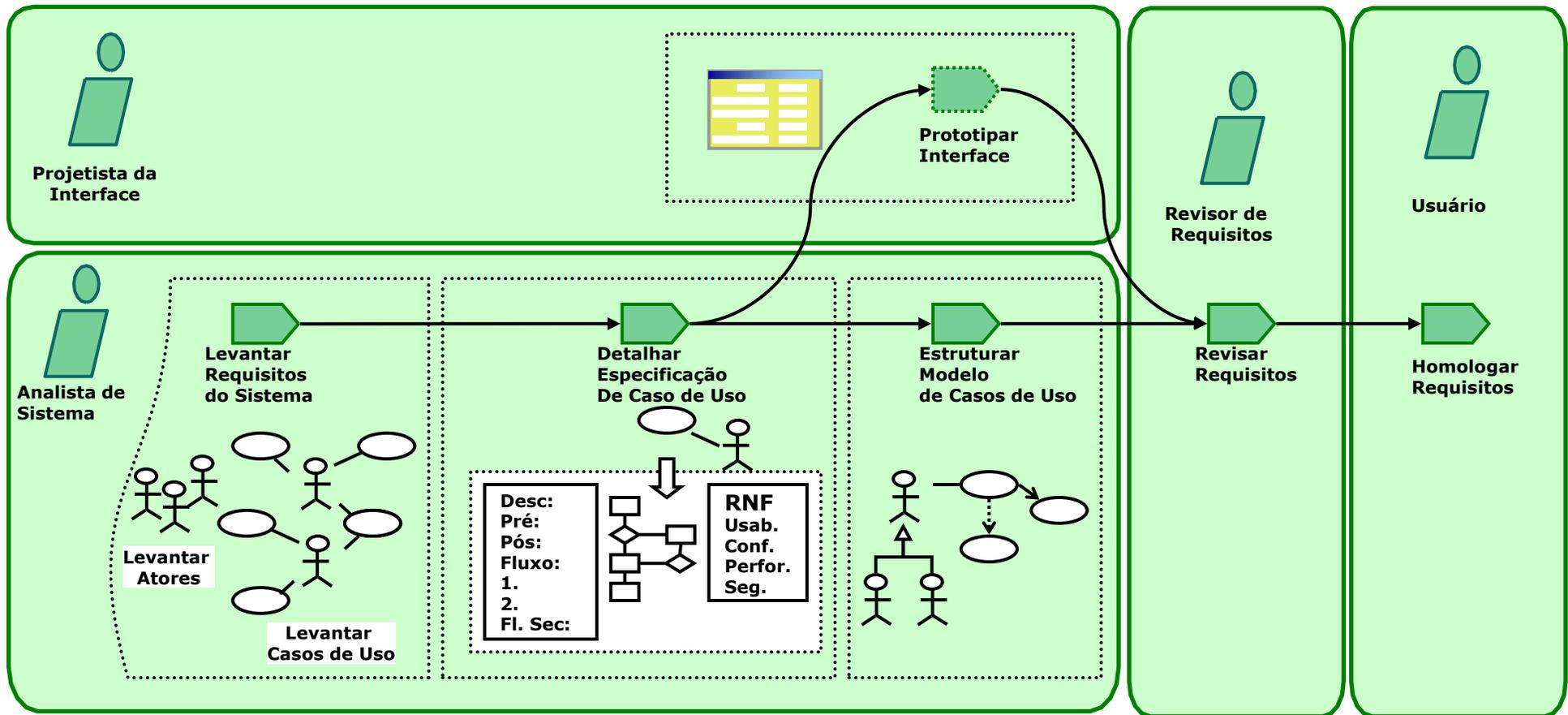
Objetivos

- ◆ Apresentar os conceitos necessários e elementos de UML usados para estruturar o modelo de casos de uso

Estruturar Modelo de Casos de Uso



Estruturar Modelo de Casos de Uso



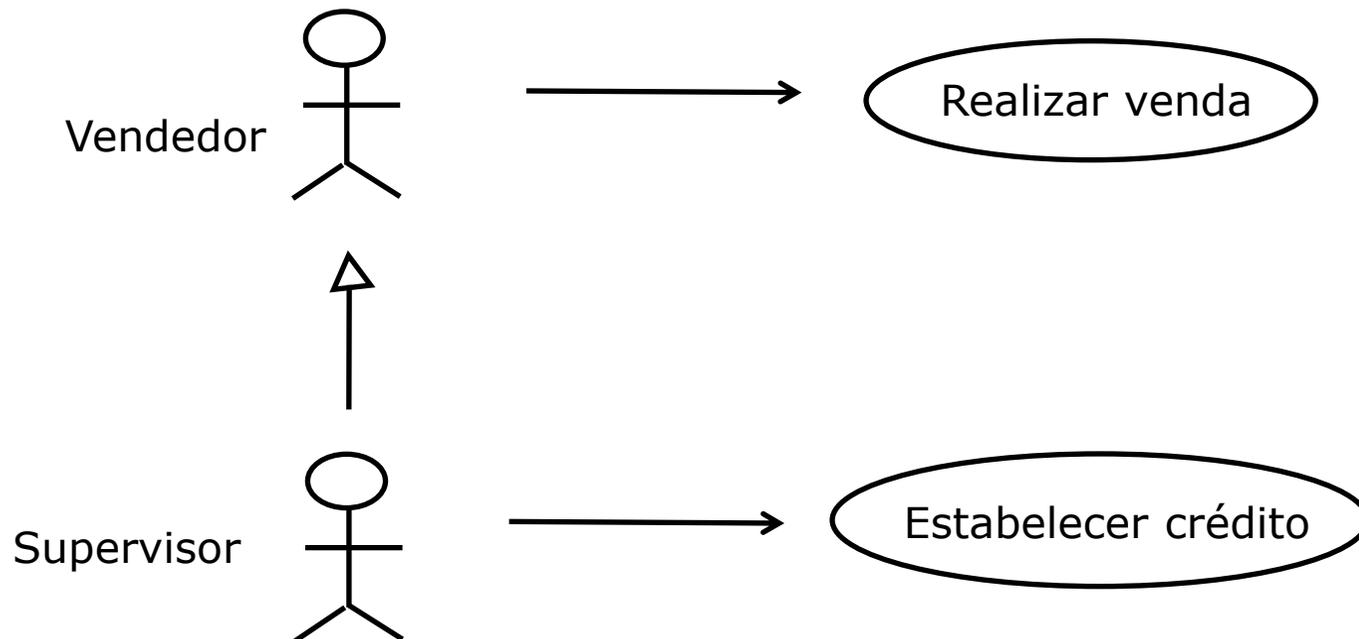
Por que estruturar o modelo?

- ◆ Extrair descrições de funcionalidades genéricas e compartilhadas que podem ser usadas por mais de um caso de uso;
- ◆ Extrair descrições de funcionalidades adicionais que possam estender descrições específicas;
- ◆ Facilitar o entendimento do modelo.
- ◆ O modelo não deve ser estruturado muito cedo!

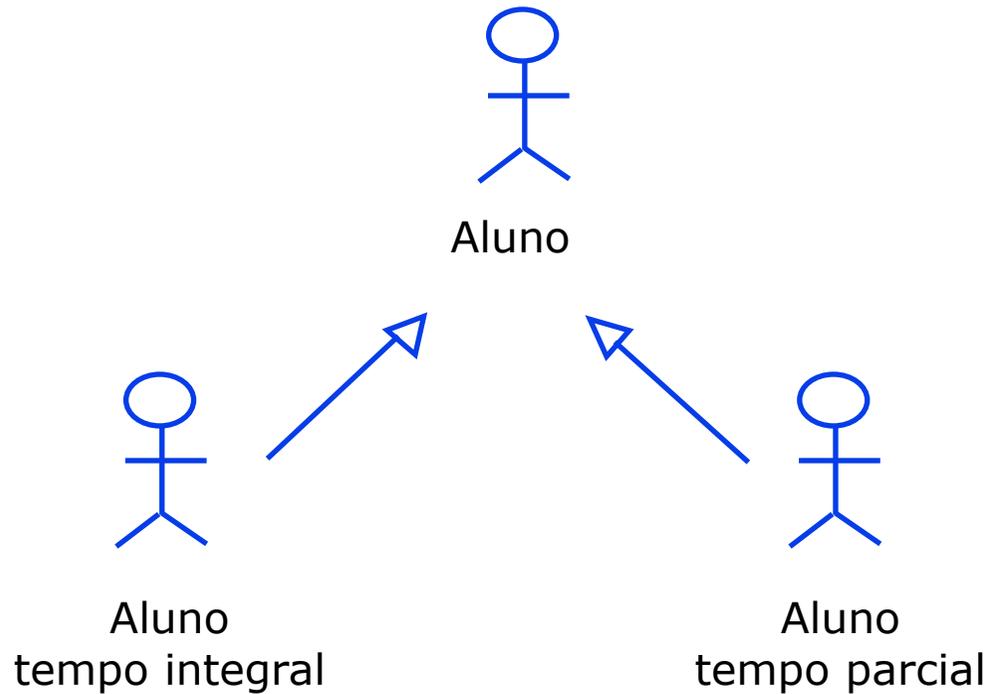


Generalização de Atores

- ◆ É possível definir tipos gerais de atores e especializá-los usando o relacionamento de especialização



Generalização de Atores: outro exemplo



Relacionamentos entre casos de uso

- ◆ Inclusão
- ◆ Extensão
- ◆ Generalização

Inclusão de casos de uso

- ◆ Use inclusão quando houver repetição entre casos de uso e você desejar evitar esta repetição.
- ◆ Um caso de uso incorpora explicitamente o comportamento de outro caso de uso, evitando assim repetições de descrição de fluxos.

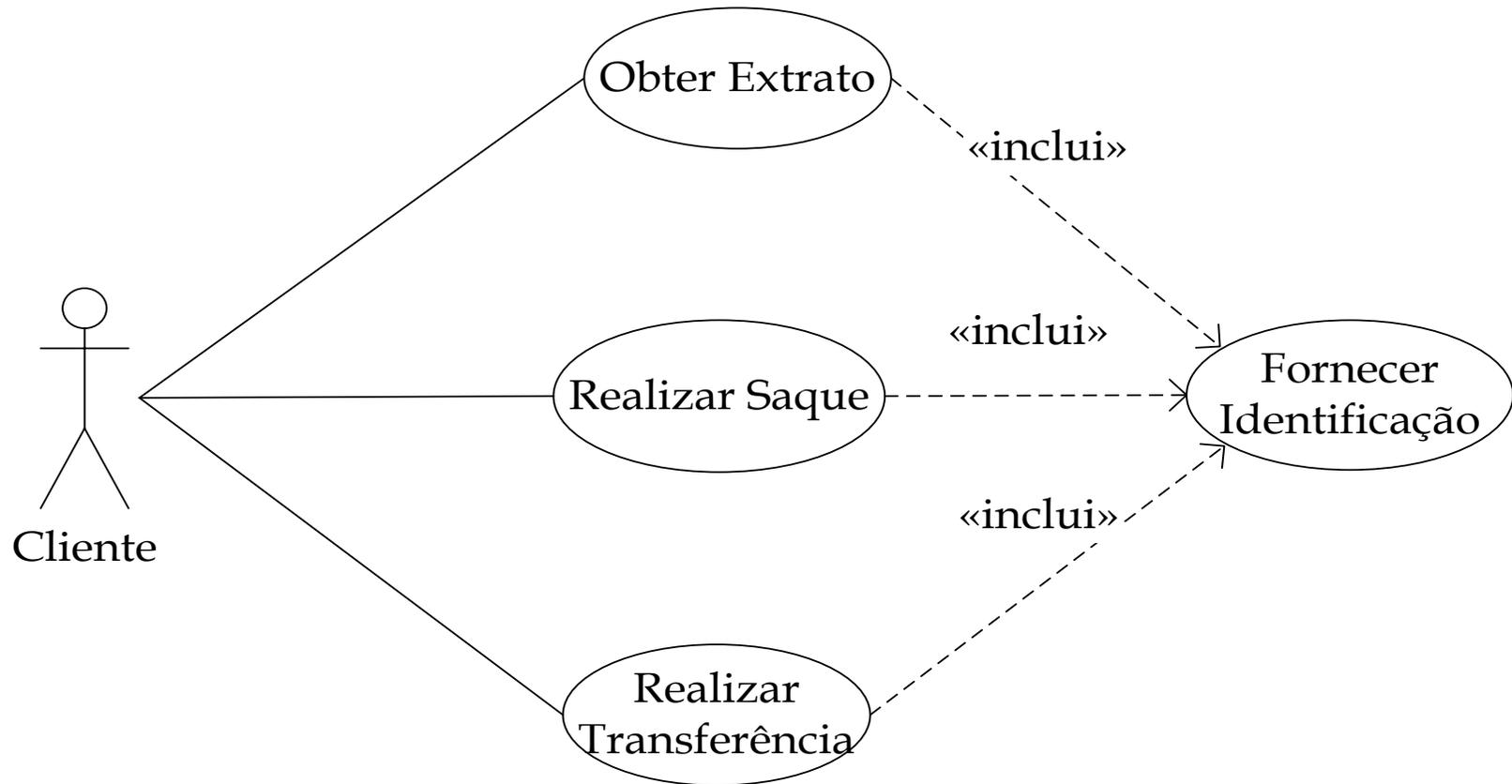
Relacionamento de inclusão

- ◆ *Existe somente entre casos de uso.*
- ◆ *Analogia útil: rotina.*
 - *Em uma linguagem de programação, instruções podem ser agrupadas em uma unidade lógica chamada rotina.*
 - *Sempre que essas instruções devem ser executadas, a rotina correspondente é chamada.*
- ◆ *Quando dois ou mais casos de uso incluem uma seqüência de interações comum, esta seqüência comum pode ser descrita em um outro caso de uso.*

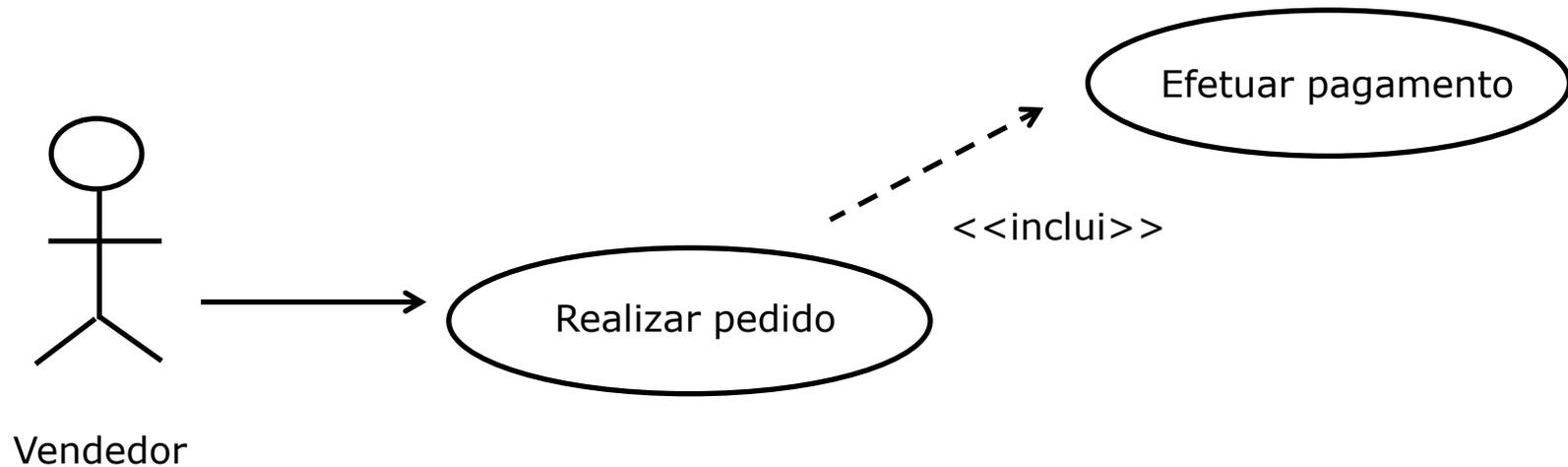
Relacionamento de inclusão

- ◆ *Este caso de uso comum:*
 - *evita a descrição de uma mesma seqüência de interações mais de uma vez e*
 - *torna a descrição dos casos de uso mais simples.*
- ◆ *Um exemplo: considere um sistema de controle de transações bancárias. Alguns casos de uso deste sistema são **Obter Extrato, Realizar Saque e Realizar Transferência.***
 - *Há uma seqüência de interações em comum: a seqüência de interações para validar a senha do cliente.*

Notação



Inclusão de casos de uso: exemplo



Exemplo de inclusão: validação de cliente no sistema

Caso de uso: Realizar Saque

O cliente seleciona a opção “sacar”

O cliente informa o valor a ser sacado

<inclui> Fornecer Identificação

O cliente recebe o dinheiro

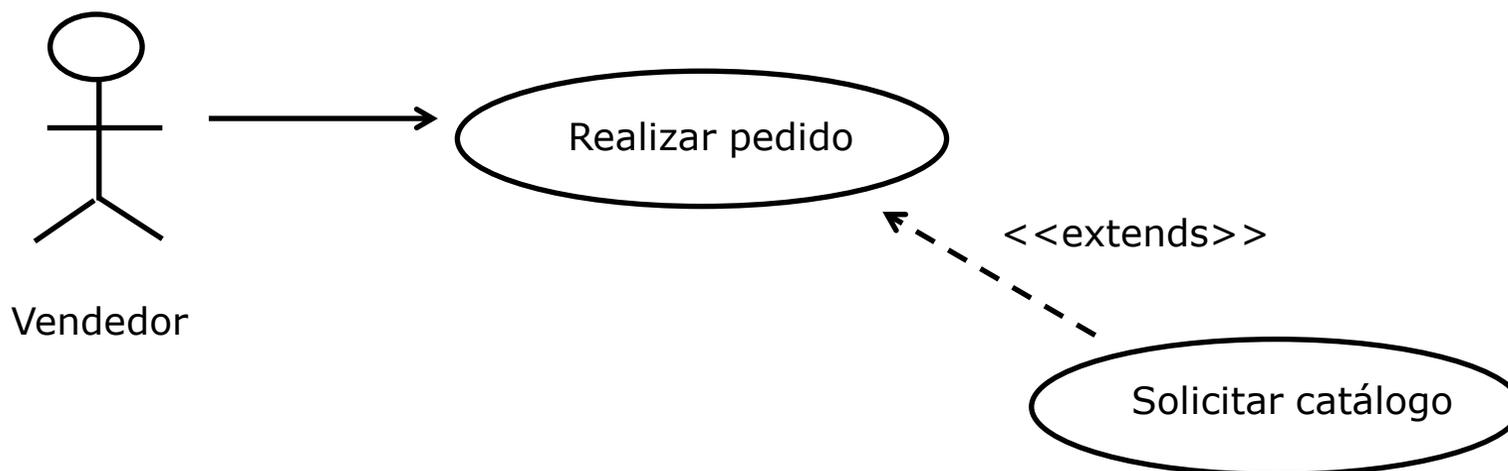
Caso de uso de Inclusão: Fornecer Identificação

O cliente informa a senha e passa o cartão

O sistema valida a senha e os dados do cartão

Extensão de casos de uso

- ◆ Use extensão quando quiser descrever uma variação do comportamento normal.
 - partes opcionais de casos de uso
 - cursos alternativos e complexos que raramente ocorrem



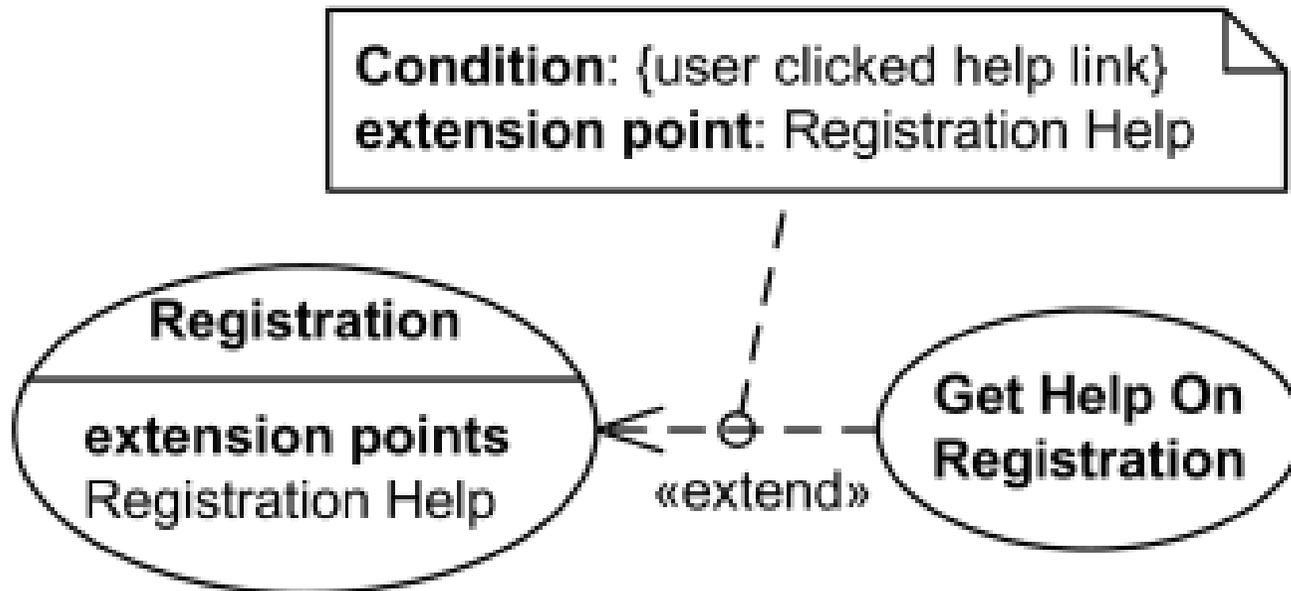
Relacionamento de extensão

- ◆ Utilizado para modelar situações onde diferentes seqüências de interações *podem* ser inseridas em um caso de uso.
- ◆ Sejam A e B dois casos de uso.
 - Um relacionamento de extensão de B para A indica que um ou mais dos cenários de A *podem* incluir o comportamento especificado por B.
 - Neste caso, diz-se que B *estende* A.
 - O caso de uso A é chamado de *estendido* e o caso de uso B de *extensor*.

Relacionamento de extensão

- ◆ Cada uma das diferentes seqüências representa um comportamento *opcional*, que só ocorre sob certas condições ou cuja realização depende da escolha do ator.
- ◆ Quando um ator opta por executar a seqüência de interações definida no extensor, este é executado.
 - Após a sua execução, o fluxo de interações volta ao caso de uso estendido, recomeçando logo após o ponto em que o extensor foi inserido.
- ◆ Importante: não necessariamente o comportamento definido pelo caso de uso extensor é realizado.

Extensão de casos de uso



Generalização de casos de uso

- ◆ Relaciona um caso de uso especializado a um mais geral
- ◆ O filho herda os atributos, operações e seqüências 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

Generalização de casos de uso: exemplo

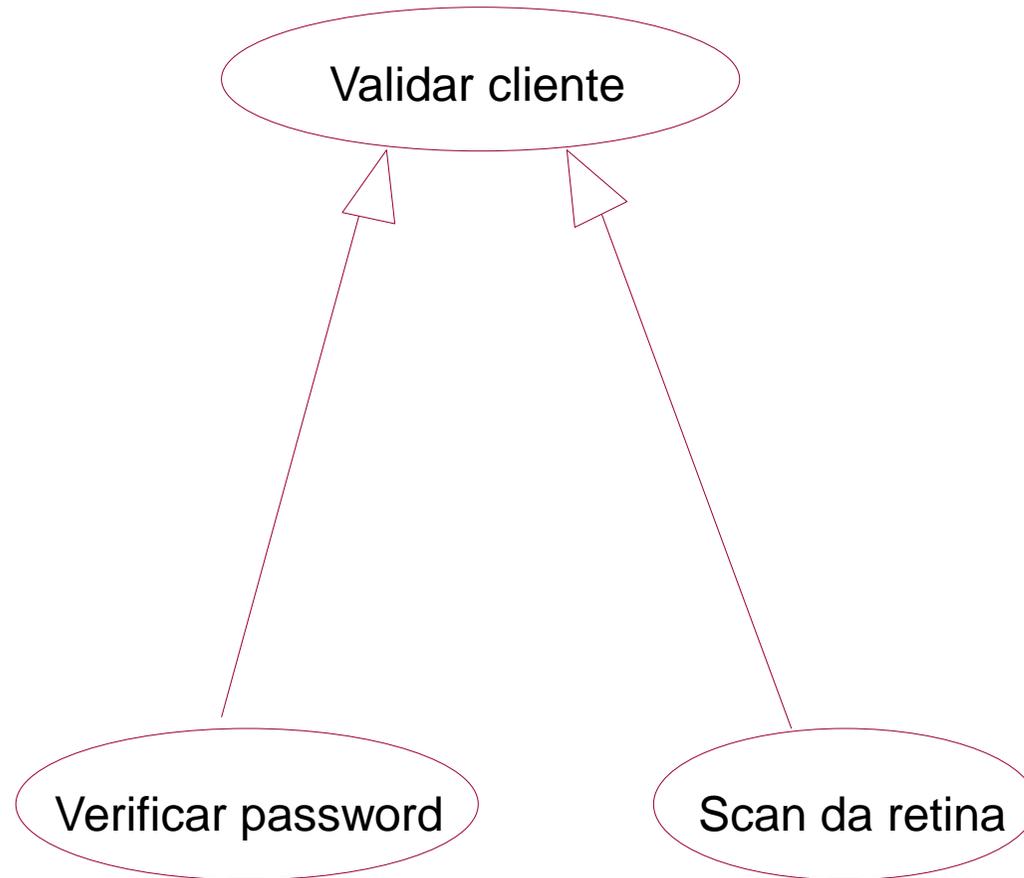


Diagrama de casos de uso estruturado: exemplo

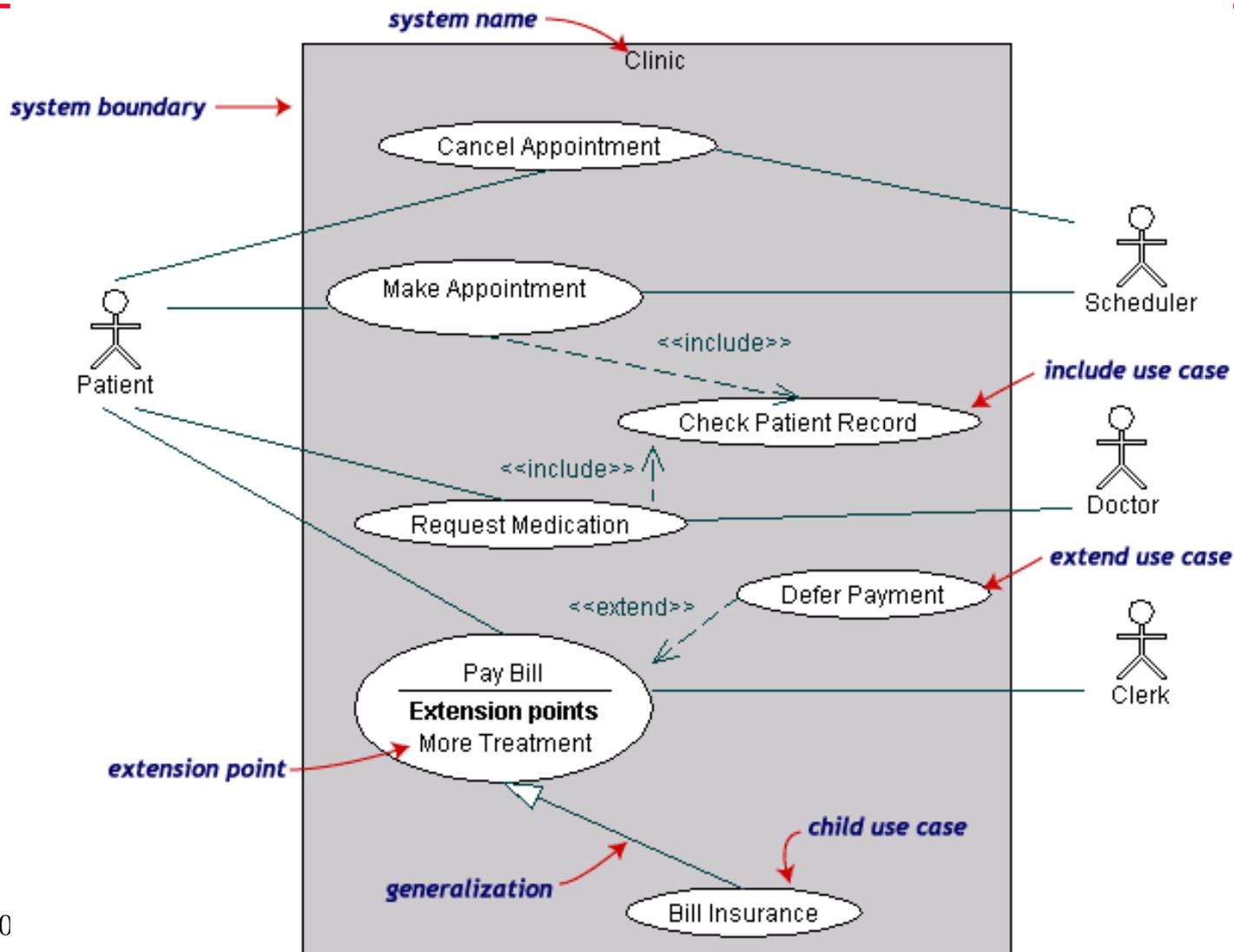
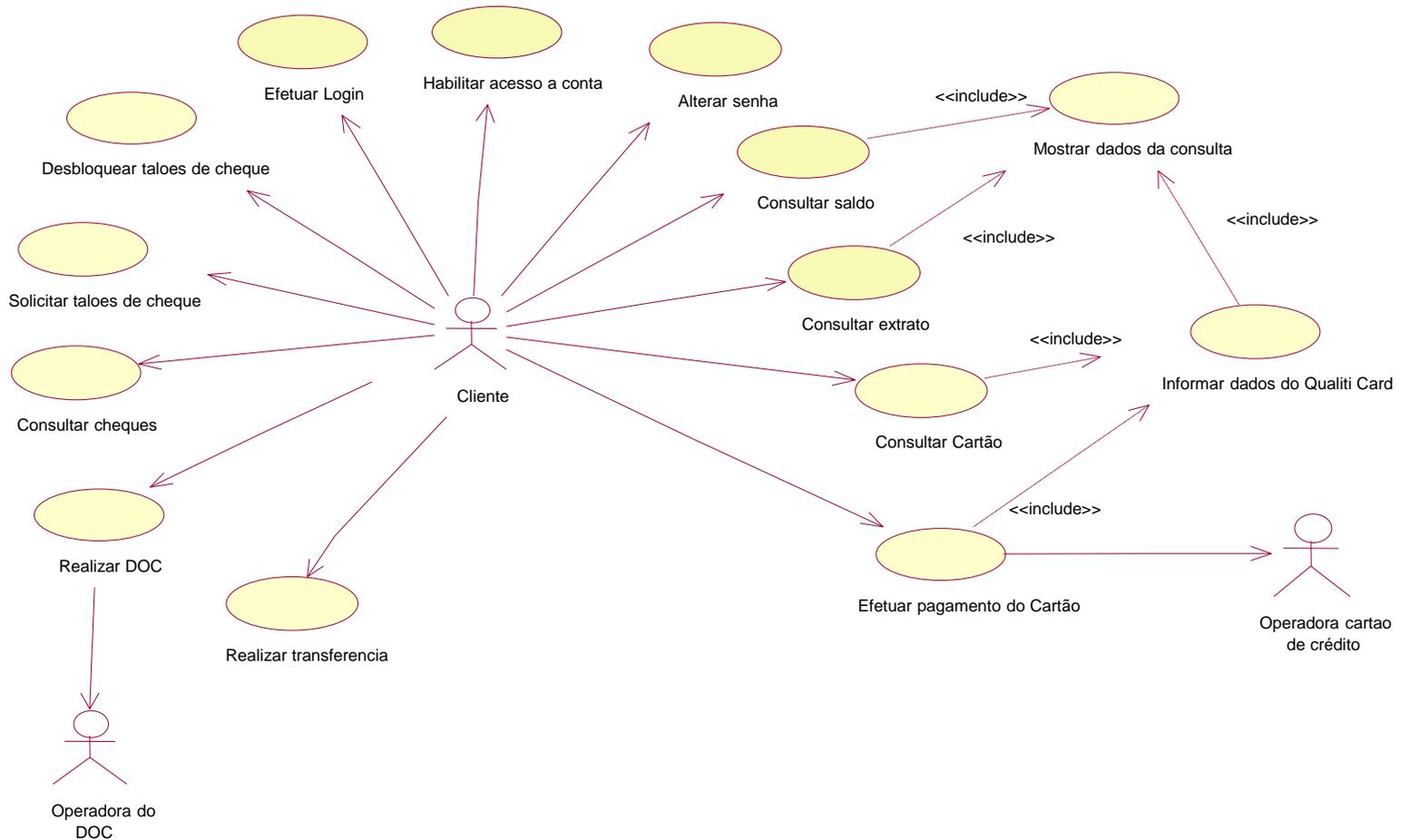


Diagrama de casos de uso estruturado: outro exemplo



Exercício (Opcional)

- ◆ Produza um diagrama de casos de uso estruturado do Amazônia.

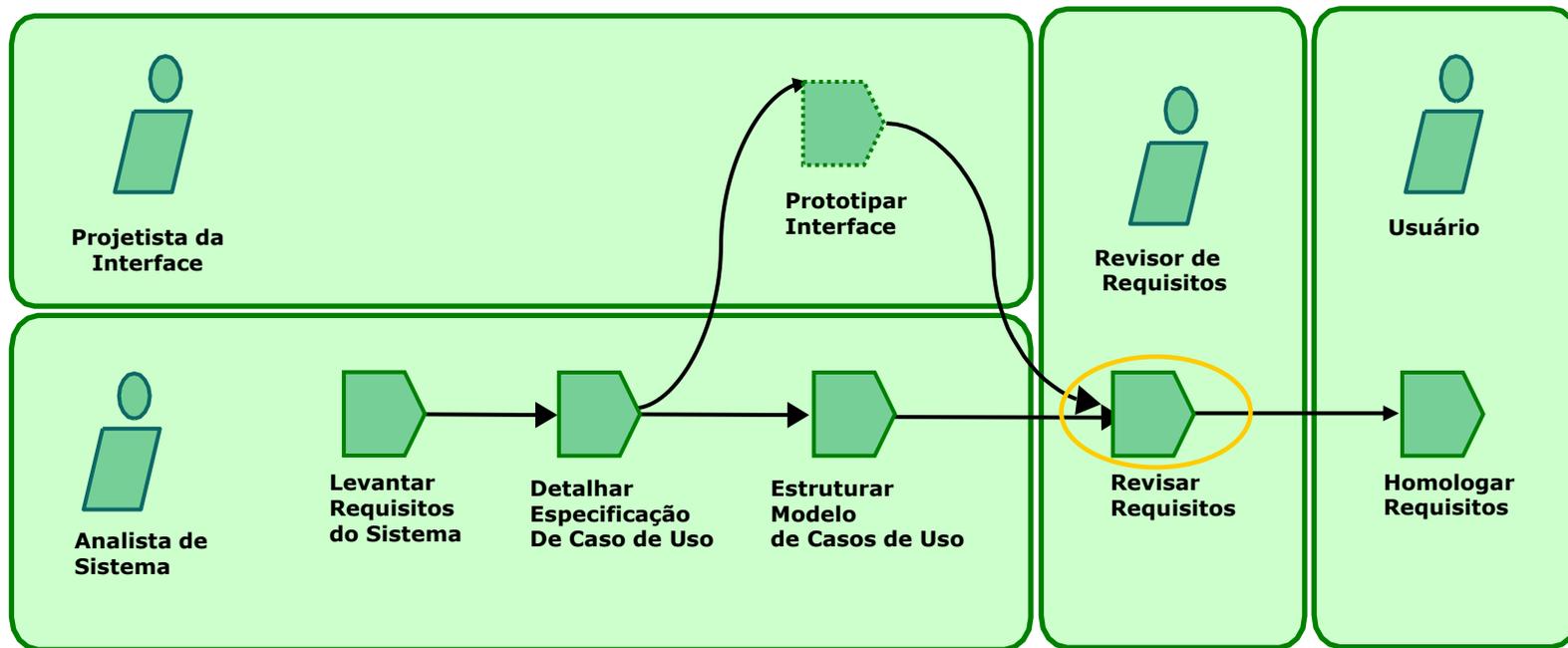


Revisão dos Requisitos

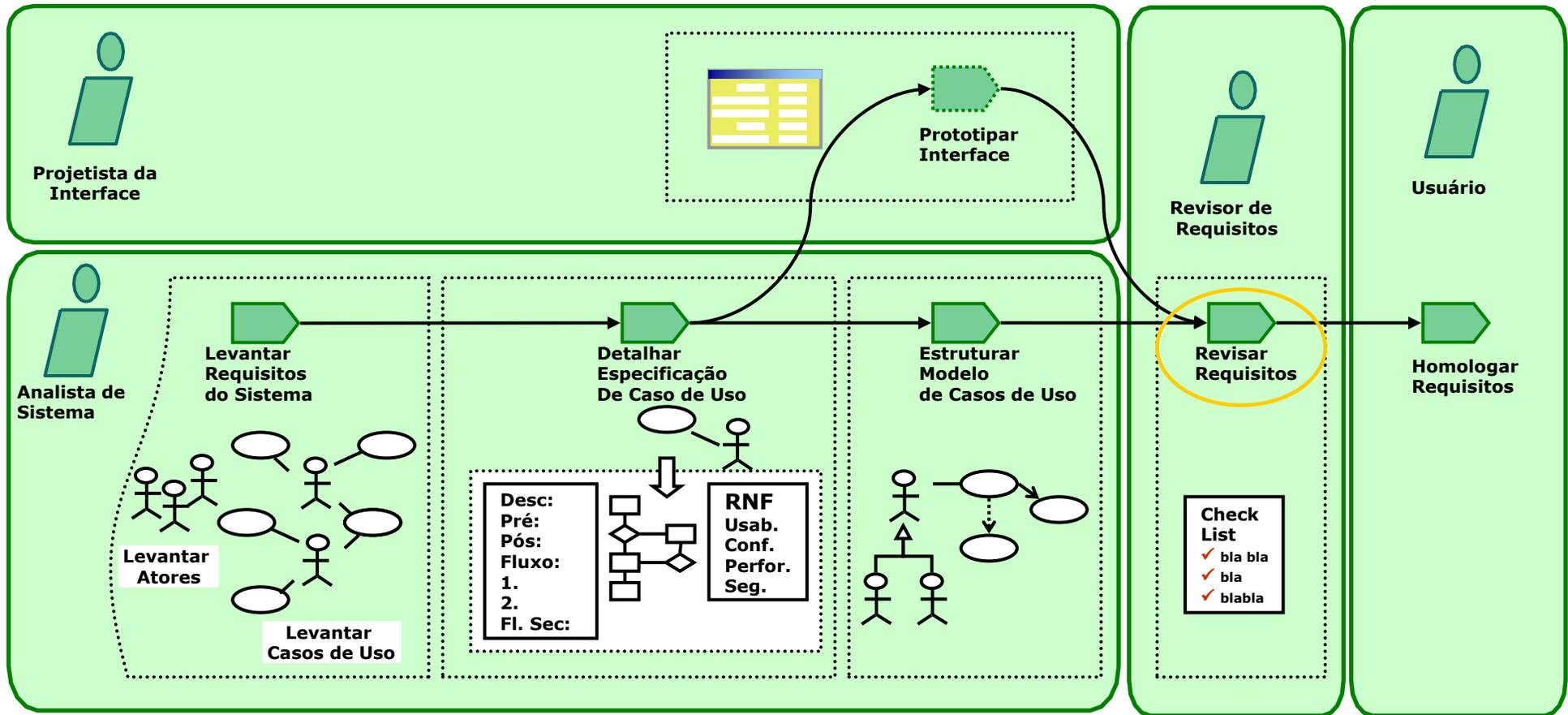
Objetivos

- ◆ Apresentar um checklist para verificação da qualidade do modelo de casos de uso

Revisar Requisitos



Revisar Requisitos



Nomeação de atores e casos de uso

- ◆ Devem ser únicos!
 - cuidado ao definir novos nomes!
- ◆ Nomes de atores
 - devem descrever claramente o papel do ator
- ◆ Nomes de casos de uso
 - devem indicar o resultado do caso de uso
 - use quantas palavras for necessário!

Casos de Uso devem ser

- ◆ Unidades testáveis e auto-contidas!
- ◆ Isso facilita:
 - distribuição de tarefas entre os desenvolvedores
 - gerenciamento do cronograma
 - planejamento e realização de testes unitários
 - integração do sistema
- ◆ Sem isso, não é viável um desenvolvimento iterativo e incremental!

Checklist - Revisão dos Atores

- ◆ Todo ator está relacionado a pelo menos um caso de uso?
- ◆ Você pode nomear pelo menos 2 pessoas que atuem como um ator específico?
- ◆ 2 ou mais atores possuem papéis similares em relação ao sistema?
- ◆ 1 determinado ator usa o sistema de várias maneiras diferentes ou tem vários objetivos ao usar o sistema?
- ◆ Os nomes dos atores são únicos, descritivos e intuitivos e correspondem aos seus papéis?

Checklist - Revisão dos Casos de Uso

- ◆ Todo caso de uso está relacionado a pelo menos um ator? E fornece um resultado de valor?
- ◆ Todo caso de uso é independente dos outros?
- ◆ Parte do fluxo de eventos de um caso de uso já foi especificado em outro caso de uso?
- ◆ 2 ou mais casos de uso possuem fluxos de eventos muito similares?

Checklist - Revisão dos Casos de Uso

- ◆ Os nomes dos casos de uso são únicos e intuitivos e descrevem os seus comportamentos? Eles não correm o risco de serem confundidos no futuro?
- ◆ Os casos de uso correlacionados estão agrupados em pacotes, facilitando o seu entendimento?

Os requisitos SEMPRE mudam

- ◆ Atualizar a documentação é fundamental!
- ◆ Lembre-se que os casos de uso serão utilizados para testes e documentação do usuário!!!

