

O trabalho consiste em duas etapas. Na primeira etapa, a equipe deverá apresentar os documentos necessários para especificação do problema e a sua análise, através da UML. Na segunda etapa, a equipe fará a apresentação do trabalho em sala de aula para o professor e para a turma.

O domínio do problema é uma descrição textual do problema e que deverá ser transformado num documento formal com a sua especificação para que, posteriormente, possa ser feita a análise do problema. Nessa primeira parte, a equipe deverá resolver questões como ambiguidade, omissão, incoerência e contradição que possam constar na descrição do problema. A equipe terá a liberdade de tomar as decisões, desde que sejam justificadas e conforme as necessidades do cliente (o professor poderá fazer o papel de cliente e ser questionado sobre algumas situações que estejam incompletas na descrição do problema). As reuniões para esclarecimento com o cliente (professor) deverão ser agendadas previamente.

Para a especificação do problema, a equipe deverá apresentar um documento com os seguintes itens:

1. Definição do contexto: o que o software deverá fazer é estipulado e os problemas que o produto se propõe a resolver são delimitados, sem entrar em detalhes.
2. Requisitos funcionais: as funcionalidades do software são descritas. É produzida uma lista de todos os requisitos funcionais.
3. Requisitos de interface: é definido como o software interage com as pessoas, com o hardware do sistema, com outros sistemas e com outros produtos. São detalhados os aspectos das interfaces do produto que os usuários consideram como requisitos. Normalmente é feito um esboço das interfaces, levantando através de um protótipo ou de estudos em papel. São também detalhadas as interfaces com outros sistemas e componentes de sistemas.
4. Requisitos não-funcionais: os requisitos de desempenho (velocidade de processamento, tempo de resposta, etc) e outros aspectos considerados necessários para que o produto atinja a qualidade desejada (por exemplo, portabilidade, manutenibilidade, confiabilidade, etc) são descritos. Restrições

impostas pela aplicação, tais como padrões, linguagem de implementação, ambientes operacionais e limites de recursos são também enumeradas. Finalmente, os requisitos são classificados e revistos estabelecendo-se prioridades e verificando-se se a especificação de requisitos está clara e bem entendida por todas as partes interessadas.

5. Método utilizado para extração de requisitos (simulado): deverá ser apresentado o método empregado para a extração de requisitos e uma justificativa pela sua escolha.

Para a análise do problema, a equipe deverá seguir as regras e a notação da UML. Para tanto, a equipe deverá selecionar uma ferramenta apropriada para o desenho dos diagramas. A opção pela ferramenta ficará sob a responsabilidade da própria equipe. Os seguintes diagramas são obrigatórios: diagrama de casos de uso, diagrama de classes, diagrama de estados e diagrama de sequência. Estes diagramas devem ser entregues completos. Os demais diagramas não são obrigatórios, porém, havendo necessidade para resolução do problema, a equipe deverá empregar os outros diagramas. Essa opção por quais diagramas deverão utilizados na análise do problema ficará a cargo da equipe e servirá como critério para avaliação.

#### Domínio do problema:

O professor Ducan ministra aula em mais de uma instituição e tem uma carga pesada de trabalho. Isso é um fato comum entre os docentes, que são obrigados a se desdobrar na sua nobre função, mas pouco valorizada, de magistério. Apesar da grande carga de trabalho, o professor Ducan se esforça em oferecer o melhor que pode.

Nos últimos semestres, o professor tem ministrado as mesmas disciplinas. O docente chega a ministrar a mesma disciplina para várias turmas no mesmo semestre. Assim, ao elaborar as provas, o professor se esforça em ser criativo na produção das questões de forma a evitar que existam provas com questões iguais. Inclusive, com questões de anos anteriores, pois muitos alunos conseguem as cópias das provas de turmas passadas, ou mesmo pelo fato de terem que repetir a disciplina.

Assim, o professor pensou em desenvolver um software que o ajudasse na elaboração das suas provas. Basicamente, o programa deveria cadastrar um grande número de questões e, a partir dessas questões, as provas seriam geradas. Com isso, o professor precisaria apenas pensar e cadastrar as questões e o próprio programa faria a geração das provas. Isso ajudaria bastante o professor nas suas atividades do dia-a-dia.

Normalmente, existem dois tipos de questões: objetivas e subjetivas. As questões objetivas são formadas por um enunciado e por cinco alternativas, sendo que apenas uma delas é a correta. Assim, o professor precisará cadastrar o enunciado, as cinco alternativas e o item que corresponde à resposta correta.

Já nas questões subjetivas, o professor deverá cadastrar o enunciado da questão e uma possível resposta. Como nesse tipo de questão é possível encontrar mais de uma resposta, o professor poderá, ao cadastrar a resposta, informar as possíveis soluções.

Cada questão deverá possuir um valor e o seu grau de dificuldade (fácil, médio, difícil). O valor da questão deverá ser entre 0,1 e 10,0, pois as provas podem ter nota no máximo igual a 10 pontos e o valor mínimo da nota, para quem não errou todas as questões, é 0,1 ponto. O grau de dificuldade é também importante porque o professor poderá escolher, por exemplo, que um determinado percentual da prova seja formado por questões fáceis e o restante de questões de nível médio. Esse balanceamento entre o grau de dificuldade e das questões deverá ser realizado pelo software, mas definido pelo professor.

Para o cadastrado de algumas questões, é possível o professor inserir uma imagem (figura). Portanto, o professor deverá selecionar uma imagem, que tenha sido criada em outro programa, e adicioná-la num ponto específico da questão. O professor poderá escolher várias imagens para uma mesma prova e essas imagens também deverão ser gravadas na base de dados. O programa deverá ser capaz de reconhecer um conjunto específico de formatos de imagens.

As questões poderão ser cadastradas por mais de um professor. Assim, é importante ter informação de quais professores poderão usar o programa, inclusive com o controle de acesso (login/senha). Para tanto, é importante cadastrar algumas informações pessoais dos professores, como nome, endereço, e-mail, telefone, instituição de ensino em que trabalha (poderá ser mais de uma), data de nascimento, formação, entre outras informações relevantes.

Como forma de organização do banco de questões, ao cadastrar uma questão, o professor também deverá informar a disciplina (exemplo: Linguagem de Programação, Estrutura de Dados, Banco de Dados ou Redes de Computadores) e o assunto (por exemplo, na disciplina de Estrutura de Dados, teríamos os assuntos de árvores, pilhas ou listas).

O programa deverá gerar a prova em dois formatos: pdf e odt. O primeiro formato é um padrão e pode ser utilizado em qualquer sistema operacional e com diversos programas. O segundo formato é necessário para que o professor tenha a possibilidade de alterar a prova, uma vez que o formato pdf não permite. Em toda prova deverá constar um cabeçalho com os seguintes campos: nome da disciplina, data da prova, nome do professor, nota, nome do aluno, turma, matrícula do aluno e um espaço para a logomarca da instituição de ensino, uma vez que a mesma base de dados poderá servir para disciplinas em diferentes faculdades. Ao final do enunciado de cada questão deverá constar o valor para que os alunos saibam quanto vale cada uma. A pontuação deverá aparecer da seguinte forma: [valor 1,0]. Lembrado, é claro, que a somatória dos pontos da prova deverá ser igual 10 pontos e que o valor da questão deve corresponder ao que foi gravado no banco de dados.

O sistema deverá gerar também um gabarito para o professor como uma maneira de ajudá-lo nas correções das provas. O gabarito deverá seguir o mesmo padrão das provas, com a diferença que nele aparecerão as respostas que foram previamente armazenadas. Assim, nas questões objetivas, basta marcar o item correto com um X. Já nas questões subjetivas a resposta deverá aparecer ao final de cada enunciado.

Uma extensão para o programa será gerar algumas provas como simulados para que os alunos possam resolver pela web. Neste caso, serão consideradas apenas as questões de múltipla escolha, pois essas questões podem ser resolvidas automaticamente, sem a necessidade de intervenção do professor. Ao cadastrar a questão, o professor poderá marcá-la como pública ou privada. Se a questão for pública, ela poderá aparecer no simulado. Se a questão for privada, não deverá aparecer como parte de uma prova simulada. As questões que são públicas também poderão compor as provas.

Para a realização dos simulados, o aluno poderá fazer um pré-cadastro na internet com as suas informações pessoais (nome, telefone, endereço, e-mail, instituição de ensino

que estuda, curso, período, ano, nome da disciplina, nome do professor da disciplina, entre outras informações). Depois desse pré-cadastro, o professor receberá um e-mail com a solicitação do aluno. Se quiser, poderá permitir que o aluno faça os simulados. Neste caso, o professor poderá determinar até quando o aluno terá acesso ao sistema. Caso não permita, o professor poderá negar e enviar uma mensagem informando a razão da sua decisão.

Além da descrição do problema, o professor da disciplina atuará como cliente do sistema. Assim, a equipe terá, em horários previamente agendados, reuniões com o professor para que o mesmo possa detalhar mais sobre as suas necessidades e para que os alunos possam sanar possíveis dúvidas que tenham surgido da leitura do domínio do problema. É responsabilidade da equipe agendar os horários de reunião com professor (“cliente”).

#### Bônus:

A equipe poderá entregar uma versão implementada do sistema com as suas principais funcionalidades. A linguagem de programação para essa atividade deverá ser a linguagem Java. De acordo com a qualidade da implementação e com o grau de completude, a equipe poderá ter a sua nota acrescida de até 2,0 pontos (pontuação extra). Essa atividade não é obrigatória e tem o objetivo apenas de servir como um bônus para o trabalho e como forma de incentivar a implementação do problema usando uma linguagem de programação orientada a objetos. Caso a equipe decida por entregar o programa, o código e a documentação, principalmente em relação à instalação do programa, deverá ser feita na mesma data em que for entregue o trabalho escrito.

#### Regras:

- As equipes deverão ser formadas por, no máximo, 3 (três) componentes.
- A nota será igual a todos os membros da equipe.
- A entrega da documentação valerá 8 pontos e a apresentação do trabalho valerá 2 pontos.
- Para cada dia de atraso na entrega do trabalho escrito a equipe perderá um ponto na nota.

ESPECIFICAÇÃO DO TRABALHO DA DISCIPLINA DE ANÁLISE  
DE SISTEMAS ORIENTADOS A OBJETOS DO CURSO DE  
TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE



- A documentação do trabalho escrito, que corresponde à primeira etapa do trabalho, deverá ser entregue impressa e encadernada.
- A apresentação deverá ser feita por apenas um dos membros da equipe.
- A equipe terá vinte minutos para apresentar o trabalho.
- No caso de ausência de um dos membros da equipe, a nota da equipe será reduzida de 0,5 ponto, por cada membro ausente.
- Data de entrega do trabalho escrito: 02.06.2009.
- Data da apresentação: 09.06.2009.