

CC6521
Modelagem de Sistemas

Requisitos e Casos de uso



Prof. Ricardo Destro
1º Semestre de 2016

Requisito : uma condição ou capacidade que um sistema deve apresentar

- Funcionalidades
- Qualidades

Necessária ao usuário do sistema, para resolver um problema e alcançar um objetivo

Aquilo que deve ser definido
antes de começar a
construir um sistema





Eu quero alguma coisa para atravessar a cidade no menor tempo possível

Definição dos Requisitos

centro universitário
FEI

A cartoon illustration shows two stick figures shaking hands. The figure on the left has a speech bubble that says 'Eu quero alguma coisa para atravessar a cidade no menor tempo possível'. The figure on the right has a thought bubble showing a person riding a bicycle.



Mas eu não quero me molhar! Como vou carregar minha pasta ?

Mudança nos Requisitos

centro universitário
FEI


A cartoon illustration shows a stick figure sitting at a desk with a document. Another stick figure stands next to a bicycle, holding a briefcase. A speech bubble from the person with the bicycle says 'Mas eu não quero me molhar! Como vou carregar minha pasta ?'.



Re-projeto



Entrega do Sistema

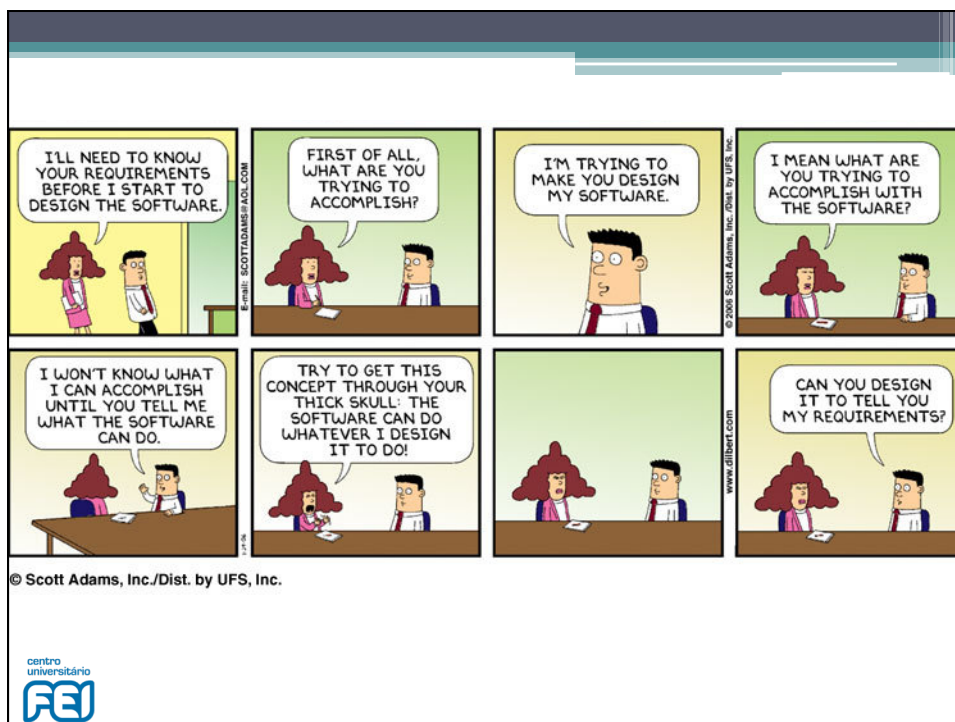
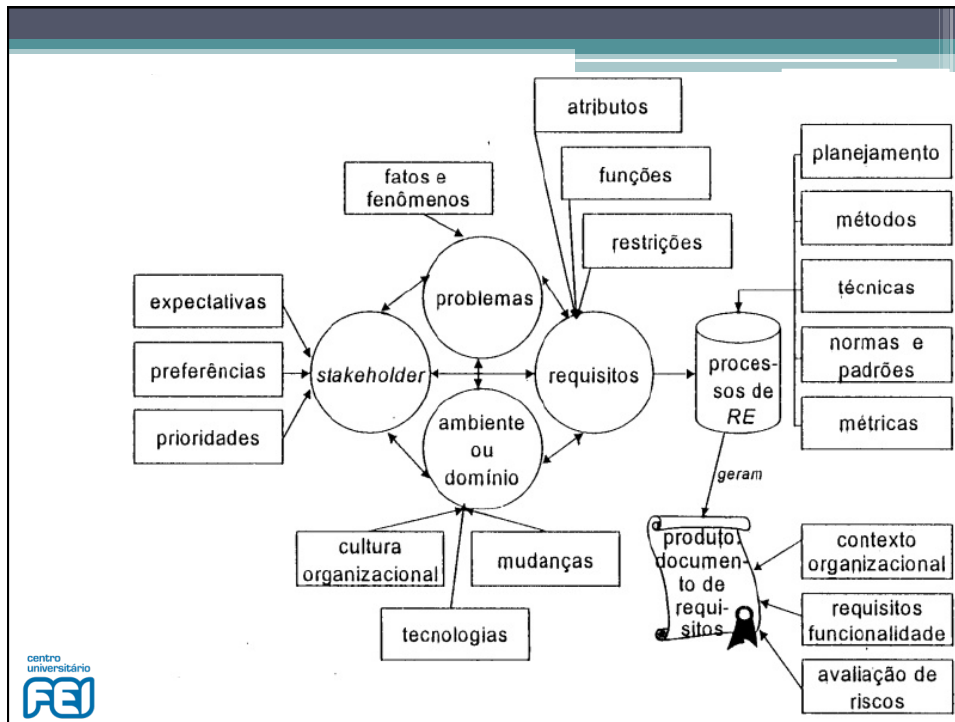


Como o cliente explicou...

FEI

Levantamento dos Requisitos do Sistema

- Base do desenvolvimento do Sistema
- Solução de problemas existentes: começo da implementação de uma mudança na organização
- Complexa:
 - Envolvimento de pessoas com pontos de vista conflitantes
 - Uso de novas tecnologias
- Necessidade de uma linguagem comum aos envolvidos



Requisitos

- **Funcionais**
 - Descrevem as funcionalidades que se espera que o sistema disponibilize, de uma forma completa e consistente.
 - Relacionados a Entradas, Funções, Saídas, Atores.
- **Não-funcionais**
 - Referem-se às restrições nas quais o sistema deve operar ou propriedades emergentes do sistema (como viabilidade ou tempos de resposta).
 - Tipos
 - Produto (Eficiência, Portabilidade, Segurança, etc.);
 - Organizacionais (Padrões, Entrega, etc.);
 - Externos (Aspectos Éticos, Legais, etc.).

Requisitos não funcionais

- | | |
|--|--|
| <ul style="list-style-type: none"> • Operacional <ul style="list-style-type: none"> • ambiente operacional • condições do usuário • sistemas relacionados • Segurança <ul style="list-style-type: none"> • Confidencialidade • Integridade • Disponibilidade • Legal <ul style="list-style-type: none"> • Leis • Regulamentações • normas existentes • Suporte <ul style="list-style-type: none"> • Capacidade manter o sistema atualizado | <ul style="list-style-type: none"> • Usabilidade <ul style="list-style-type: none"> • facilidade de uso pelos usuários • Confiança <ul style="list-style-type: none"> • frequência e resistência a falhas • capacidade de recuperação • Predibilidade • Precisão • Desempenho <ul style="list-style-type: none"> • Capacidade • taxas em relação ao tempo (tempo de resposta, disponibilidade) • Precisão • Uso de memória • Aparência <ul style="list-style-type: none"> • Visual • Design gráfico |
|--|--|

Problemas

- Grande parte dos problemas de um projeto decorre de:
 - Falta / Ineficiente compreensão dos requisitos;
 - Pouco / Inexistente feedback do cliente;
 - Requisitos mal especificados.

Possíveis soluções

- Feedback
 - Contar sempre com o cliente próximo na hora de especificar/validar um requisito.
- Casos de Uso
 - Descrição e/ou Diagrama UML.
- Prototipação
 - Ferramentas RAD (Rapid Application Development);
 - Paper Prototype – rápida e feedback imediato.

Requisitos

Gerar nota de restituição

Identificação:	Nome:
RF 018	Gerar nota de restituição
Descrição:	
O usuário pode gerar uma nota que será enviada via correios para contribuintes que tenham direito a restituição. Na nota deve constar o endereço do imóvel correspondente e os dados do proprietário, além de informar os passos para realizar a solicitação de restituição do valor informado, juntamente com o valor a ser restituído.	
Usuários: DPLAN e ROOT	
Essencial	Importante
	Desejável



Requisitos - Matriz Rastreabilidade

- Código
- Prioridade
- Tipo
- Nome
- Descrição
- Justificativa
- Critérios de Aceitação
- Solicitante
- Status
- Data do Status
- Requisitos Relacionados



Requisitos - Matriz Rastreabilidade

• Código	ID sequencial...eventualmente identifica o tipo do requisito
• Prioridade	Prioridade do requisito: Imprescindível, Importante, Desejável, Indiferente
• Tipo	
• Nome	Tipo do requisito: funcional, não funcional, técnico, qualidade, custo, inverso etc
• Descrição	
• Justificativa	Como testar ou validar o requisito
• Critérios de Aceitação	
• Solicitante	"responsável" pela aceitação do requisito.
• Status	
• Data do Status	
• Requisitos Relacionados	Código dos requisitos relacionados Rastreabilidade



Casos de uso



Use Case e Atores

- Um *use case* (caso de uso) é uma seqüência de ações executada pelo sistema que gera um resultado de valor observável para um ator particular;
- Um *ator* é alguém ou alguma coisa fora do sistema que interage com o sistema;



Algumas Definições

- Ações
 - Procedimento computacional ou algorítmico acionado quando o ator fornece um sinal para o sistema (mensagem) ou quando o sistema recebe um evento de tempo.
 - Ações são atômicas.
- Seqüência de Ações
 - Fluxo de eventos específicos através do sistema.
 - Devem-se agrupar fluxos de eventos similares em *use case* único.
- Execução do Sistema
 - O que o sistema faz de forma a realizar a seqüência de ações.
 - Ações internas ao sistema.



Algumas Definições

- **Resultado de Valor Observável**
 - A seqüência de ações deve gerar algo que tenha valor para um ator do sistema.
 - Um ator não deveria realizar vários *use cases* para obter algo útil.
- **Ator Particular**
 - Entidade individual que força o isolamento do valor fornecido a grupos de usuários específicos do sistema, garantindo que o sistema faz o que se espera dele.



Use Case e Ator

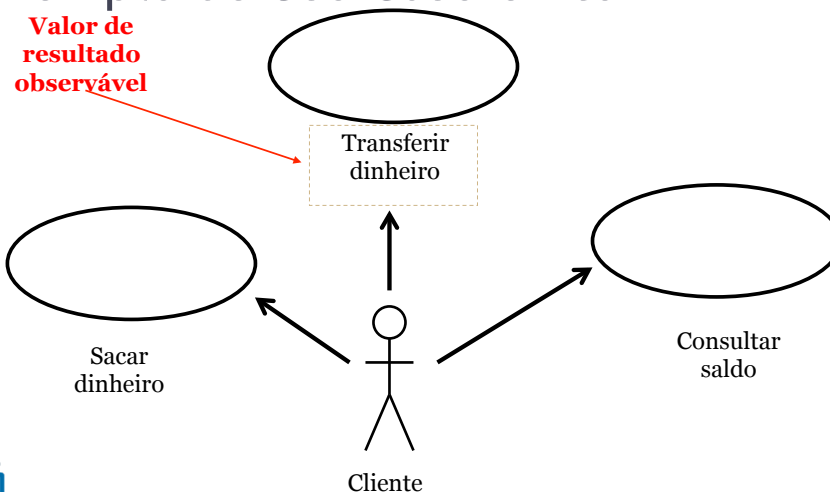
- A descrição de um *use case* define o que o sistema faz quando o *use case* é realizado;
- A funcionalidade do sistema é definida por um conjunto de *use cases*, cada um representando um fluxo de eventos específico;



Exemplo de *Use Case* e Ator

- Um cliente de um banco pode usar um caixa automático (CA) para sacar dinheiro, transferir dinheiro ou consultar o saldo da conta
- Ator: **Cliente**
- *Use cases*: **Sacar dinheiro, transferir dinheiro e consultar saldo**

Exemplo de *Use Case* e Ator



Identificando *Use Cases*

- Em geral é difícil decidir se um conjunto de interações usuário-sistema é um ou são vários *use cases*;
- Por exemplo, seriam *use cases*
 - Inserir cartão em um ATM?
 - Entrar com a senha?
 - Receber o cartão de volta?
- *Use cases* têm que representar valor para o ator;
- Portanto, transferir dinheiro, sacar, depositar e consultar saldo seriam *use cases*;



Identificando *Use Cases*

- Determinam-se a partir de interações ou agrupamento de sequência de ações com o sistema que resultam valores para atores;
- Uma alternativa seria que um *use case* satisfaz um objetivo particular de um ator que o sistema deve prover;
- A razão para agrupar certas funcionalidades e chamá-las de *use cases* é facilitar seu gerenciamento em conjunto durante todo o ciclo de desenvolvimento;



Evolução de *Use Cases*

- Inicialmente, *use cases* podem ser tão simples que apenas um esboço sobre seu funcionamento é suficiente;
- Porém, com a sedimentação da modelagem, uma descrição detalhada de seu fluxo de eventos faz-se necessária;
- O fluxo de eventos deve ser refinado até que todos os *stakeholders* envolvidos estejam de acordo com a descrição;



Organizando *Use Cases*

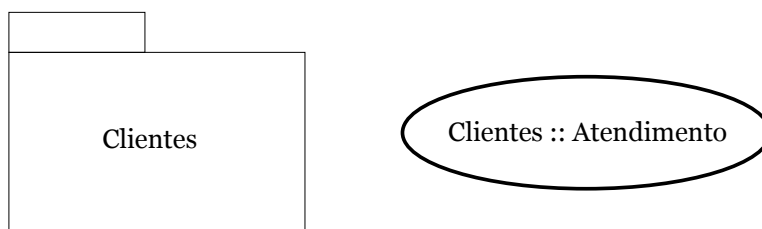
- Um sistema pequeno pode ser expresso através de uns seis *use cases*, envolvendo dois a três atores;
- Porém, para sistemas maiores, princípios de estruturação e organização devem ser usados;
- Caso contrário, atividades como planejamento, atribuição de prioridades, etc., podem se tornar difíceis;



Pacote de *Use Case*

- Pacote de *use case* é o primeiro princípio de estruturação;
- Agrupam-se use cases relacionados em um *container*;
- Pode-se ainda explorar relacionamentos entre *use cases* a partir de seus fluxos de eventos;

Pacote de *Use Cases*



Explorando Reuso em *Use Cases*

- Se o comportamento é comum a mais de dois *use cases* ou se ele forma uma parte independente
- Então pode-se modelá-lo como um *use case* em si para ser reusado por outros
- Existem três maneiras de reusar:
 - Inclusão
 - Extensão
 - Generalização/Especialização

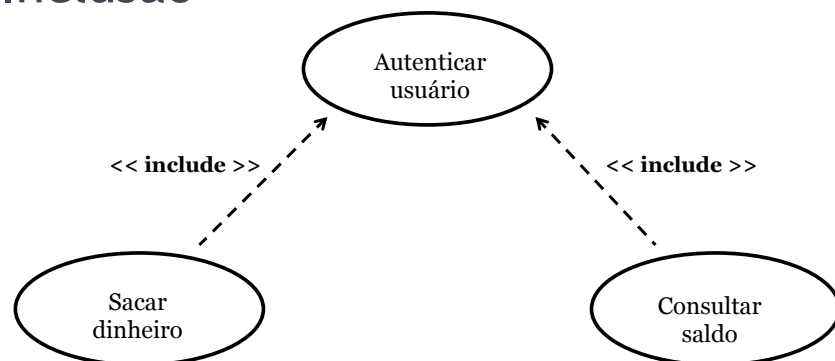


Inclusão

- Vários use cases podem incluir um texto de fluxo de eventos comum organizado como um use case;
- Equivalente a fatoração feita em programação através de sub-programas;
- Por exemplo, tanto “Sacar dinheiro” quanto “Consultar saldo” necessitam da senha;
- Poder-se-ia criar “Autenticar usuário” e incluí-lo nos use cases anteriores;
 - Mas atenção, não se deve criar use cases mínimos.
- O que importa é que um use case tenha algum valor para ator particular



Inclusão



Inclusão

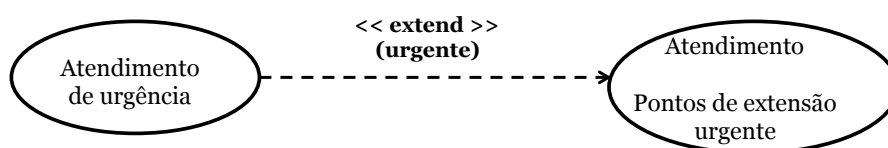
- Descrição de **Consultar saldo**
 - Fluxo de Eventos Principal:
 - **include (Autenticar usuário)**. Sistema pede a Cliente que selecione tipo de conta (corrente, etc). ...

Extensão

- Um *use case* pode ser estendido por outro (estender funcionalidade);
- A extensão ocorre em pontos específicos chamados de **pontos de extensão**;
- Extensão se dá pela inclusão de texto adicional (fluxo de eventos), nos pontos de extensão sob condições particulares;
- Pode ser usada para simplificar fluxos de eventos complexos, representar comportamentos opcionais e lidar com exceções;



Extensão



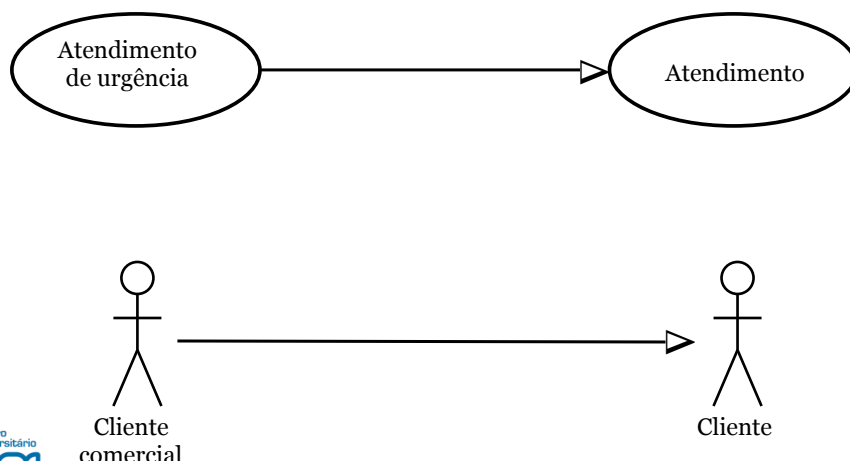
Extensão

- Descrição de **Atendimento**
 - **Fluxo de Eventos Principal:**
 - Colete os itens do pedido. **(urgente)**. Submeta pedido para processamento.

Especialização

- Um *use case* pode especializar um outro através da adição ou refinamento do fluxo de eventos original
- Especialização fornece uma maneira de modelar o comportamento de estruturas de aplicação em comum
- Especialização é o caminho oposto ao de generalização

Especialização



Fluxo de Eventos

- A parte mais importante de um *use case* na atividade de requisitos é seu fluxo de eventos;
- O fluxo de eventos define a seqüência de ações entre o ator e o sistema;
- Geralmente escrito em linguagem natural, com um uso preciso de termos definidos no glossário de acordo com o domínio do problema;
- Também pode ser descrito usando texto formal (pré e pós-condições) ou pseudo-código;

Exemplo de Fluxo de Eventos

- Um esboço inicial sobre **Sacar dinheiro** seria
 1. O *use case* inicia quando o Cliente insere um cartão no CA. Sistema lê e valida informação do cartão
 2. Sistema pede a senha. Cliente entra com a senha. Sistema valida a senha.
 3. Sistema pede seleção do serviço. Cliente escolhe “Sacar dinheiro”
 4. Sistema pede a quantia a sacar. Cliente informa.
 5. Sistema pede seleção da conta (corrente, etc). Cliente informa.
 6. Sistema comunica com a rede para validar a conta, senha e o valor a sacar.
 7. Sistema pede remoção do cartão. Cliente remove.
 8. Sistema entrega quantia solicitada.



Fluxo de Eventos

- Quando se tenta descrever o que o sistema faz através de fluxos de eventos completos, surgem vários caminhos possíveis
- Existem caminhos alternativos e casos diferentes a considerar, e efeitos ou valores diferentes são produzidos
- Coletivamente, o fluxo de eventos do *use case* eventualmente descreve todos esses caminhos possíveis



Sub-fluxos de Eventos

- Pode-se ver um fluxo de eventos como vários sub-fluxos de eventos;
- Um sub-fluxo é tido como o principal e os demais como alternativos;
- O interessante dessa abordagem é o reuso de fluxos de eventos de certos *use cases* por outros *use cases*;



Exemplo de Sub-fluxos

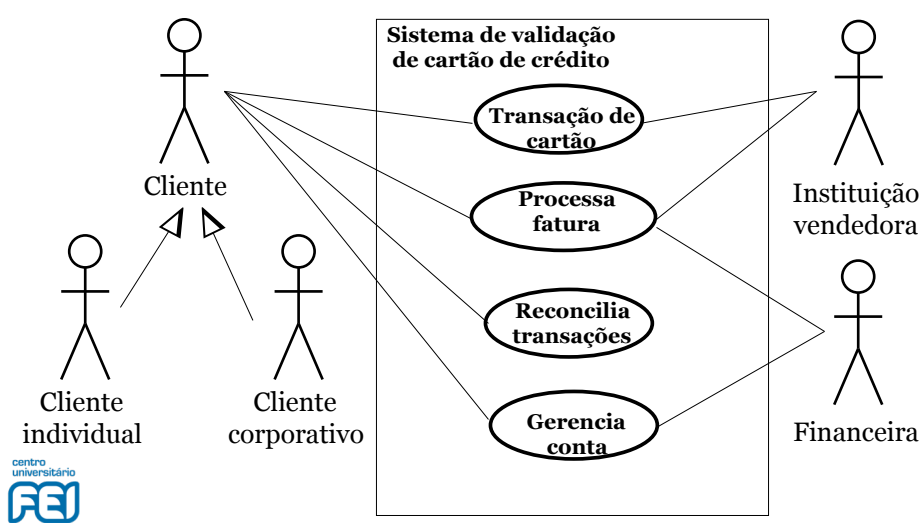
- Seja o *use case* **Validar usuário**
 - **Fluxo principal:**
 - O *use case* inicia quando o sistema pede ao Cliente a senha. Cliente entra com senha. Sistema verifica se a senha é válida. Se a senha é válida, sistema confirma e termina o *use case*.
 - **Fluxo excepcional:**
 - Cliente pode cancelar a transação a qualquer momento pressionando a tecla ESC, reiniciando o *use case*. Nenhuma modificação é feita na conta do Cliente.
 - **Fluxo excepcional:**
 - Se Cliente entra com senha inválida, o *use case* reinicia.



Diagramas de *Use Cases*

- Com o intuito de demarcar os limites da funcionalidade do sistema, *use cases* são organizados dentro de um diagrama;
- Quando do uso de diagramas de *use cases*, os atores são modelados através de relacionamentos de generalização/especialização;

Exemplo de Diagrama



Exemplo de Caso de uso

- Realizar um saque no caixa eletrônico

Identificação	UC_01
Função	Retirar Dinheiro do caixa eletrônico
Atores	Cliente, Caixa eletrônico
Prioridade	Essencial
Pré-condição	Cliente precisa ter em mãos o cartão do banco
Pós-condição	Dinheiro sacado com sucesso
Fluxo Principal	<ul style="list-style-type: none"> • Cliente insere cartão no dispositivo • Cliente digita a senha • Máquina autoriza login [FS001] • Cliente digita o montante • Máquina checa o saldo [FS002] • Máquina debita o dinheiro sacado do saldo inicial • Máquina dispõe cédulas para cliente • Máquina mostra na tela o novo saldo • Máquina ejeta cartão • Cliente retira cartão
Fluxo Secundário [FS001]	<ul style="list-style-type: none"> • Senha digitada é inválida • Máquina ejeta cartão • Cliente retira cartão
Fluxo Secundário [FS002]	<ul style="list-style-type: none"> • Saldo é menor que o montante requerido • Máquina mostra na tela o saldo • Máquina ejeta o cartão • Cliente retira o cartão

Dica Final: <<Include>>

- Use inclusão quando o mesmo comportamento se repete em mais de um caso de uso.
 - Esse comportamento comum deve ser definido em um novo caso de uso, o chamado caso de uso incluído.
 - Note que esse comportamento comum está contido em todos os cenários dos casos de uso.
 - A ligação ocorre com seta saindo dos casos de uso "inclusor" para o "incluído".
 - O "incluído" é esse que você criou devido o comportamento se repetir.

Dica Final: <<extend>>

- Use extensão quando um comportamento opcional de um caso de uso tiver de ser descrito.
 - Note que alguns cenários de caso de uso estendido podem não utilizar esse comportamento opcional.
 - O extensor (aquele caso de uso opcional) faz referência (ligação da seta) ao estendido
 - sendo que o estendido não sabe que o extensor existe.