

**APLICANDO *PROJECT-BASED LEARNING* NO ESTUDO
INTEGRADO DE ENGENHARIA DE SOFTWARE, ANÁLISE E
PROJETO DE SISTEMAS E BANCO DE DADOS**

David Moises B. Santos – davidmbs@uefs.br

Hugo Saba – hugosaba@uefs.br

João Rocha Junior – joao@uefs.br

Victor Sarinho – vsarinho@gmail.com

Universidade Estadual de Feira de Santana (UEFS) – Departamento de Ciências Exatas Km 03, BR-116 Campus Universitário – 44.031-460 – Feira de Santana – BA – Brasil

Resumo: *A Engenharia de Software abordada como um estudo integrado adotando o Project-Based Learning (PBL), possibilita uma experiência profissional, permitindo ao estudante vivenciar situações reais de engenharia de software. A automação de processos na busca por competitividade e a evolução tecnológica, deram uma posição de destaque a engenharia de software. Como uma disciplina extensa a Engenharia de Software agrega diversas áreas do conhecimento (i.e. engenharia de requisitos, qualidade de software, entre outras). O ensino de tais disciplinas de forma isolada, levando em conta o seu caráter prático e suas limitações de carga horária durante um semestre, torna-se uma tarefa difícil, custosa e de qualidade questionável. Este artigo vem disseminar uma experiência no ensino de Engenharia de Software através de um estudo integrado de Engenharia de Software, Análise e Projeto de Sistemas e Banco de Dados, adotando o PBL (Project-Based Learning) como metodologia prática de ensino.*

Palavras-chave: *Engenharia de Software, Metodologia PBL, Disciplinas Integradas*

1 INTRODUÇÃO

A automação de processos na busca por competitividade e a evolução tecnológica deram à engenharia de software uma posição de destaque nas linhas de pesquisa e de trabalho no contexto das ciências computacionais. Sua importância a transformou num componente curricular que abrange diversas áreas do conhecimento (i.e. engenharia de requisitos, gerência de projetos, qualidade de software, entre outras). O volume de informações tratado neste componente curricular é diversificado, sendo distribuído em diversas disciplinas que são

lecionadas ao longo do currículo dos cursos de computação, por exemplo: programação, análise de sistemas, engenharia de software e banco de dados.

Entretanto, a vivência de tais disciplinas de forma isolada dificulta a elaboração de uma visão sistêmica do processo de desenvolvimento pelos estudantes [Alves & Benitti, 2006]. Além disso, o caráter prático dos componentes curriculares relacionados à engenharia de software e a baixa carga horária (i.e. entre 45 e 90 horas) atribuída às disciplinas para integrar teoria e prática durante o semestre, torna mais difícil e custosa a elaboração de softwares bem elaborados.

Outro aspecto importante envolvido no ensino de engenharia de software e que não costuma ser abordado durante esta disciplina é a relação inter-pessoal existente ao longo de um processo de software, onde são abordadas atividades de trabalho em grupo como composição de equipe, delegação de atividades, colaboração de atividades, comunicação de grupo, entre outras.

Este artigo vem disseminar a experiência no ensino de Engenharia de Software através de um estudo integrado das disciplinas de Engenharia de Software, Análise e Projeto de Sistemas e Banco de Dados. Este Estudo Integrado, que foca tanto o aspecto técnico quanto pessoal, forma um único componente curricular que é oferecido em um único semestre, com carga horária de 180 horas. Além disso, considerando o caráter prático do componente, é adotada uma metodologia de ensino prática centrada no estudante, denominada PBL (*Project-Based Learning*) [Thomas, 2000][Boud & Feletti, 1997][Duch *et al*, 2001]], onde o aprendizado é motivado através de projetos. Desta forma, além de apresentarmos a abordagem de ensino utilizada, também apontamos os benefícios e dificuldades encontradas, bem como possíveis soluções para melhorar o desempenho dos alunos dentro desta abordagem de ensino. Vale ressaltar que embora os métodos *Project-Based Learning* e *Problem-Based Learning* sejam similares, o primeiro é mais adotado na área de computação, enquanto o último prevalece na área médica [Ryan & Koschmann, 1994].

Na aprendizagem baseada em projetos, cria-se um ambiente de aprendizagem no qual o projeto guia o aprendizado dos alunos. Assim, antes dos estudantes aprenderem um determinado conteúdo, eles recebem um desafio em forma de projeto. Este projeto é colocado de modo que os estudantes descubram que precisam buscar novos conhecimentos antes de poder solucioná-lo. Colocar o problema antes do aprendizado motiva os estudantes e o projeto torna-se a força motriz que motiva o aprendizado.

Detalhes do aprendizado baseado em projetos são apresentados na Seção 2. Já a estrutura elaborada para o ensino do estudo integrado é tratada na Seção 3, enquanto que uma discussão em torno dos resultados alcançados é realizada na Seção 4. Trabalhos correlatos são descritos na Seção 5. E na Seção 6, são traçadas as conclusões e perspectivas.

2 APRENDIZADO BASEADO EM PROJETOS

Para entender como funciona a metodologia empregada na disciplina, é necessário compreender o funcionamento do PBL, abordagem didática-pedagógica utilizada. Segundo Thomas (2000), há cinco pontos principais que caracterizam o aprendizado baseado em projetos:

- **Centralidade:** o projeto é central no curso, e não periférico. Neste sentido, o projeto é a estratégia central de ensino, e os estudantes aprendem os principais conceitos da disciplina via projeto;
- **Orientado a questões:** o projeto contém problemas que guiam os estudantes ao encontro dos conceitos principais da disciplina;
- **Investigação construtiva:** uma pesquisa é um processo orientado a objetivos que envolvem pesquisa, construção do novo conhecimento e solução. Em outras palavras, é um processo evolutivo. Se o projeto não oferece dificuldades para o estudante, ou ele pode resolvê-lo com o conhecimento já adquirido até então, o mesmo não pode ser considerado como projeto; neste caso, tem mais característica de um simples exercício;
- **Autonomia:** PBL envolve autonomia no sentido de que estudantes devem ficar livres para resolver o problema, tomando decisões e dedicando tempo não supervisionado ao mesmo;
- **Realismo:** projetos devem se assemelhar ao máximo a problemas do mundo real, possuindo características que dão aos alunos um sentimento de autenticidade.

Com este conjunto de características, é possível criar um ambiente de aprendizagem no qual o projeto motiva e guia o aprendizado dos alunos. O projeto é dividido em etapas para que os estudantes possam se concentrar em alguns tópicos. Além disso, cada etapa é posta para a turma antes do aprendizado para que os estudantes se sintam motivados a aprender sobre os tópicos relacionados e possam concluí-la.

Do ponto de vista da alocação de tempo para a aprendizagem, buscou-se um híbrido de Aulas Expositivas (AE) e Grupos Tutoriais (GT). As AEs seguem o modelo tradicional de aulas dialogadas com o objetivo de introduzir assuntos que estão sendo tratados no projeto ou até mesmo discutir outros assuntos que não foram abordados no mesmo. Além disso, o professor pode atuar como consultor, dirimindo dúvidas fomentadas por discussões relativas ao projeto, durante os GTs.

Os Grupos Tutoriais são compostos por grupos de, no máximo, 10 alunos, sendo cada grupo, tutelado por um único professor [Woods, 1996]. Nestes encontros, os alunos discutem o projeto, levantando e elegendo idéias e questionamentos. Ao término da discussão, eles devem estabelecer metas que deverão ser alcançadas até a próxima reunião. No GT, o professor atua como tutor, não expondo sua visão acerca do assunto, apenas guiando os aprendizes na busca de soluções. Desta forma, os alunos chegam ao conhecimento através do processo de auto-reflexão e das relações com outros alunos do grupo. Maiores detalhes sobre a forma de conduzir discussões durante GTs podem ser encontrados em [Deslile, 1997].

2.1. A metodologia PBL dentro da sala de aula

Na metodologia PBL os papéis do aluno e tutor (i.e. professor) são bastante diferentes da abordagem tradicional. No PBL os professores atuam como direcionadores (técnicos) ajudando os alunos a identificarem um caminho para alcançar o aprendizado necessário para solucionar o projeto. O tutor pode seguir um conjunto de passos para conduzir a seção tutorial dentro da sala de aula (Figura 1).

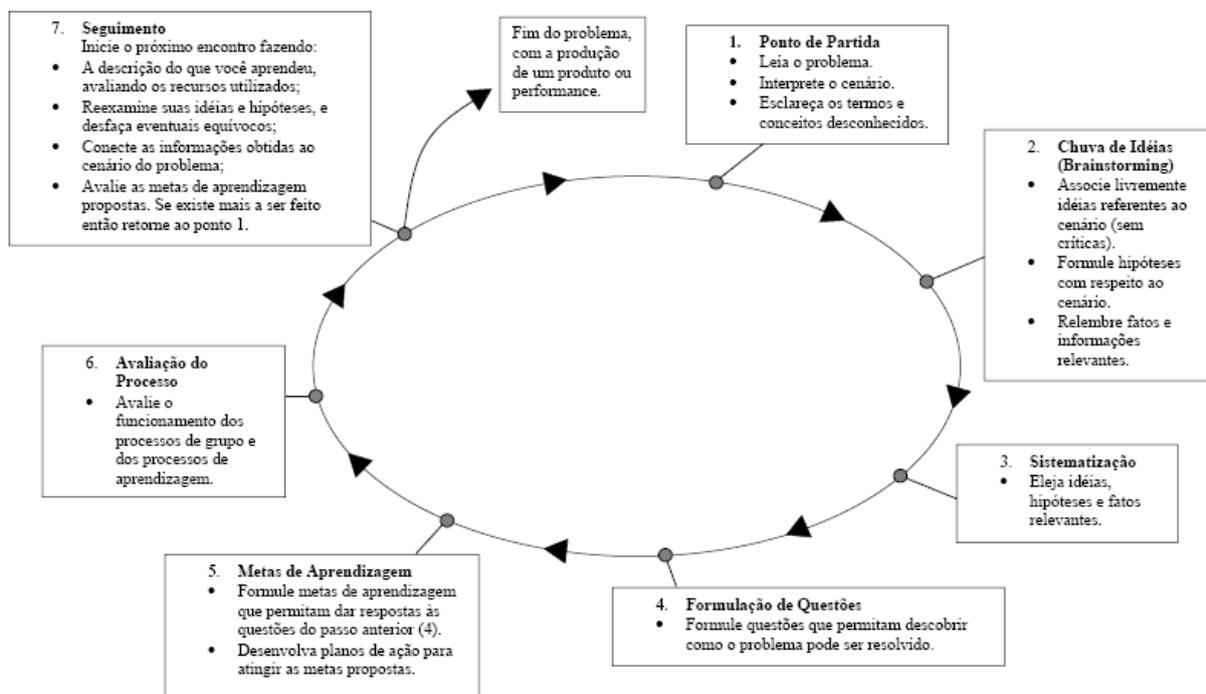


Figura 1: Passos da metodologia PBL dentro da sala de aula

O papel do tutor dentro da sala de aula deve iniciar com a distribuição do problema (primeira seção) e com uma breve motivação a respeito do mesmo, ressaltando a sua importância na formação dos alunos. Após isto, o tutor deve delegar um tempo (15 minutos em média) para a leitura do problema, posteriormente os alunos devem interrogar o tutor sobre termos e conceitos desconhecidos. Quando um termo ou conceito é essencial para a compreensão do problema o tutor pode responder imediatamente o que significa, caso seja algo que os alunos precisam se aprofundar o tutor pode sugerir a adição desse termo nas questões que os alunos deverão resolver para a próxima seção tutorial.

Após a interpretação, o tutor abre espaço para o *brainstorming*, neste momento todos os alunos devem tentar contribuir com idéias, fatos e hipóteses que auxiliem na resolução do problema. Nesta etapa, o tutor deve interferir o mínimo possível deixando que os alunos discutam livremente sobre tema durante cerca de trinta minutos. Para auxiliar na discussão o tutor poderá fazer questionamentos que conduzam os alunos ao aprendizado, ou interferir na discussão caso ela esteja sendo direcionada para um caminho diferente do contexto do problema, sendo assim o tutor deverá esclarecer mais uma vez o problema e reiniciar a discussão.

Após o *brainstorming*, durante quinze minutos os alunos devem selecionar as idéias, hipóteses e fatos mais pertinentes e importantes para a solução do problema. Uma vez concluída a seleção os alunos devem formular questões cujas soluções permitam que eles avancem em direção a solução do problema, esta etapa deve levar no máximo quinze minutos.

Depois de ter identificado as principais questões que auxiliam na resolução do problema, ou projeto, os alunos devem definir as metas a serem alcançadas até o próximo tutorial, (atividade que dura em média quinze minutos) neste caso o aluno deve ter consciência do

tempo disponível para se dedicar a esta atividade e saber realmente o que poderá ser feito até a próximo tutorial.

Ao final desta etapa, o tutor juntamente com os alunos deve fazer uma avaliação sobre a seção tutorial, indicando os pontos positivos e negativos do aprendizado em grupo. O tutor também poderá avaliar o desempenho individual de cada um e indicar, quando pertinente, o que poderá ser melhorado, todo o processo de avaliação não excede dez minutos.

As demais seções de tutorial devem iniciar com o que os alunos aprenderam desde a última seção, portanto eles deverão avaliar as idéias e discutir os questionamentos levantados na última seção tutorial, corrigir eventuais equívocos e avaliar as metas propostas. Este processo consome em média trinta minutos. Caso exista mais algo a ser feito, deve se executar todo o ciclo novamente, neste caso, o tutor deverá diminuir o tempo em cada uma das etapas para a seção tutorial não ficar muito cansativa, tanto a primeira seção tutorial quanto as demais devem durar em média cem minutos (i.e. uma hora e quarenta minutos).

3 ESTUDO DE CASO

Este componente curricular é ministrado no curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS), período vespertino, na cidade de Feira de Santana - BA. É importante ressaltar que o modelo proposto deste estudo integrado temático é aplicado ao longo de todo o currículo do curso, tendo como finalidade fazer com que componentes curriculares de um mesmo período letivo possam ser trabalhadas em conjunto, compartilhando trabalhos, desafios e oportunidades de aprendizado [Bittencourt & Figueiredo, 2003]. Neste intuito, é reagrupado o corpo de conhecimentos ao longo dos anos de aprendizado, de modo que disciplinas que já possuem elos normalmente estejam agrupadas num mesmo período letivo. Assim, pode-se, por exemplo, agrupar outras disciplinas como linguagens formais e autômatos com a disciplina de compiladores, já que as linguagens formais dão subsídios para a construção dos compiladores [Aho *et al*, 1995].

Este Estudo Integrado de Engenharia de Software já foi aplicado em quatro semestres distintos e sua carga horária foi planejada para abranger um total de 180 horas sendo dividido em 3 módulos, contendo 60 horas de duração cada um: Engenharia de Software, Análise e Projeto de Sistemas e Bancos de Dados. Para cada módulo, há um professor responsável. São estes mesmos professores que, geralmente, atuam como tutores dos GTs. Ao longo desse período, alguns aspectos do estudo integrado foram aperfeiçoados. Assim, nesta seção será abordada a última edição deste componente curricular.

3.1. Objetivo do estudo de integrado de engenharia de software

Capacitar o estudante para a engenharia de software, tratando as distintas atividades relacionadas a essa prática: definição do problema, desenvolvimento dos requisitos, planejamento da construção, arquitetura do software, projeto de banco de dados, projeto de software detalhado, codificação e depuração, testes unitários, integração, testes de integração, testes de sistemas e manutenções [Pressman, 2002][Sommerville, 2003][McConnell, 2005][Silberschatz *et al*, 1999].

3.2. Conteúdo programático

O conteúdo programático está disposto como descrito a seguir. O módulo de Análise e Projeto de Sistemas envolve [Sommerville, 2003]: Técnicas de análise de sistemas; Levantamento de informações; Estudo de viabilidade; Projeto de interfaces; Mecanismos de coleta e disseminação de informações; Sistemas de informação gerenciais; Especificação lógica e física de sistemas; Técnicas de documentação do projeto e preparação de manuais; Ferramentas de desenvolvimento de sistemas.

Já o módulo de Engenharia de Software abrange [Pressman, 2002][Sommerville, 2003][McConnell, 2005]: A engenharia de software; A produção de software; Qualidade de software; Ciclo de vida do software; Especificação de requisitos; Projeto de software; Implementação, teste e verificação de software; Padrões de desenvolvimento e documentação de software; Alocação e administração de pessoal e recursos; Estudo e uso de metodologias, ambientes e ferramentas; Programação em grupo.

Por fim, o módulo de Banco de Dados aborda [Silberschatz *et al*, 1999][Date, 2000]: Introdução a sistemas de bancos de dados; Projeto de bancos de dados: modelo de entidades e relacionamentos e suas extensões; modelo de dados relacional; algoritmos de decomposição e formas normais; Controle de restrições de integridade; Noções básicas de transações; Linguagens de consultas: cálculo relacional, álgebra relacional e SQL; visões e índices secundários.

3.3. Projeto

O projeto do Estudo Integrado foca o desenvolvimento de um software direcionado para um cliente real, dividido em três etapas básicas: (1) análise de requisitos, (2) projeto arquitetural e (3) implementação e testes. O processo de desenvolvimento adotado segue o modelo espiral, usando simplificações dos artefatos do RUP (*Rational Unified Process*). Foi feita uma parceria com a Assessoria de Informática da UEFS, responsável pela política de informatização da Universidade, no intuito de levantar a demanda de aplicativos nos diversos setores da mesma. Após uma avaliação, foi eleito como projeto da disciplina um sistema escolar para a Escola de ensino básico da UEFS. Entre os principais critérios adotados na seleção pode-se destacar a disponibilidade do cliente e o escopo da aplicação.

A Figura 2 ilustra a forma como o projeto foi descrito, na primeira etapa. A descrição conteve: um título; o tema correspondente à etapa; os objetivos de aprendizagem; cronograma de aulas e grupos tutoriais; a descrição da etapa do projeto; o produto que deveria ser entregue pelo grupo; e recursos de aprendizagem.

Na primeira etapa, que teve como foco a análise de requisitos, o produto foi um documento de análise de viabilidade e de requisitos do sistema que contemplava artefatos como: documento de visão, análise de risco, análise de custos, cronograma de atividades e papéis de cada membro na equipe.

Título: Sistema Escolar
Tema: Análise e definição de requisitos
Objetivos de aprendizagem: Conhecer os fundamentos da Engenharia de Software; Compreender como se dá a comunicação entre os membros da equipe; Realizar tarefas em conjunto de forma mais eficiente; ...

Cronograma			
Semana	Data	Grupo Tutorial	Aula
	22/02	Projeto: Início da discussão da primeira etapa	-
1	22/02	-	Apresentação da Disciplina
	23/02	-	Visão Geral de Engenharia de Software

...			

Descrição: Você faz parte de uma equipe de desenvolvimento que foi contratada para confeccionar um sistema escolar para a Escola da UEFS (pré-escolar a 8ª série). Primeiramente, vocês devem fazer um estudo de viabilidade. Em seguida, sua equipe deverá iniciar as atividades de análise e definição de requisitos. É importante especificar o tempo e custo necessário para implantar esse sistema dentro da escola...

Produto: Haverá três produtos a serem entregues. O primeiro é um relatório do estudo de viabilidade de desenvolvimento do software. A data final será dia 27/09. O segundo produto deverá ser uma proposta escrita para o desenvolvimento do sistema escolar, a qual deve ser elaborada pela equipe do grupo tutorial e entregue até o dia 20/10. Nesta proposta deverá conter, pelo menos, os seguintes artefatos: Documento de Visão, Análise de Custo, Análise de Risco,... Por fim, o último produto é uma apresentação desta proposta deverá ser realizada no dia 24/10.

Recursos para aprendizagem
 SOMMERVILLE, IAN. "Engenharia de Software", 6ª Edição, Addison Wesley, 2003.
 PFLEEGER, SHARI L.. "Engenharia de Software: teoria e prática", Pearson, 2004.
 PRESSMAN, ROGER S.. "Engenharia de Software", 5ª Edição, Pearson, 1999.

Figura 2: Primeira etapa do projeto

Na segunda etapa, foi solicitado o documento da arquitetura do software contendo diagramas UML (casos de uso, classe, interação, implantação, componentes e comportamento), entidade-relacionamento e relacional. Para tanto, foi preciso usar os artefatos gerados na etapa anterior, principalmente, os requisitos. Neste sentido, também foi solicitado o documento de requisitos atualizado, pois, com a discussão do projeto e novas reuniões com o cliente, alguns requisitos, em geral, foram alterados.

Por fim, na etapa de implementação e testes, além dos documentos de requisitos e arquitetura atualizados, também foi requisitado o software implementado, juntamente com o plano de testes, o qual foi usado na validação do próprio sistema.

3.4. Adequação do Estudo Integrado ao aprendizado baseado em projetos

A proposta apresentada neste trabalho atende aos critérios de uma abordagem de aprendizado baseado em projetos, descritos na Seção 2: (1) centralidade, (2) orientado à questões, (3) investigação construtiva, (4) autonomia, e (5) realismo. Quanto ao primeiro critério, as aulas expositivas foram planejadas de acordo as etapas do projeto. Por exemplo, aulas de banco de dados foram ministradas após o início da etapa relacionada (segunda). Já o aprendizado dos conceitos do Estudo Integrado, por parte dos alunos, ocorreu à medida que eles evoluíram no desenvolvimento do projeto, através do acúmulo de experiência. Assim, o projeto foi o elemento central da disciplina.

As questões que guiaram o aprendizado (segundo critério) foram postas, principalmente, através dos artefatos requisitados em cada etapa (produto) e entrevistas com o cliente. Por exemplo, na etapa de análise, foram solicitados documento de visão, cronograma de atividades, papéis de cada membro na equipe, etc. Com o cliente, os alunos se depararam com

questões como: O que é considerado um requisito (não) funcional? Um determinado requisito fará parte do escopo do software?

Neste sentido, cada etapa do projeto envolveu tarefas não-triviais, não configurando assim, um simples exercício. Foi preciso que o estudante realizasse de fato um processo de investigação construtiva (terceiro critério) em busca da solução adequada. E ao longo deste percurso, os estudantes tiveram que mostrar autonomia (quarto critério), pois muitas decisões foram tomadas, por exemplo: escolha das tecnologias, papéis dos membros da equipe, decisões de projeto, etc. Isto ficou evidente após a apresentação das soluções por cada um dos grupos, que foram diferentes, apesar de se tratar do mesmo projeto.

Considerando o último critério, o desenvolvimento de um software por si só já se assemelha com problemas do mundo real. Todavia, no intuito de aumentar esse realismo, na última edição da disciplina, através de uma parceira com a Assessoria de Informática da UEFS, como já foi citada, os alunos desenvolveram um software real, para uma escola interna da universidade. Assim, em vez do professor atuar como cliente fictício, como nas edições anteriores, utilizou-se um cliente real.

3.5. Sistema de Avaliação do Estudo Integrado

Com relação a avaliação, a mesma tem duas dimensões: a dimensão global, onde se avalia as competências gerais e específicas adquiridas pelo estudante, e a dimensão modular, onde se avalia a proficiência de cada estudante no conhecimento dos assuntos específicos de cada módulo do estudo integrado (Engenharia de Software, Análise e Projeto de Sistemas e Bancos de Dados).

Avaliação do Estudo Integrado

Para o estudo integrado, a nota final é calculada a partir de uma média ponderada (MPEI) de produtos de grupos tutoriais (PGT) e do desempenho nos grupos tutoriais (DGT), dada a seguir:

$$\text{MPEI} = 0,7 \text{ PGT} + 0,3 \text{ DGT}.$$

A média PGT foi calculada a partir de uma ponderação das notas obtidas nos produtos. Durante os Grupos Tutoriais, os alunos foram avaliados por seu desempenho (DGT), sendo consideradas as dimensões de participação, contribuição efetiva, criatividade, entrosamento e respeito mútuo em cada um dos passos do ciclo PBL [Deslile, 1997].

A média ponderada de cada unidade é usada para a avaliação integrada dos alunos, correspondente a um dos requisitos para a aprovação no estudo integrado. As datas para entrega dos produtos da resolução dos problemas são definidas ao longo do curso.

Avaliação dos Módulos do Estudo Integrado

A nota final de cada módulo é calculada a partir de uma média ponderada (MPM) de produtos de grupos tutoriais referentes ao módulo (PGTM) e avaliações de conteúdo do módulo (AC), dada abaixo:

1. Engenharia de Software: $\text{MPM} = 0,7 \text{ PGTM} + 0,3 \text{ AC}$
2. Análise e Projeto de Sistemas: $\text{MPM} = 0,7 \text{ PGTM} + 0,3 \text{ AC}$

3. Banco de Dados: $MPM = 0,7 PGTM + 0,3 AC$

A média PGTM é calculada a partir de uma ponderação das notas obtidas nos produtos relativos ao módulo em questão. Cada problema (etapa) que agregue conteúdo do módulo tem uma avaliação específica dos aspectos relativos a este conteúdo e esta nota foi ponderada na média PGTM de acordo com a participação do conteúdo do problema no total do conteúdo do módulo. Por exemplo, para o módulo de Análise e Projeto de Sistemas, os dois primeiros produtos devem ter um peso maior do que o último, que tem foco na codificação e testes.

As provas tratam dos conteúdos abordados no módulo e são provas escritas e individuais. As datas para entrega dos produtos da resolução dos problemas e provas são definidas ao longo do curso.

Provas Finais

Não obtendo conceito suficiente na avaliação de competências do estudo integrado, o estudante tem a possibilidade de fazer prova final do estudo integrado, e o conceito final fica de acordo com o sistema de avaliação vigente na UEFS.

Não obtendo nota igual ou superior a 7,0 na avaliação da proficiência no módulo, o estudante pode fazer prova final do módulo. O conceito final do módulo é uma média ponderada do conceito anterior (peso 6) com a prova final (peso 4). O conceito final do módulo deve ser igual ou superior a 5,0 para aprovação no módulo.

Aprovação no Estudo Integrado

Para ser aprovado no estudo integrado, o estudante precisa cumprir os seguintes requisitos:

1. Ter frequência igual ou superior a 75% da carga horária efetiva ministrada no estudo integrado, caso contrário haverá reprovação por frequência e o estudo integrado completo deve ser cursado novamente;
2. Ser aprovado na avaliação integrada das competências para o estudo integrado, caso contrário haverá reprovação por conceito e o estudo integrado completo deverá ser cursado novamente.

Caso os requisitos anteriores tenham sido cumpridos, o estudante deverá ainda ser aprovado na avaliação de proficiência de cada módulo componente do estudo integrado, caso contrário sua situação ficará pendente até que seja aprovado em novas avaliações do módulo em período(s) subsequente(s), não necessitando, entretanto, frequentar novamente as atividades regulares do estudo integrado, bastando realizar apenas as atividades de avaliação.

4 DISCUSSÃO

Nesta seção serão discutidos os resultados alcançados através do exercício de habilidades técnicas e não-técnicas, por parte dos alunos, bem como questões relativas a experiência adquirida pelos professores com a aplicação do Estudo Integrado.

Quanto a integração proposta entre os três componentes curriculares, pode-se afirmar que os alunos conseguiram, através do projeto, relacionar facilmente aspectos de Análise e Projeto de Sistemas, Engenharia de Software e Banco de Dados, visualizando o componente curricular como um todo. A exploração do conteúdo na medida em que o desenvolvimento do projeto evoluía, favoreceu tal integração.

A divisão em etapas facilitou o direcionamento dos alunos, principalmente, devido ao extenso conteúdo programático. Mesmo assim, em alguns tópicos abordados, os alunos tiveram dificuldade em identificar o que era realmente necessário aprender para alcançar uma solução específica. Por exemplo, em uma das etapas, os estudantes, em um primeiro momento, não conseguiram definir claramente quais as metas de qualidade que deveriam ser aplicadas ao software em construção. Nestes casos, houve interferência do tutor.

Merece destaque o desenvolvimento de um software real, que segundo os próprios alunos, foi um fator motivante na disciplina. O software foi dividido pelas equipes em três camadas: dados, negócio e interface. Todas as equipes conseguiram codificar as camadas de negócio e de dados. A interface Web, requerida pelo cliente, foi implementada parcialmente, devido a inexperiência prevista de alguns alunos. Apesar disto, todo o código desenvolvido foi entregue a Assessoria de Informática, que ficou responsável por continuar o projeto, conforme acordado antes, caso o mesmo não fosse finalizado.

Um ponto que também merece importância é o trabalho em equipe, aspecto de suma importância para o desenvolvimento de softwares. No Estudo Integrado, com a adoção de PBL, os alunos já são estimulados a desempenhar trabalho em grupo, colaborando na concepção do novo conhecimento, respeitando opiniões divergentes e aperfeiçoando a expressão oral.

Levando em conta aspectos mais técnicos da Engenharia de Software, para cada grupo, inicialmente, foi eleito um membro para atuar como gerente do projeto. À medida que as atividades eram identificadas, as mesmas eram divididas entre os membros do grupo tutorial, que se sentiam parte de uma equipe. Apesar da existência do gerente de projeto, a delegação de atividades era decidida em conjunto, assim como outras decisões do grupo. Para facilitar a comunicação e colaboração do grupo foram usados recursos tecnológicos importantes como ferramentas para controle de versões e gerência de projetos, lista de e-mails, entre outros.

A principal vantagem deste trabalho em equipe foi a dinâmica de grupo que acabou sendo aprendida e empregada pelos membros, enquanto a desvantagem foi a especialização das atividades, uma vez que as tarefas ficaram, às vezes, bastante divididas, fazendo com que o conhecimento ficasse disseminado entre alguns membros.

A turma da disciplina, 20 alunos, foi dividida em dois grupos de 10. Este número foi fundamental para o sucesso da disciplina, pois, desta forma, as atividades puderam ser mais bem divididas entre os membros, não gerando uma sobrecarga. Adicionalmente, o não cumprimento de alguma tarefa por parte de algum membro não ocasionou grande impacto para o grupo, diferente de outras edições da disciplina, que os grupos foram compostos, em média, por 5 alunos. Nestes casos, a desistência de um aluno da disciplina ou o não cumprimento de suas atividades implicou negativamente no desempenho do grupo.

Para um bom desenrolar do Estudo Integrado, a interação entre os professores que atuaram na disciplina foi muito importante. Com este fim, houve reuniões obedecendo pelo menos uma periodicidade semanal. Essa sinergia permitiu um melhor direcionamento do conteúdo da aula expositiva e condução dos grupos tutoriais, tornando o aprendizado mais estimulante. Porém, há um consenso que o ideal é que haja uma reunião antes de cada Grupo Tutorial para que seja discutido o andamento de cada grupo e como os projetos devem ser discutidos pelos alunos.

Por fim, vale mencionar o sistema de avaliação do Estudo Integrado que envolve uma certa complexidade. Os alunos relataram que as notas finais parecem ser “números mágicos”.

Isto acontece devido a tantas ponderações, e até o momento, os professores concordam que é uma boa forma de avaliação. Para uma boa aceitação deste sistema pelos estudantes, é essencial que os professores demonstrem de forma transparente como funcional tal sistema.

Quanto a integração das disciplinas, uma proposta de integração foi realizada por [Alves & Benitti, 2006], onde é feita uma integração entre Engenharia de Software e Análise e Projeto de Sistemas ao longo de três semestres. E mesmo Banco de Dados sendo uma disciplina extremamente relacionada com tópicos da Engenharia de Software, não foi encontrado nenhum trabalho que expressa-se tal preocupação. Normalmente, os projetos das disciplinas de Engenharia de Software usam banco de dados, porém os alunos cursam a disciplina relacionada separadamente.

5 TRABALHOS CORRELATOS

Quanto a integração das disciplinas, uma proposta de integração foi realizada por [Alves & Benitti, 2006], onde é feita uma integração entre Engenharia de Software e Análise e Projeto de Sistemas ao longo de três semestres. E mesmo Banco de Dados sendo uma disciplina extremamente relacionada com tópicos da Engenharia de Software, não foi encontrado nenhum trabalho que expressa-se tal preocupação. Normalmente, os projetos das disciplinas de Engenharia de Software usam banco de dados, porém os alunos cursam a disciplina relacionada separadamente.

Buscando na literatura, verifica-se que nas disciplinas de Engenharia de Software geralmente aplica-se projeto para ser desenvolvido em equipe como meio para o aprendizado prático da disciplina [Alves & Benitti, 2006; Sarinho, 2005; Laplante, 2006; Soares, 2004]. Entretanto, apesar de o trabalho em equipe ser fundamental para o bom desenvolvimento de um projeto de software, este não é o foco destas disciplinas uma vez que as questões relatadas se restringem aos aspectos mais técnicos. Além disso, não são adotados projeto real, até mesmo porque é uma tarefa árdua, que envolve uma série de variáveis como lhe dar com a expectativa gerada para o cliente, carga horária da disciplina versus prazos, entre outras.

Em parte, esses pontos (i.e. trabalho em equipe e projeto real) podem ser contornados com adoção de PBL. O que reforça isso, é que outros autores [Armarego, 2002; Ludi et al, 2005; Beng et al, 2003] têm adotado o método, relatando bons resultados, apesar de que [Ludi et al, 2005] não deixa claro se usou ou não projeto real.

Traçando um comparativo destes trabalhos com o adotado, em termos de inovação, conclui-se que o trabalho desenvolvido é o único que aplica PBL na integração de três disciplinas correlatas, em um único semestre, usando projeto real (i.e. projeto de um sistema a ser usado na comunidade).

6 CONSIDERAÇÕES FINAIS

A abordagem de ensino de Engenharia de Software com base em um Estudo Integrado de Engenharia de Software, Análise e Projeto de Sistemas e Banco de Dados, possibilitou uma experiência profissional e permitiu ao estudante vivenciar situações reais de engenharia de software. Além desta integração ter ficado muito clara para os alunos, através da aplicação da metodologia PBL, foi possível aplicar os conceitos das disciplinas na prática, ao mesmo que competências técnicas e não-técnicas dos alunos eram exercitadas.

Um importante benefício desta abordagem proposta de ensino da Engenharia de Software é a vivência profissional participativa, obtida através do desenvolvimento de um sistema real. Os trabalhos apresentados pelos alunos refletiram sistemas bem próximos da realidade exigida pelo mercado de trabalho. As diversas etapas necessárias para o desenvolvimento de um software profissional foram executadas.

Além disso, este trabalho possibilitou uma dinâmica de grupo que acabou sendo aprendida e empregada pelos membros através de atividades comuns no trabalho em equipe tais como expressar e ouvir opiniões, tomada de decisões, comunicação colaborativa, entre outras.

Para o próximo período, busca-se construir um novo processo de software adaptado a disciplina, baseado em alguma modelagem ágil [Beck, 2006], no intuito de que os alunos consigam finalizar o software e tenha uma maior experiência na etapa de testes, que foi pouco explorada. Com isto, pode-se eliminar artefatos que acabaram não sendo usados na etapa de implementação como diagramas de comportamento.

Para tanto, é preciso que a rever a seqüência de apresentação das aulas teóricas de cada um dos módulos abordados no Estudo Integrado. Por fim, ainda será escolhido um framework para aperfeiçoar o desenvolvimento do software a ser entregue, como produto do projeto.

7 REFERÊNCIAS BIBLIOGRÁFICAS

AHO, A.V.; SETHI, R.; ULLMAN, J.D. **Compiladores: Princípios, Técnicas e Ferramentas**. Rio de Janeiro: LTC, 1995.

ALVES, A.G.; BENITTI, F.B.V. **Processo de Desenvolvimento Integrando Disciplinas de Engenharia de Software**. In: XIV Workshop sobre Educação em Computação – Anais do XXVI Congresso da Sociedade Brasileira de Computação – Anais do, p. 206-215, Campo Grande, Mato Grosso do Sul, 2006.

ARMAREGO, J. **Advanced Software Design: a Case in Problem-based Learning**. In: CSEET.02. 2002.

BECK, K. **Programação Extrema (XP) explicada**. Porto Alegre: Bookman, 2004.

BENG, J.D. Delaney *et al.* Software Engineering Meets Problem-Based Learning. **The Engineers Journal** 17(6). 2003.

BITTENCOURT, R.A. & FIGUEIREDO, O.A. **O Currículo do Curso de Engenharia de Computação da UEFS: Flexibilização e Integração Curricular**. In: XI Workshop sobre Educação em Computação – Anais do XXIII Congresso da Sociedade Brasileira de Computação – Anais do, p. 171-182, Campinas, São Paulo, 2003.

BOUD, D. & FELETTI, G. **The Challenge of Problem-Based Learning**. London: Kogan Page, 1997.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. Rio de Janeiro: Campus, 2000.

DELISLE, R. **How to use problem-based learning in the classroom**. ASCD: Alexandria, Virginia, EUA. 1997.

- DUCH, B.J.; GROH, S.E.; ALLEN, D.E. **The Power of Problem-Based Learning**: a practical “how to” for reaching undergraduate courses in any discipline. Virginia: Stylus Publishing, LLC, 2001.
- LUDI, S. *et al.* **An Introductory Software Engineering Course that Facilitates Active Learning**. In: SIGSCE. St. Louis, Missouri, 2005.
- MCCONNELL, S. C. **Complete**: Um guia prático para a construção de software. Porto Alegre: Bookman, 2005.
- PRESSMAN, R.S. **Engenharia de Software**. Rio de Janeiro: McGraw-Hill, 2002.
- RYAN, C.; KOSCHMANN, T. **The Collaborative Learning Laboratory**: A Technology-Enriched Environment to Support Problem-Based Learning. Proceedings of the Annual National Educational Computing Conference. Boston, 1994.
- SARINHO, V. “**Usando Atividades Práticas e Avaliação Contínua no Ensino de Engenharia de Software**”. In: XXV Congresso da Sociedade Brasileira de Computação – Anais do XIII Workshop sobre Educação em Computação. São Leopoldo, Rio Grande do Sul, 2005.
- SILBERSCHATZ, A.; KORH, H.F.; SUDARSHAN, S. **Sistema de Banco de Dados**. São Paulo: Pearson Education do Brasil, 1999.
- SOARES, M.S. **Uma experiência de ensino de Engenharia de Software orientada a trabalhos práticos**. In: I Workshop sobre Educação em Computação do Rio de Janeiro/Espírito Santo. Campo Grande, Mato Grosso, 2004.
- SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson, 2003.
- THOMAS, J.W. **A Review Of Research On Project-Based Learning**. Relatório Técnico. Fundação Autodesk, San Rafael, Califórnia, EUA, 2000.
- WOODS, D.R. **Problem-based Learning**: resources to gain the most from PBL". Waterdown, ON, 1996.

APLICANDO *PROJECT-BASED LEARNING* NO ESTUDO INTEGRADO DE ENGENHARIA DE SOFTWARE, ANÁLISE E PROJETO DE SISTEMAS E BANCO DE DADOS

Abstract: *Software Engineering, adopting Project-Based Learning as an integrated course, provides a professional experience, allowing the student to feeling real software development situations. The process automation following technological evolution and competitiveness, gives to the Software Engineering an important position at technological courses. As a large course, Software Engineering includes many knowlege areas (i.e. requirements engineering, software quality, and others). Teaching these knowleges without integration, looking for your practical aspects and time constraints to apply it, becomes a difficult work, expensive and without quality assurance. This paper describes an experience in Software Engineering teaching using an integrated course of Software Engineering, Systems Design and Database Systems, adopting PBL (Project-Based Learning) as the practical teaching approach.*

Key-words: *Software Engineering, Project-Based Learning, Discipline Integration*