# Roteiro de exercícios sobre Requisitos, disciplina Engenharia de Software e Sistemas

## Paulo Borba

## Centro de Informática

## Universidade Federal de Pernambuco

**Adicione as respostas imediatamente após cada pergunta (inclusive nas questões que têm várias perguntas)**, usando uma outra cor **bem legível**. Leia com calma e atenção as instruções, e reflita com cautela sobre as consequências de cada comando.

## Aula 1

1. Quem você considera que seriam os *stakeholders* de um sistema de gerenciamento de metas (como Elicitar e especificar requisitos com qualidade) e conceitos (como MA, MPA, e MANA) dos alunos das disciplinas de graduação do Centro de Informática da Universidade Federal de Pernambuco?
2. Suponha que, em uma entrevista com um *stakeholder* de um sistema de controle acadêmico como o SIGA, ele diz que deseja uma funcionalidade de cálculo de média de um aluno. Que perguntas você faria para entender melhor a necessidade do *stakeholder* e especificar essa funcionalidade? Que perguntas você faria para entender que funcionalidades auxiliares seriam necessárias para o funcionamento da funcionalidade de cálculo de média?
3. Que *storyboards* ou esboços de tela você acha que deveria desenhar e mostrar aos *stakeholders* para validar inicialmente as necessidades dele com relação à funcionalidade de cálculo de média da questão anterior? Basta apresentar a descrição do que seriam os esboços, não os esboços em si.
4. Suponha que o *stakeholder* Marianaestabeleceu maior prioridade para a funcionalidade de cálculo de média, e que o *stakeholder* Pedro estabeleceu maior prioridade para a funcionalidade de geração de relatório com a listagem dos alunos de uma turma (apenas com o nome de cada aluno). Como você, no papel de analista de requisitos, deveria proceder para definir que *feature* especificar e implementar primeiro? E como você procederia se Pedro tivesse dado mais prioridade à feature de cadastro de notas dos alunos?
5. Como usuário do sistema de gerenciamento de metas, que requisitos não funcionais você acha que seriam importantes para você? Liste em ordem decrescente de prioridade, ou aplique um processo de triagem, se achar mais adequado.
6. Além dos requisitos funcionais, e dos requisitos não funcionais associados a fatores de qualidade de software, que outros tipos de restrições você acha que poderiam ser exigidas pelos *stakeholders* e seriam importantes considerar como requisitos não funcionais durante o desenvolvimento de um sistema?

## Aula 2

1. Veja o [*template*](https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=16&ved=2ahUKEwjlpvKhz9fgAhUPH7kGHX7mAOMQFjAPegQIABAC&url=https%3A%2F%2Fcsis.pace.edu%2F~marchese%2FCS775%2FRequirements%2520Specification%2520Template.doc&usg=AOvVaw2jGtqP_Gds0CWWl37tABQX) para documento de requisitos de uma instituição americana. Em particular, observe o índice e leia a Seção 3. Com base nisso, como você compara (em termos de detalhamento, dificuldade de escrita, nível de abstração, facilidade para derivação de testes, etc.) especificações de requisitos baseadas em documentos que seguem esse template com especificações baseadas em cenários?
2. Uma das funcionalidades importantes do sistema de gerenciamento de metas (SGM) é a auto-avaliação. Cada aluno deve ser capaz de atribuir o conceito (MA, MPA, e MANA) que ele acha que merece em cada uma das metas (Entender conceitos de requisitos, Especificar requisitos com qualidade, Entender conceitos de gerência de configuração, etc.) avaliadas. Durante a entrevista com os interessados, ficou decidido que seria interessante visualizar os conceitos atribuídos pelo professor na mesma tela de preenchimento da auto-avaliação. Também foi acertado que todas as metas têm que ser auto-avaliadas de uma só vez, não havendo possibilidade de auto-avaliação parcial. Considerando esses aspectos, siga as seguintes recomendações:
   1. Desenhe esboços de páginas do sistema relacionadas à essa funcionalidade (cole, logo abaixo, imagens dos esboços)
   2. Especifique, no formato ilustrado nos slides, a feature de auto-avaliação
   3. Especifique, no formato ilustrado nos slides, um cenário de GUI e um de serviço relatando o preenchimento bem sucedido da auto-avaliação
   4. Especifique, no formato ilustrado nos slides, um cenário de GUI e um de serviço relatando o preenchimento mal sucedido da auto-avaliação devido a ausência de conceitos a serem preenchidos pelo aluno para pelo menos uma das metas
   5. Verifique se os seus cenários estão seguindo rigorosamente as recomendações para escrita de cenários com qualidade (mencionadas nos *slides* e no livro texto). Se for necessário, altere as respostas dos itens anteriores.
3. Do ponto de vista dos professores que vão usar o sistema, é interessante ter uma funcionalidade para visualizar a quantidade, o percentual, e a lista de alunos com auto-avaliações discrepantes (com viés prejudicial ao aluno) das avaliações feitas pelo professor. Uma auto-avaliação é considerada discrepante (com viés prejudicial ao aluno) quando o conceito atribuído pelo aluno é superior (MANA < MPA < MA) ao conceito atribuído pelo professor em pelo menos 25% das metas avaliadas. Considerando esses aspectos, siga as seguintes recomendações:
   1. Desenhe esboços de páginas do sistema relacionadas à essa funcionalidade (cole, logo abaixo, imagens dos esboços)
   2. Ajuste a especificação da feature de auto-avaliação do exercício anterior, contemplando agora o novo interessado e objetivo
   3. Especifique, no formato ilustrado nos slides, um cenário de GUI quando não há nenhuma discrepância (1 aluno tem conceito superior ao do professor em 1 de 5 metas, outro só tem conceitos inferiores ao do professor, e outro só tem conceitos iguais)
   4. Especifique, no formato ilustrado nos slides, um cenário de GUI quando há discrepância em 1 de 3 alunos
   5. Verifique se os seus cenários estão seguindo rigorosamente as recomendações para escrita de cenários com qualidade (mencionadas nos *slides* e no livro texto). Se for necessário, altere as respostas dos itens anteriores.
4. Envie o nome do seu projeto pelo *slack* para que o professor possa criar um canal *slack* para o seu projeto; se cadastre no canal criado. Colar logo abaixo a imagem do canal do *slack*.

## Aula 3

1. Analise as descrições de *features* e cenários do [repositório](https://github.com/spgroup/rgms) do RGMS (veja, no diretório [test/cucumber](https://github.com/spgroup/rgms/tree/master/test/cucumber), os arquivos .feature)
   1. Leia 3 cenários no total, de pelo menos 2 arquivos (cole os cenários logo abaixo)
   2. Identifique, e liste logo abaixo, quais recomendações para escrita de cenários com qualidade (mencionadas nos *slides* e no livro texto) não foram seguidas na escrita desses cenários

*Seguem exemplos de respostas. Substitua esses exemplos pelas suas respostas.*

*O cenário A não segue a recomendação X*

*Os cenários A e B não seguem a recomendação Y*

*Os cenários B e C não seguem a recomendação Z*

*O passo 2 do cenário C deveria ser um When*

1. Solicite, a um colega de turma, dois cenários escritos por ele como resposta dos exercícios 8 e 9.
   1. Cole os cenários logo abaixo.
   2. Identifique, e liste logo abaixo, quais recomendações para escrita de cenários com qualidade (mencionadas nos *slides* e no livro texto) não foram seguidas nos cenários especificados pelo seu colega.

*Seguem exemplos de respostas. Substitua esses exemplos pelas suas respostas.*

*O cenário A não segue a recomendação X*

*Os cenários A e B não seguem a recomendação Y*

*O passo 2 do cenário B deveria ser um When*

1. Realize as entrevistas com os *stakeholders* do sistema escolhido, e cole logo abaixo as principais perguntas feitas por você, e os resumos das respostas dadas pelos *stakeholders.*
   1. Siga as práticas sugeridas nos *slides* e livro, e o *checklist* associado

## Aula 4

1. Defina e coloque a seguir a lista de cenários a serem cobertos (quatro cenários de GUI **relevantes**, não triviais) por você, como parte do projeto da disciplina.
   1. Inicialmente, nessa questão, só é preciso o nome de cada cenário e a ideia geral associada
   2. É recomendado que cada aluno foque em cenários de uma única *feature*, relacionados a uma mesma funcionalidade, assumindo que essa *feature* envolva cenários relevantes que precisem ser especificados e testados. Naturalmente, algumas *features* vão precisar de mais cenários que outras. Assim, se for o caso, foque em mais de uma *feature*.
   3. Cada aluno, individualmente, pede *feedback* ao professor sobre os seus cenários (a melhor forma de fazer isso é colocar os cenários em um Google docs, **liberado para comentários**, e enviar o *link* pelo slack)
   4. Se já existir uma versão funcional do sistema escolhido pela sua equipe, os cenários novos devem corresponder a novas funcionalidades a serem oferecidas pelo sistema, não a funcionalidades já implementadas e funcionando na versão atual do sistema
2. Desenhe e cole abaixo *sketches* dos cenários
   1. Cada aluno, individualmente, pede *feedback* ao professor sobre os *sketches* dos seus cenários
3. Opcionalmente, e só se fizer sentido para o sistema escolhido para o seu projeto, indique (como na Questão 14) dois cenários de serviço **relevantes**, não triviais.

## Aula 5

1. Elaborar e colocar abaixo as especificações dos cenários
   1. Cada aluno, individualmente, pede *feedback* aos monitores e professor sobre os seus cenários
      1. especificações de cenários com erros básicos não serão analisados
      2. cada especificação de cenário só será analisada uma vez, assim capriche na primeira versão
   2. Seguir rigorosamente as práticas sugeridas nos *slides* e livro, e o *checklist* associado
2. Solicite, a um colega de equipe, os cenários escritos por ele como resposta do exercício anterior.
   1. Cole os cenários logo abaixo.
   2. Identifique, e liste logo abaixo, quais recomendações para escrita de cenários com qualidade (mencionadas nos *slides* e no livro texto) não foram seguidas nos cenários especificados pelo seu colega.

*Seguem exemplos de respostas. Substitua esses exemplos pelas suas respostas.*

*O cenário A não segue a recomendação X*

*Os cenários A e B não seguem a recomendação Y*

*O passo 2 do cenário B deveria ser um When*