

Prática Integrada: Uma Abordagem Didático-Pedagógica Baseada em Projetos Colaborativos

Carlos Michel Betemps, Cristian Cechinel, Reginaldo da Nóbrega Tavares

Universidade Federal do Pampa – UNIPAMPA/Bagé
Rua Carlos Barbosa S/Nº - Bairro Getúlio Vargas
CEP 96412-420 Bagé, RS, Brasil

{carlos.betemps@unipampa.edu.br, contato@cristiancechinel.pro.br,
regi@unipampa.edu.br}

Abstract. *This paper describes the discipline of Introduction to Computer Engineering of the Computer Engineering course at the Federal University of Pampa. This discipline uses the Project Based Learning Pedagogical Approach and is executed using projects as a way to integrate and enhance the studied subjects in other two disciplines. This paper also describes the projects developed during the semester, the evaluation criteria adopted, the problems founded, and some suggestions for future improvements. Finally, some advantages in the adoption of this pedagogical approach and the final remarks are presented.*

Resumo. *Este trabalho descreve a disciplina de Introdução à Engenharia de Computação do curso de Engenharia de Computação da Universidade Federal do Pampa. Esta disciplina utiliza a Abordagem Pedagógica de Aprendizagem Baseada em Projetos Colaborativos e é conduzida utilizando estes projetos de forma a integrar e aprofundar os conteúdos abordados em outras duas disciplinas do curso. Neste relato são descritos os projetos desenvolvidos ao longo do semestre, os critérios de avaliação adotados, as dificuldades e problemas, além de alternativas e soluções para os problemas encontrados. Por fim, são descritas algumas das vantagens observadas na adoção dessa abordagem pedagógica e as considerações finais.*

1. Introdução

O ritmo acelerado das mudanças no campo da computação exige que o engenheiro de computação seja um aprendiz ao longo de toda sua vida, sendo capaz de manter seu conhecimento e suas habilidades dentro das disciplinas por ele escolhidas [Shackelford et. al. 2005]. Sendo assim, o currículo do curso deve estar preocupado em formar um profissional autônomo, inquiridor, motivado e desejoso de novas informações, capaz de aplicar praticamente os conhecimentos teóricos estudados, mas sobretudo, pró-ativo na busca das soluções para os problemas que lhe são freqüentemente apresentados.

As abordagens tradicionais de ensino, baseadas em aulas expositivas e experimentos em laboratório, restringem os estudantes a uma postura passiva frente aos conteúdos que lhe são apresentados. Normalmente, as disciplinas são distribuídas por áreas de conhecimento, que por sua vez estão relacionadas aos departamentos das universidades. Nesse contexto, o professor trabalha na preparação de sua disciplina específica (que funciona da mesma maneira para os inúmeros cursos existentes com os

quais o departamento interage), sem a preocupação de relacionar os conteúdos da mesma com as aplicações práticas possíveis, nem com a sua importância para a formação dos acadêmicos de um curso específico. Essa postura faz com que os acadêmicos, por sua vez, ignorem a importância daquele componente curricular, preocupando-se apenas em passar nos testes agendados e eliminar a disciplina. Uma vez aprovado, dificilmente o estudante retoma aqueles conteúdos para aplicá-los na solução dos problemas apresentados em outras disciplinas, e normalmente ainda divulga para os demais colegas a imagem de que tais conteúdos foram inúteis para sua formação. Para romper com os métodos tradicionais de ensino, é preciso que os currículos sejam organizados de maneira que os acadêmicos entendam a utilidade dos conteúdos que lhe são apresentados, permitindo que os mesmos sejam utilizados na resolução de problemas e situações confrontadas no cotidiano pelos profissionais daquela área de formação.

Conscientes da necessidade de reestruturação dos currículos tradicionais, diversas universidades já realizaram mudanças radicais nas abordagens utilizadas nos seus processos de ensino-aprendizagem. Uma das principais abordagens que veio contribuir nesse sentido é a Aprendizagem Baseada em Problemas (ABP) [Fink 2001] - que surgiu no final da década de 60 no Curso de Medicina da Universidade de McMaster no Canadá. Nessa metodologia o aluno é considerado como o ator responsável por seu aprendizado, o que inclui a organização de seu tempo e a busca das oportunidades para aprender. Problemas que refletem a realidade são utilizados como estímulos para a aquisição de conhecimentos e habilidades, conduzindo os estudantes a reflexão de uma determinada temática dentro de contextos específicos. A discussão desses problemas é realizada sem que necessariamente aconteçam exposições formais prévias das informações necessárias para sua resolução, colocando os estudantes frente a uma situação desafiadora, onde obstáculos na aprendizagem devem ser superados para a resolução apropriada do problema apresentado.

Para facilitar a discussão, os acadêmicos são divididos em grupos de número reduzido, sendo que cada grupo é orientado por um professor tutor. Os acadêmicos discutem o problema, elegem os seus próprios objetivos de aprendizagem, e retornam para discutir e compartilhar os resultados que alcançaram em seus estudos. Em paralelo às discussões sobre o problema, os acadêmicos são apresentados a conteúdos relacionados ao mesmo, relacionando assim, esse conteúdos teóricos com a sua prática imediata.

Essa metodologia transfere o acadêmico de um ambiente em que ele memoriza, armazena e reproduz informações, para um ambiente onde ele desenvolve a habilidade de resolver problemas, descobre relações entre as diferentes disciplinas e entende os diferentes tópicos estudados como naturalmente conectados [Lambrix and Kamkar 1998].

Apesar de ter surgido nos cursos de Medicina, a ABP também é largamente utilizada em cursos de engenharia, por meio de um modelo adaptado denominado de Aprendizagem Baseada em Projetos. Nesse modelo, o projeto deve ser encarado de forma diferente dos projetos tradicionais já utilizados nas universidades, que normalmente servem como síntese de conhecimentos prévios e que costumam vir após a explanação e avaliação teórica [UEFS 2008]. Aqui, o projeto funciona de modo semelhante a um problema, servindo para guiar o aprendizado dos acadêmicos ao longo do semestre e colocando-os frente a situações reais semelhantes as encontradas no

cotidiano de sua futura profissão. O processo percorrido pelos acadêmicos para o desenvolvimento do projeto deve possibilitar também o desenvolvimento de habilidades de comunicação e argumentação, além da experiência de atuação em equipes multidisciplinares (situação comum no cotidiano das empresas e indústrias). Exemplos de sucesso na utilização da ABP em cursos de Engenharia podem ser observados no curso de Engenharia Elétrica da Universidade de Tecnologia de Delf na Holanda; no Curso de Engenharia de Computação da Universidade de Aalborg na Dinamarca [Fink 2001]; e aqui no Brasil no curso de Engenharia da Computação da Universidade Estadual de Feira de Santana-UEFS.

O presente trabalho está organizado da seguinte maneira: na seção 2 são apresentados algumas das características desejadas para o profissional formado no curso de Engenharia da Computação da Unipampa; na seção 3 são descritos o funcionamento de uma disciplina que utiliza a Prática Integrada e os projetos executados na experiência aqui relatada; a seção 4 apresenta alguns dos problemas encontrados ao longo dessa experiência e algumas das possíveis soluções para os mesmos; na seção 5 são ressaltadas as principais vantagens observadas na utilização da abordagem proposta; e na seção 6 são apresentadas as considerações finais. Acreditamos que a experiência aqui apresentada pode ajudar a promover a melhoria do processo de ensino-aprendizagem da computação no Brasil, dentro do contexto dos “Grandes desafios da Computação” no tema “Formação multidisciplinar: experiências e tendências”.

2. O Curso de Engenharia de Computação da Unipampa

O Curso de Engenharia de Computação da Universidade Federal do Pampa iniciou suas atividades no segundo semestre de 2006 e tem uma proposta pedagógica tradicional de ensino e aprendizagem. Porém, em seu currículo são reservados componentes curriculares voltados para trabalhos integrados na área da engenharia de computação, de forma que conteúdos abordados em diferentes disciplinas dentro do mesmo semestre sejam aplicados nos trabalhos desenvolvidos no componente curricular que integra (daí o termo *Prática Integrada*) os conteúdos daquele semestre (ou de algumas disciplinas do semestre). O principal objetivo dentro desses componentes curriculares é o de desenvolver no acadêmico as seguintes características: 1) habilidade para organizar trabalhos na forma de projetos e planejar as atividades de acordo com algum conceito metodológico; 2) habilidade de sempre pensar em termos de soluções alternativas baseadas na prévia formulação de critérios; 3) forte capacidade de comunicação e de argumentação; e apurada desenvoltura para atuar participativa e colaborativamente em equipes multidisciplinares; 4) raciocínio articulado para a resolução de problemas no âmbito de engenharia de computação e capacidade de relacionar as diferentes teorias e técnicas estudadas com suas aplicações práticas; 5) habilidade para atuar de forma autônoma e inquiridora, reconhecendo a necessidade de um aprendizado contínuo e vitalício; e 6) capacidade para engajar-se nesse aprendizado em seus diversos níveis - conhecimento, compreensão, aplicação, análise, síntese, e avaliação - conforme proposto na taxionomia de Bloom [Bloom et. al. 1996].

3. A Prática Integrada Inserida em uma Disciplina

Na primeira fase do curso, foram reservadas 34 horas/aula presenciais (uma disciplina de 2 horas aula semanais) para o componente curricular de Introdução à Engenharia de Computação. Esta disciplina interagiu diretamente com os componentes curriculares de

Algoritmos e Programação, e de Introdução à Arquitetura de Computadores (disciplinas colaborativas), embora outros conteúdos de outros componentes curriculares do curso também tenham sido utilizados.

3.1. Estrutura de Funcionamento da Disciplina

Uma disciplina de Prática Integrada é baseada na execução de projetos colaborativos. Os projetos visam a discussão e o estudo mais aprofundado de conhecimentos adquiridos nas disciplinas colaborativas.

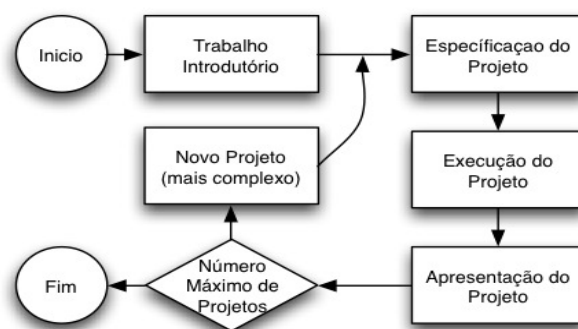


Figura 1. Fluxo das Atividades dos Projetos Realizados na Disciplina

O primeiro projeto é um trabalho introdutório sobre algum conteúdo teórico relevante para o desenvolvimento da disciplina, tal como a pesquisa bibliográfica e a confecção de documentos científicos, normalmente envolvendo conteúdos estudados em disciplinas de conteúdos básicos de cursos de engenharia e também específicos da área de computação. Os demais projetos são direcionados para assuntos práticos, normalmente envolvendo aspectos teóricos da computação que são programados e/ou simulados através da programação. A Fig. 1 ilustra o fluxo de execução das atividades dos projetos da disciplina. Após a apresentação do projeto, uma especificação do mesmo é estudada e formas de resolução são propostas e analisadas; em seguida parte-se para a execução do projeto, que é conduzida pelos grupos com a orientação do professor da disciplina e/ou outros professores do curso de Engenharia de Computação; finalmente a apresentação dos resultados é realizada. Se o número máximo de projetos para a disciplina não foi alcançado, um novo projeto é apresentado e um novo ciclo é iniciado.

3.2. Relato de Execução da Disciplina de Introdução à Engenharia de Computação

Durante a primeira execução da disciplina de Introdução à Engenharia de Computação (com o uso de Prática Integrada), os acadêmicos foram organizados em grupos (com 5 integrantes em média), que se mantiveram iguais ao longo de todo o semestre. Foram apresentados pequenos projetos envolvendo diferentes conteúdos relacionados aos componentes curriculares do 1º semestre do curso de Engenharia de Computação.

À medida que os projetos aumentavam de complexidade eram disponibilizados documentos detalhados que descreviam especificações quanto aos procedimentos que deveriam ser adotados para o seu desenvolvimento. Após a apresentação e a explicação de cada projeto, as diferentes equipes eram dispensadas, ficando encarregadas de desenvolver o projeto dentro de um período aproximado de duas semanas (dependendo

de cada projeto). Durante o período de desenvolvimento os acadêmicos contavam com a orientação do professor responsável pelo componente curricular, assim como, com a orientação dos outros professores do corpo docente. Após o período estabelecido, e dependendo do projeto, os acadêmicos retornavam para um encontro com toda a turma e apresentavam os resultados de suas soluções para os demais colegas, ou simplesmente para o professor.

Quanto a avaliação, e no que se refere especificamente às apresentações expositivas, foram utilizados os seguintes critérios: 1) o tempo de apresentação (quando a apresentação tinha um formato de seminário, cada grupo teve cerca de 30 minutos para apresentar seus trabalhos com tolerância aproximada de 5 minutos); 2) a organização do grupo (quanto ao andamento da apresentação e questionamento sobre a forma de execução do trabalho – quanto a divisão das tarefas e organização do grupo); e 3) o domínio do assunto por cada aluno do grupo (quanto a desenvoltura do aluno em apresentar a sua parte do trabalho e/ou responder as dúvidas apresentadas pelos demais alunos e pelo professor).

As apresentações expositivas dos projetos seguiram um cronograma previamente elaborado no início do semestre, sendo que ao todo foram trabalhados 6 (seis) projetos ao longo do mesmo, descritos a seguir:

Projeto 1 - Estudo sobre a importância das diversas áreas estudadas no curso de Engenharia de Computação. Entre as áreas estudadas estavam: Física, Cálculo, Lógica, Matemática Discreta e Teoria da Computação, Algoritmos, Programação e Estruturas de Dados, Probabilidade e Estatística, Arquitetura de Computadores, Álgebra Linear e Geometria Analítica, Eletrônica e Análise de Circuitos, Sistemas Operacionais, Comunicação de Dados e Redes de Computadores. Cada grupo de alunos foi responsável por estudar uma dessas áreas e apresentar um artigo descrevendo a importância e/ou aplicação da respectiva área dentro da profissão do Engenheiro de Computação. **Formato de Apresentação dos Resultados:** Exposição dos projetos desenvolvidos para os demais acadêmicos da turma em formato de seminário e entrega de artigo sobre o assunto estudado. **Crêterios de Avaliaçãõ:** O artigo final foi avaliado com relação a boa escrita, a existência de cópia de textos da internet (integral ou parcial), a existência e qualidade de referências bibliográficas, e a iniciativa de apresentação de exemplos práticos de aplicação dos conteúdos do tema apresentado. Para a avaliação do seminário, foram utilizados os critérios anteriormente descritos para as apresentações expositivas, além de observar a utilização adequada dos recursos audiovisuais. As equipes que não apresentaram seus artigos oralmente foram avaliadas somente quanto ao artigo.

Projeto 2 - Estudo sobre algumas funções, procedimentos e comandos da linguagem de programação Pascal. As equipes estudaram características de funções, procedimentos e comandos em Pascal que trabalhavam com: operações e manipulação de conjuntos, operações trigonométricas e aritméticas especiais, manipulação e formatação de números reais, operações de manipulação de *strings* (literais), definição de tipos abstratos de dados (TAD) e criação e manipulação de arquivos. Como resultado deste estudo, cada equipe gerou um documento em formato de glossário (ou manual de consulta rápida) que continha a descrição das funcionalidades encontradas, além de algumas situações e exemplos de aplicação destas funcionalidades. **Formato de Apresentação dos Resultados:** Exposição dos projetos desenvolvidos para os demais acadêmicos da turma em formato de seminário e entrega de documento contendo a

descrição das funcionalidades encontradas. Ocorreu também a realização de debate entre as equipes que apresentaram. **Avaliação:** O documento entregue e a exposição realizada foram avaliados a partir dos mesmos critérios utilizados no projeto 1. As equipes que não apresentaram seus trabalhos foram avaliadas somente quanto ao documento (glossário) elaborado.

Projeto 3 – Identificação e correção de erros em programas codificados em Pascal. As equipes foram apresentadas aos códigos-fontes de alguns pequenos programas em Pascal que continham erros léxicos, sintáticos, semânticos e de adequação às funcionalidades previstas (juntamente com alguns destes programas estavam descritas quais as funcionalidades e objetivos dos mesmos). Os erros foram identificados e corrigidos por cada equipe que, por sua vez, redigiram um relatório sobre as soluções desenvolvidas. **Formato de Apresentação dos Resultados:** Exposição dos projetos desenvolvidos para o professor, e entrega de relatório sobre os erros encontrados e as soluções propostas para estes erros nos programas estudados. **Avaliação:** Além dos critérios utilizados nos projetos anteriores para a avaliação dos trabalhos escritos, considerou-se também se as equipes conseguiram identificar todos os erros existentes nos códigos e se sugeriram alternativas para a resolução dos mesmos. A qualidade das alternativas sugeridas também foi avaliada, sempre observando se a proposta continha uma melhoria do código, uma otimização do tempo de execução do programa e/ou da utilização de memória. Os grupos que não apresentaram os seus resultados ao professor, foram avaliados somente quanto ao relatório entregue. Os programas com a inserção de erros apresentados aos grupos foram, em sua maior parte, extraídos de [Ascencio and Campos 2002].

Projeto 4 – Conversor de bases numéricas. Nesse projeto as equipes desenvolveram um programa de conversão de números entre diferentes bases numéricas (binária, octal, decimal e hexadecimal). No documento de descrição do projeto, além da funcionalidade principal de conversão, também foi definido que o programa deveria permitir a entrada do número a ser convertido em qualquer base, e ser capaz de lidar com números binários inteiros e positivos de no mínimo 32 bits. **Formato de Apresentação dos Resultados:** Exposição para o professor do programa desenvolvido e de seu código-fonte, além da entrega em meio digital desse código-fonte. **Avaliação:** O programa desenvolvido foi avaliado de acordo com alguns critérios estabelecidos como: tratamento de erros de entrada de dados e de erros de tempo de execução, informações fornecidas na interface do programa (ex.: existência de informações sobre o que está acontecendo internamente no programa e sobre o que o usuário deve informar para o mesmo, apresentação dos resultados de forma clara e direta), qualidade da documentação do código – todos os grupos deveriam entregar o código-fonte bem documentado, indicando para cada função, procedimento ou parte de programa a sua contribuição para o programa como um todo, adequação do programa quanto às funcionalidades indicadas na descrição do projeto, e inexistência de laços infinitos e/ou erros durante a execução do programa.

Projeto 5 – Calculadora de números binários. O quinto projeto estava voltado para a implementação em Pascal de uma calculadora de números binários. A calculadora deveria fornecer soluções para cálculos aritméticos utilizando os operadores de soma, subtração, multiplicação (obrigatoriamente) e divisão (opcionalmente) entre números binários. Também foi estabelecido que: os números binários deveriam ser representados em complemento de 2 (dois) e que as operações deveriam ser realizadas considerando

esta representação (nas demais bases era permitida a utilização do sinal negativo '-'), deveriam ser consideradas situações de estouro de representação, a calculadora deveria ter capacidade de lidar com números binários de, no mínimo, 32 bits, poderia ser utilizado qualquer método para realizar a multiplicação de números binários inteiros desde que fosse indicada qual referência em que o método havia sido consultado (o método descrito em [Weber 2001] foi indicado como um dos algoritmos possíveis, sendo que este deveria ser modificado para lidar com números negativos), e o programa deveria simular os itens de hardware que são utilizados para as operações em computadores reais (por exemplo, se fosse necessário ser utilizado um somador completo para realizar uma operação de multiplicação, então este somador deveria ser simulado por uma parte do programa da calculadora). **Formato de Apresentação dos Resultados:** Exposição para o professor do programa desenvolvido e de seu código-fonte, além da entrega, em meio digital, desse código-fonte. **Avaliação:** O projeto foi avaliado a partir dos mesmos critérios utilizados no projeto 4.

Projeto 6 - Calculadora para realizar operações aritméticas entre números binários representados em ponto flutuante. O último projeto tinha como objetivo a construção de uma calculadora para operações aritméticas entre números binários com representação em ponto flutuante. Para isto deveria ser utilizado o padrão 754 da IEEE [Goldberg 1991]. Da mesma maneira que no projeto anterior, foram estabelecidos alguns aspectos que deveriam ser contemplados pelo programa, tais como: todo número fornecido como operando para a calculadora deveria ser representado internamente em ponto flutuante (na base binária) e de acordo com o padrão IEEE 754, os números representados em ponto flutuante neste formato deveriam estar sempre normalizados, uma função de conversão de um número qualquer para um número representado em ponto flutuante deveria ser definida, o programa deveria aceitar, no mínimo, números na base decimal e na base binária, mas o mesmo deveria fazer a conversão para o formato de ponto flutuante (na base binária) e as operações deveriam ser programadas para serem realizadas com base neste formato (ponto flutuante ou vírgula flutuante) e números fracionários (reais) deveriam ser aceitos pela calculadora. Além dessas características para o programa, também foi solicitado que o código-fonte desenvolvido fosse bem documentado e que os grupos entregassem um artigo (desenvolvido pelo próprio grupo) explicando o formato da IEEE para números em ponto flutuante e o funcionamento da aritmética entre esses números. Por último, os alunos foram fortemente encorajados a reutilizar nesse projeto os códigos desenvolvidos para os projetos anteriores. **Formato de Apresentação dos Resultados:** Exposição para o professor do código-fonte do programa desenvolvido (exposição realizada por todas as equipes de alunos), entrega em meio digital desse código-fonte, e entrega de artigo descrevendo o formato para representação de números em ponto flutuante (padrão IEEE 754). **Avaliação:** O projeto foi avaliado a partir dos mesmos critérios utilizados no projeto 4. O artigo descrevendo o padrão IEEE 754 para representação de números em ponto flutuante e a aritmética de números representados neste padrão foi avaliado segundo os critérios citados no projeto 1 (itens referentes ao artigo).

4. Dificuldades e soluções para os problemas encontrados

Na tabela 1 são apresentadas algumas dificuldades e problemas observados ao longo do semestre durante a execução da disciplina de Introdução à Engenharia de Computação com a utilização de Prática Integrada.

Tabela 1. Problemas encontrados e soluções possíveis

Problema	Solução (Possível)
1. Pouca utilização dos horários de atendimento: Nem todos os alunos utilizavam o horário de atendimento (com o professor) previamente reservado no cronograma da disciplina.	Estabelecer junto ao cronograma a entrega de um relatório parcial sobre o projeto (ou de partes da solução) de modo que os alunos se organizem para discussão desses resultados com os professores disponíveis.
2. Desconhecimento dos critérios de avaliação dos projetos por parte dos acadêmicos: Foi repassado para os alunos uma descrição geral de como seria realizada a avaliação dos projetos, porém as especificidades relacionadas a avaliação de cada projeto não estava bem clara para os mesmos.	Apesar desse problema não ter comprometido o desenvolvimento dos trabalhos, julga-se importante estabelecer desde o primeiro dia de aula os critérios de avaliação que serão utilizados em cada um dos projetos, assim como uma lista de objetivos que devem ser atingidos durante o desenvolvimento dos mesmos.
3. Distribuição desigual de atividades entre os membros dos grupos: Foi constatado que na maioria dos grupos a carga de trabalho de cada componente não estava sendo igual em termos de tempo de dedicação e de tipos de atividades executadas. Em relação as atividades, percebeu-se que uma parte dos membros dos grupos normalmente se ocupavam de aspectos relacionados a pesquisa, documentação e testes, enquanto uma segunda parte tratava dos aspectos práticos, como a programação.	Solicitar com antecedência aos grupos qual a função que cada integrante irá desempenhar em determinado projeto e estabelecer que tal função deve ser diferente para cada projeto ao longo do semestre. A definição de um gerente de projeto que seja responsável por essa distribuição pode ser uma alternativa interessante. Uma outra solução seria a implementação de avaliações individuais (e não somente de grupos) de modo que os acadêmicos se preocupem em participar de todos os tipos de atividades existentes.
4. Não houve compartilhamento de todas soluções entre todos os acadêmicos da turma: As avaliações dos projetos 3, 4, 5 e 6 foram realizadas de modo que somente o grupo e o professor interagiam naquele momento, fazendo com os demais grupos ficassem realizando tarefas relativas aos seus projetos (que seria, de certa forma, natural) ou mesmo atividades relacionadas a outras disciplinas do curso.	Essa situação indica a necessidade de que todos os projetos devem ser apresentados por todos os grupos - isto pode gerar um cronograma apertado para a disciplina quando existirem turmas com muitos alunos (situação normal em turmas de primeiro período). Uma solução aparentemente eficiente seria a diminuição do número de projetos e o conseqüente aumento do tempo de elaboração e de apresentação dos mesmos.
5. Avaliação não mediu suficientemente o progresso dos acadêmicos: Ocorreram casos em que alunos que foram aprovados na disciplina de Prática Integrada, não conseguiram o mesmo êxito nas disciplinas de Algoritmos e Programação e de Introdução a Arquitetura de Computadores, indicando possíveis falhas no processo de avaliação dessa disciplina.	Como alternativa para esse problema poderiam ser utilizadas, como parte da avaliação, provas práticas individuais com os alunos, de forma a identificar mais facilmente os alunos com dificuldade em programação ou no entendimento de conteúdos de arquitetura de computadores.

5. Vantagens observadas na inserção da Prática Integrada

Os projetos de Prática Integrada permitiram aos acadêmicos ter uma visão geral das diversas áreas que compõem a formação de um engenheiro de computação e da importância e aplicação dessas áreas na resolução de problemas cotidianos enfrentados por esses profissionais. Ainda, os acadêmicos foram precocemente introduzidos em situações que necessitavam obrigatoriamente de interação em grupo, enfrentando dificuldades relacionadas a distribuição de tarefas e a administração de conflitos pessoais.

Voltando aos aspectos técnicos, pôde-se observar um crescimento significativo no processo de pesquisa científica, tanto nos aspectos relacionados a busca pela informação quanto no que se refere a construção de documentos como artigos e relatórios. Essas habilidades são consideradas fundamentais em qualquer curso

universitário, entretanto a forma utilizada para o desenvolvimento das mesmas é ainda objeto de estudo dentro do meio acadêmico. A inserção de componentes curriculares com o uso de Prática Integrada indica uma possível alternativa para facilitar o desenvolvimento desse tipo de habilidade.

O componente curricular que faz uso de Prática Integrada também foi importante por proporcionar um ambiente para a exploração de práticas inerentes à programação de computadores que não puderam ser abordadas na disciplina de Algoritmos e Programação (por questões de tempo), tais como: incentivo a organização das tarefas de cada integrante para um grupo de programação e pré-definição das estruturas de dados e de programas que serão utilizadas nos projetos de programação; construção de rotinas que fazem o tratamento de erros dos dados de entrada, e realização da documentação detalhada dos códigos fonte dos programas construídos. Esse espaço extra para o desenvolvimento de programas pode ter servido como catalisador para o aprimoramento de habilidades de programação em alguns acadêmicos que já demonstraram no primeiro período do curso um excelente domínio e desenvoltura nessa área. Além de auxiliar na prática da programação, a Prática Integrada auxiliou na compreensão da arquitetura dos computadores, pois a partir do desenvolvimento de alguns dos projetos, os acadêmicos tiveram a oportunidade de fazer uma analogia de como o hardware do computador funciona através da simulação de itens de hardware.

A disciplina de Introdução à Engenharia de Computação foi executada uma segunda vez utilizando a mesma formatação, no entanto ocorreu uma diminuição do número de projetos realizados - 4 projetos. Esta diminuição ocorreu por causa da constatação que alguns grupos não conseguiram executar seus projetos por completo e que o tempo de apresentação dos resultados e conseqüente avaliação foram curtos, podendo ser melhorados se o tempo de execução para cada projeto fosse aumentado. Os projetos propostos nesta segunda execução foram os projetos de números 1, 2, 5 e 6 (conforme os números de identificação de cada projeto na descrição anterior – seção 3.2). Os projetos tiveram uma duração média de 3 semanas, chegando a 4 semanas nos últimos projetos. Essa alteração no formato da disciplina proporcionou aos alunos mais tempo para a execução de seus projetos, possibilitando que os mesmos fossem desenvolvidos de forma mais completa quando comparados ao semestre anterior. Para o professor, foi possível realizar uma avaliação mais detalhada dos resultados alcançados em cada projeto de cada grupo.

6. Considerações Finais

Pode ser observado, como descrito na seção 5, que a execução de um componente curricular com o uso de Prática Integrada propiciou aos alunos uma experiência positiva em vários aspectos. No que diz respeito ao aprendizado de algoritmos, programação e arquitetura de computadores, observou-se que uma boa forma de solidificar conteúdos abordados em diferentes disciplinas é a aplicação destes conteúdos em projetos como os descritos na seção 3.2 - como nos trabalhos que envolviam a implementação de itens de software que simulavam itens de hardware. Salienta-se, também, que mesmo com pouca experiência em atividades de programação, os alunos tiveram a oportunidade de serem avaliados e/ou cobrados quanto a aspectos não necessariamente ligados a uma linguagem de programação, mas quanto a forma como conduzir atividades de programação, como, por exemplo, mostrando preocupação quanto à documentação de código, tratamento de erros de usuário, otimização de código, melhora de interface com

o usuário, entre outras; este tipo de experiência adquirida em um semestre inicial de um curso de engenharia de computação poderá ser bastante útil para estes alunos.

Para os próximos semestres será elaborado um sistema de avaliação desse componente para medir não somente o desempenho dos acadêmicos, mas também o impacto desse componente curricular dentro da matriz curricular da primeira fase do curso. O formato para essa avaliação ainda é objeto de estudo dos autores. O fato da disciplina ter sido implantada nesse modelo desde o início do curso dificulta a realização de um estudo comparativo com relação a condução da disciplina no método tradicional.

Por último, é importante ressaltar que a abordagem apresentada propõe que os projetos sejam iniciativas didático-pedagogicamente estruturadas para alavancar determinados conteúdos ou habilidades necessários a formação do acadêmico. Nesse sentido, os projetos devem funcionar como catalisadores de um processo de busca por determinado conhecimento que deverá ser adquirido. Entretanto, a iniciativa descrita nesse artigo funciona mais como uma oportunidade de aplicação e aprofundamento de conhecimentos obtidos em outras disciplinas, do que propriamente como um evento catalisador para a busca de um novo conhecimento.

7. Referências

- Ascencio, A.F.G.; and Campos, E.A.V. (2002) “Fundamentos da Programação de Computadores: Algoritmos, Pascal e C/C++”, Prentice Hall, São Paulo.
- Bloom, B. S.; et. al. (1976) “Taxionomia de Objetivos Educacionais”, Globo, Porto Alegre
- Fink, F.K. (2001) “Problem Based Learning in Engineering Education - a catalyst for Regional Industrial Development”, In: World Transactions on Engineering and Technology Education, Vol.1, No.1., pp. 2932, Melbourne.
- Goldberg, D. (1991) “What Every Computer Scientist Should Know About Floating-Point Arithmetic”, In: ACM Computing Surveys, Vol.23, No.1., pp-548, ACM, New York.
- Lambrix, P.; and Kamkar, M. (1998) “Computer Science as an Integrated Part of Engineering Education”, In: SIGCSE Bulletin, Vol.30, No.3, pp. 153156, ACM, New York.
- Shackelford, R.; et. al. (2005) “Computing Curricula 2005 - The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology and Software Engineering”, ACM/AIS/IEEE-CS Joint Task Force for Computing Curricula 2005, IEEE Computer Society Press and ACM Press.
- UEFS. (2008) “Projeto Didático Pedagógico do Curso de Engenharia de Computação da Universidade Estadual de Feira de Santana”, Disponível em <http://www.uefs.br/ecomp/ProjetoDidaticoPedagogico.htm>. Acesso em: 27 março 2008.
- Weber, R.F. (2001) “Fundamentos da Arquitetura de Computadores”, Sagra-Luzzato, Porto Alegre.