

Sobre as pré-condições:

Primeiro vamos ao conceito de pré-condições: São CONDIÇÕES sobre o ESTADO DO SISTEMA necessárias para o caso de uso possa ser INICIADO. Então, imagine um software para um caixa de supermercado. Existe um caso de uso chamado Registrar Venda que corresponde ao que você vê quando vai a um supermercado ou a uma loja de departamentos, ou seja, desde a abertura da venda, passando pela identificação dos produtos e o registro do pagamento e emissão do cupom fiscal. O que você talvez não saiba é que todo dia esses "caixas" precisam ser abertos no início do dia (caso de uso Abrir Caixa) e fechados no final do dia (caso de uso Realizar Redução Z). Uma venda não pode ocorrer em um caixa que não tenha sido aberto. Portanto, nesse exemplo, teríamos que o caso de uso Registrar Venda tem a pré-condição "Caixa está disponível para venda" (que significa mais do que estar aberto, mas isso é uma outra discussão sobre o domínio de frente de loja). Note que não coloquei como pré-condição que o caso de uso Abrir Caixa deve ter sido executado, pois não interessa o que foi executado antes e sim a situação que o sistema precisa estar para que o caso de uso possa ser executado.

Portanto o que vocês colocaram como pré-condições corresponde a uma interação normal dentro do caso de uso. Repare que não é obrigatória a existência de pré-condições. Nem todo caso de uso possui pré-condições, o que significa que ele sempre pode ser executado (iniciado).

Fluxos alternativos devem ser descritos em uma seção à parte (uma seção Fluxo Normal e uma seção Fluxos Alternativos com os diversos fluxos alternativos).

Evite usar termos como clicar, pois isso denota uma forma específica de interface com o usuário.

No passo 4, O usuário **seleciona uma categoria de dificuldade**. Evite usar verbos como "deve". "da lista apresentada" é um detalhe de interface irrelevante aqui e, pior, não foi mencionada no passo anterior (daí vem a pergunta: foi apresentada quando?). Melhor não colocar esse detalhe. Veja que nos passos 5 e 6, vocês usaram a mesma terminologia (lista de contos). Melhor assim, mas de qualquer forma, poderia ficar simplesmente **Usuário escolhe um conto da lista**.

Os passos 7 a 11 podem ser reescritos da seguinte forma:

O sistema exibe sequencialmente as páginas do conto, seguindo a seguinte sequência para cada página:

Sistema exibe a página corrente do conto.

Para cada trecho destacado na página, usuário seleciona um trecho destacado na página.

Sistema exibe a dica associada ao trecho selecionado.

Usuário seleciona a opção de ir para a página seguinte.

Uma dúvida que eu fiquei: ele é obrigado a selecionar todas as dicas. Se isso for opcional, melhor colocar essa interação como um fluxo alternativo.

De onde vem as perguntas e respostas? Não haveria um cadastro ou importação de arquivo com as perguntas e respostas? Se sim, isso deveria ser um caso de uso.

No diagrama de classes:

Um vejo Conto associado a Pagina (1-*) e Pagina associada a Dica (1-*).

Dificuldade poderia ser representado ao invés de um número int como um tipo enumerado, conforme ilustrado abaixo. Dessa forma, o atributo em Conto seria dificuldade : Dificuldade.



Onde está o conteúdo (texto) do conto? Creio que está faltando uma estrutura do tipo Pagina contém Paragrafos e cada parágrafo possui um atributo texto. Para melhor associar a dica, um parágrafo poderia ter marcações (trecho do texto) e cada marcação estaria associada a uma dica.

A classe Quis não tem um atributo Pergunta, e sim um relacionamento com n Perguntas conforme já está no diagrama. Remova o atributo pergunta da classe Quis.

Idem para o atributo contoRelacionado.

Eu preferiria representar uma questão contendo o enunciado e um conjunto de opções de resposta onde uma delas seria a opção correta.

Acho que ficou faltando Classe representando a execução do quis, isto é, as respostas que um determinado usuário colocou para cada pergunta. O Historico não captura isso. Ainda em Histórico, remover os atributos usuário e quis.

Note que os atributos são removidos apenas do diagrama. Eles podem estar presentes na implementação em Java, pois em Java não existe um elemento que corresponda diretamente a uma associação. Dessa forma, uma associação do diagrama é implementada na forma de atributos em uma ou em ambas as classes.

A implementação ainda está muito preliminar. Vocês devem dar uma acelerada aqui, pois eu esperava ver algo mais do que foi entregue.