GABARITO

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Arquitetura e Projeto de Sistemas I

AP 2 1° semestre de 2011.

O Laboratório de Desenvolvimento de Softwares Avançados (LabDESA) é uma instituição que apóia o desenvolvimento de produtos de software inovadores. Para ser admitido no LabDESA um projeto tem que fazer uma proposta de desenvolvimento, usando uma adaptação do processo ágil SCRUM. Você deve desenvolver um sistema que apóie o processo definido a seguir. O sistema para apoiá-lo deve funcionar via Web, utilizando a linguagem de programação Ruby.

O laboratório trabalha com projetos que são liderados por ProductOwners. Cada projeto pode ter vários colaboradores. Um colaborador pode participar de apenas um projeto. Um dos colaboradores é denominado ScrumMaster, todos os outros são TeamMembers.

Cada ProductOwner solicita recursos para seus projetos, que podem ser dos seguintes tipos: posto de desenvolvimento, software, servidores e espaço em disco de servidores. Postos de desenvolvimento, servidores e software são padronizados em uma lista.

Para isso o seguinte fluxo de negócio acontece:

1. O ProductOwner se cadastra no site do laboratório
2. O gerente do laboratório aprova o cadastro
3. O ProductOwner apresenta um projeto
4. O Comitê de Avaliação avalia o projeto
5. Se o projeto for aprovado
    a. O gerente de projeto inclui projeto no laboratório
    b. O ProductOwner indica o ScrumMaster
    c. O ProductOwner inicia o projeto.
    d. O ScrumMaster indica cada TeamMember
    e. Cada TeamMember aceita participar do projeto
    f. Cada TeamMember indica os recursos que precisa na lista de recursos.
    g. O gerente aprova ou reprova os recursos, justificando.
6. Se o projeto não for aprovado
    a. O Comitê prepara uma recomendação de melhoria no projeto
    b. A recomendação é enviada para o ProductOwner
7. O processo se encerra e os dados são passados para o processo de desenvolvimento.

Se um recurso não for aprovado, o TeamMember deve solicitar outro recurso da lista (ou seja, o processo volta ao passo 5f).

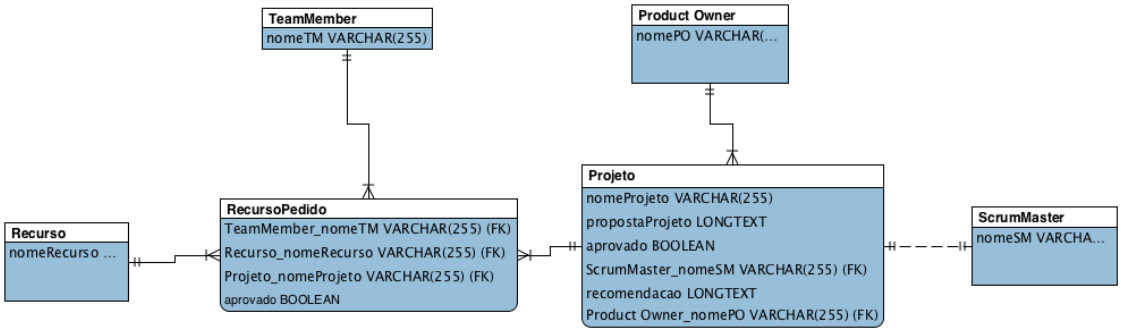
GABARITO

Caso o ProductOwner já esteja cadastrado, deve decidir se é um projeto novo ou uma extensão de um projeto existente. Um ProductOwner pode ter vários projetos. Um Projeto é composto de uma visão (texto de até 3 páginas) e uma lista de histórias do usuário (texto de até 2048 caracteres, com título de até 120 caracteres).

Se necessário, complete as especificações do sistema de acordo com o que achar necessário para o bom funcionamento do mesmo.

1. Faça um modelo de dados para o sistema (2,5 pontos).

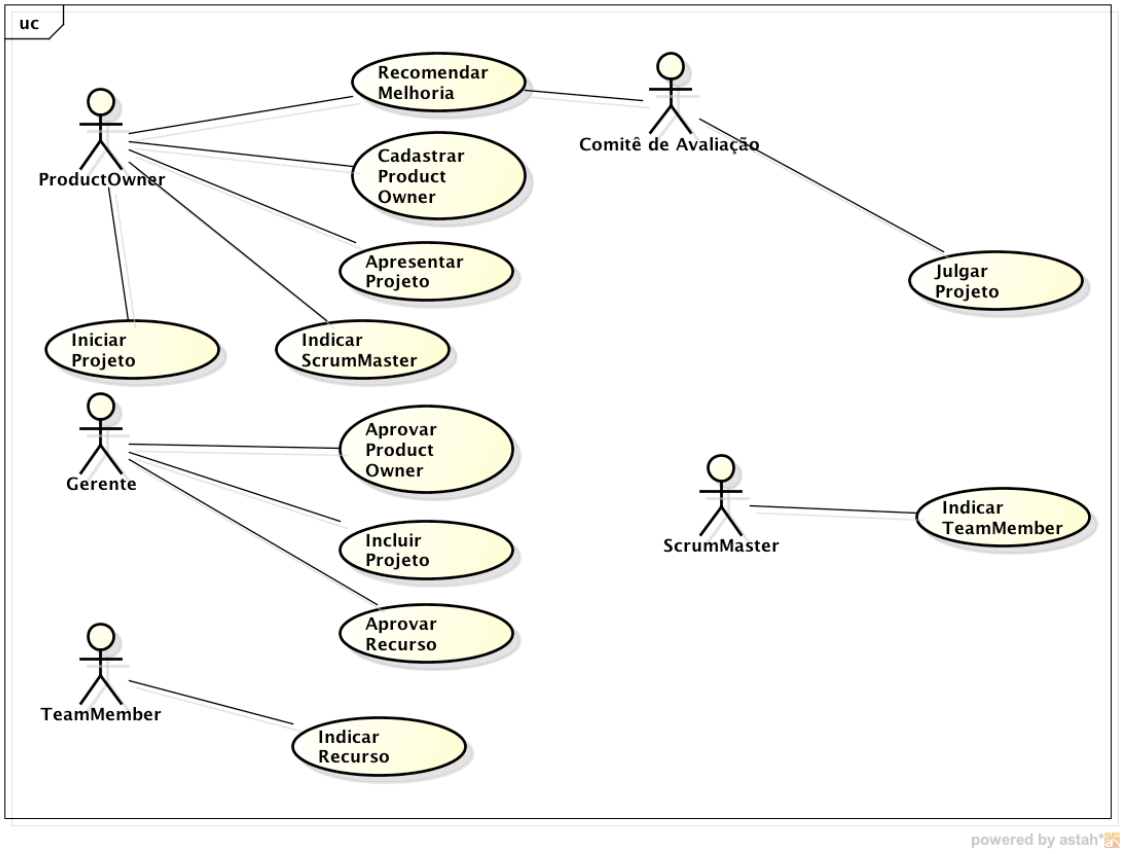
O modelo a seguir foi feito em MySQLWorkbench e apresenta as Chaves Estrangeiras (marcadas com FK). Elas não são necessárias no modelo ER (lógico) e não serão consideradas certas ou erradas no modelo.



2. Faça o Diagrama de Casos de Uso para o sistema (2,0 pontos).

   O aluno tem incluir obrigatoriamente os casos abaixo. Outro caso possível é o CriarRecurso, ou ManterRecursos.

GABARITO



3. Faça um protótipo de baixa fidelidade para o caso de uso onde o gerente aprova os recursos solicitados pelos *TeamMembers*. (1,0 ponto)

O aluno deve desenhar uma tela contendo informações sobre o teammember, talvez alguma informação sobre o projeto, a lista de recursos com um check box ou marca semelhante para aprovar cada um.

GABARITO

4. Faça uma Tabela CRUD relacionando os Casos de Uso apresentados na questão 2 com os dados modelados na questão 1. (1,5 pontos)

A tabela a seguir é aceitável. Ela pode fazer o aluno perceber a necessidade de um caso de uso Criar Recurso, que se incluído na prova estará certo.

	Entidades					
Casos de Uso	ProductOwner	TeamMember	ScrumMaster	Projeto	Recurso	RecursoPedido
Cadastrar Product Owner	C					
Apresentar Projeto	R			C		
Iniciar Projeto				U		
Indicar ScrumMaster			C	R		
JulgarProjeto				RU		
SolicitarRecurso		R		R	R	C
AprovarRecurso		R		R	R	RU
IncluirProjeto				RU		
AprovarProductOwner	RU					
RecomendarMelhoria				RU		
Indicar TeamMember		C				

5. Calcule a quantidade de pontos de função (básico) para o sistema. Apresente a contagem em função das respostas das questões 1 e 2, de preferência na forma de tabelas. (1,5 pontos).

A contagem depende dos casos de uso e dos arquivos que os alunos escolheram.

Pelos casos de uso que eu escolhi, temos 11 funções do tipo entrada, todas simples.

11 x 3 =33

Só temos arquivos internos, são 5 (a tabela de recurso é apenas uma referência).

5 x 5 = 25

total = 58

GABARITO

6. Considerando que cada 45 linhas de código em Ruby equivalem a um ponto de função, calcule o tempo de desenvolvimento do sistema e o esforço necessário segundo o método COCOMO II (1,5 pontos)

58 x 45 = 2160

Usando as equações simplificadas do COCOMO II

$$PM = 2.94 \times MLDC^{0,28}$$

$$TDEV = [3.67 \times (PM_{NS})^{0.32}]$$

EN = 2.94x(2160/1000)^{0,28}=3,647

TDEV=5,55 meses

BOA SORTE!