

Relacionamento entre Casos de Uso

Relacionamentos entre Casos de Uso, principalmente para os profissionais que estão tendo o primeiro contato com o assunto, quase sempre geram alguma confusão. É natural.

As dúvidas sobre **Inclusão** (Include), **Extensão** (Extend) e **Generalização** [ou Herança] (Generalization) ocorrem com frequência, são comuns.

Vamos ao que significa cada um dos três tipos de relacionamento citados. Consideremos que temos três Casos de Uso – A, B e C, e com base nos três vamos descrever cada um dos relacionamentos.

Include

Quando o caso de uso **A** “inclui” o caso de uso **B**, significa que **sempre** que o caso de uso **A** for executado o caso de uso **B** também será executado. A direção do relacionamento é do caso de uso que está **incluindo** para o caso de uso **incluído**.

Extend

Quando o caso de uso **B** estende o caso de uso **A**, significa que quando o caso de uso **A** for executado o caso de uso **B** **poderá** (poderá – talvez não seja) ser executado também. A direção do relacionamento é do caso de uso **extensor** (aqui o caso de uso B) para o caso de uso **estendido** (aqui o caso de uso A).

Generalization

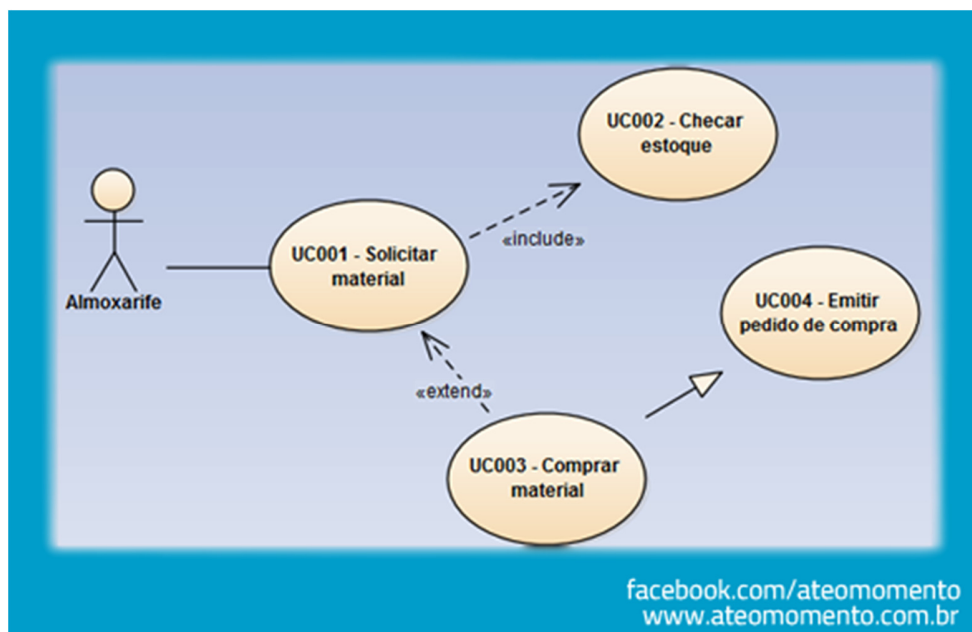
Quando o caso de uso **B** generaliza o caso de uso **C** isso significa que, além de fazer tudo que nele está especificado (ele = B), ele **também executará tudo que está especificado** no caso de uso **C**.

Muitos profissionais falam que isso não deve ser compreendido como a herança da orientação a objetos, mas na minha opinião deve ser sim, apenas (em tempo de modelagem de caso de uso) estamos num nível de abstração diferente, mas o produto final desta modelagem será software codificado.

A direção do relacionamento é sempre do **generalizador** (aqui o caso de uso B) para o **generalizado** (caso de uso C).

Exemplificando

Abaixo um diagrama com um cenário semelhante ao utilizado acima, ilustrando os relacionamentos.



No diagrama temos quatro Casos de Uso, e três relacionamentos diferentes: Include, Extend e Generalization.

Explicando o Include

O caso de uso “Solicitar Material” faz include no caso de uso “Checar Estoque”. Isso se dá porque **sempre** que houver a solicitação de material **sempre** haverá a consulta ao estoque para saber se o material está disponível.

Se sempre haverá, o relacionamento correto é o include.

Explicando o Extend

O caso de uso “Comprar Material” estende o caso de uso “Solicitar Material”. Isso se dá porque quando houver a solicitação de material, **caso o material não exista em estoque** (após consulta via o caso de uso “Checar estoque”) **poderá** ser solicitado a compra do item.

Mas também poderá não ser solicitada a compra, pois o item pode existir em estoque. Se **poderá** ser solicitada a compra (e não **sempre** será solicitada a compra) o relacionamento correto é o extend.

Explicando o Generalization

O caso de uso “Comprar Material” generaliza o caso de uso “Emitir pedido de compra”. Isso se dá porque no caso de uso “Emitir pedido de compra” **existee** especificação de como se realiza o pedido de compra, **processo que não se dá somente no contexto do almoxarifado**, mas é o mesmo em qualquer área do negócio.

Dessa forma, não justifica-se duplicar a especificação pertinente em outro caso de uso, basta **reaproveitar** o que já está pronto mas generalizado a ponto de poder ser aproveitado por alguém que o especialize.

A importância do uso correto nos relacionamentos

Especificações são feitas para serem **interpretadas**, e com base na interpretação, viabilizar a produção de software executável.

Quanto **mais qualidade** houver na especificação, **mais fácil** será de entendê-la, e **mais correta** será a interpretação de quem utilizá-la.

Essa facilidade gera **velocidade** na produção dos outros modelos (incluindo o modelo de código fonte, casos de teste etc.), **diminui a quantidade de defeitos** em potencial (quanto mais clara uma especificação, menor a chance dela ser interpretada de forma errada) e gera outros benefícios diversos.

Concluindo

A clareza e corretude nos relacionamentos entre os casos de uso influencia diretamente na qualidade do projeto.

Muitos profissionais acham que o diagrama serve apenas para “colar as bolinhas” para que alguém os identifique e consiga “ver o que tem dentro”, ou seja, ver os cenários do Caso de Uso. Vimos que vai muito além disso.

