

# Uma Proposta de Metodologia para o Ensino de Engenharia de Software

Rossana Maria de Castro Andrade Fabiana Gomes Marinho<sup>1</sup> Valéria Lelli  
Leitão<sup>2</sup> Lincoln Souza Rocha<sup>3</sup>

Departamento de Computação – Universidade Federal do Ceará (UFC)

{rossana,fgmarinho,valerialelli,lincoln}@lia.ufc.br.br

**Abstract.** In some Computer Science undergraduate courses, Software Engineering (SE) has been taught with insufficient hours to cover all theoretical concepts. Thus, the amount of concepts taught in class increases, leaving few time for discussions and practical work. This paper presents a practical-theoretical teaching methodology that has been adopted in the SE course of the Computer Science (CS) undergraduate course of the Federal University of Ceará (UFC). This methodology consists of including, in parallel to the theoretical part, the development of a software project. The proposed methodology facilitates the understanding of the concepts presented, in class and can also improve the student performance.

**Keywords:** Software Engineering, teaching methodology.

**Resumo.** Em alguns currículos de Ciência da Computação, a disciplina de Engenharia de Software (ES) é ministrada com créditos insuficientes para cobrir todos os conceitos da área. Isso causa um excesso de conteúdo abordado em sala de aula, não deixando espaço para discussões e trabalhos práticos. Este artigo apresenta uma metodologia de ensino teórico-prática adotada na disciplina de ES do Departamento de Computação (DC) da Universidade Federal do Ceará (UFC) que consiste em incluir, em paralelo ao ensino teórico, o desenvolvimento de um projeto de software. Com o uso dessa metodologia, espera-se facilitar o entendimento dos conceitos teóricos apresentados em sala de aula bem como melhorar o desempenho dos alunos.

**Palavras-chave:** Engenharia de Software, metodologia de ensino.

---

1 Aluna do Curso de Doutorado do Mestrado e Doutorado em Ciência da Computação (MDCC). Bolsista CNPQ desde 2008.1 – Processo N° 552924/2008.3.

2 Aluna do Curso de Mestrado do MDCC. Bolsista FUNCAP desde 2007.1.

3 Aluno do Curso de Doutorado do MDCC. Bolsista CNPQ desde 2008.1 – Processo N° 550924/2007-8.

## 1 Introdução

Segundo Piaget (PIAGET,1975), a educação permanente e a renovação incessante do conhecimento são extremamente necessárias, no entanto, o essencial não está somente em aprender, mas sim em aprender a aprender. A idéia proposta por Piaget, afirma que o que caracteriza a aprendizagem é o movimento de um saber fazer a um saber. Neste contexto, é enfatizada a necessidade de que a relação pedagógica entre professor e aluno seja elaborada com base metodológica e planejamento adequado. O professor deve ser responsável por um esforço construtivo, agrupando teorias modernas de aprendizagem.

De acordo com Demo (DEMO,1997), o aluno constrói o conhecimento através da interação com o que está sendo apresentado, portanto, cabe ao professor conduzir uma metodologia de ensino que enfatize a prática do conteúdo ensinado e estimule as idéias dos alunos. Seguindo essa mesma linha de pensamento, Fiorentini (FIORENTINI, 2002) e Pinheiro e Gonçalves (PINHEIRO e GONÇALVES, 2001), afirmam que o professor precisa utilizar técnicas de ensino dinâmicas e adequadas aos interesses dos alunos, com o objetivo de conquistar sua participação durante as aulas, ou seja, o professor precisa desafiar seus alunos de forma que eles busquem soluções para os problemas propostos.

A Engenharia de Software (ES) é uma disciplina relacionada com todos os aspectos da produção de software, desde os estágios iniciais até a sua manutenção, envolvendo não apenas os processos técnicos do desenvolvimento, como também as atividades de gerenciamento de projeto e a utilização de ferramentas, métodos e teorias que apoiem a sua produção (SOMMERVILLE, 2007). Portanto, ensinar todo o conteúdo da disciplina de ES por meio de aulas expositivas tradicionais pode dificultar o aprendizado dos alunos, devido à grande quantidade de informações na teoria envolvida.

A partir do enfoque educacional mencionado anteriormente é apresentada neste artigo uma metodologia de ensino proposta para a disciplina de ES do Departamento de Computação (DC) da Universidade Federal do Ceará (UFC). Este artigo apresenta ainda a avaliação dos resultados obtidos com a utilização da metodologia de ensino proposta desde 2005.

O restante deste artigo está organizado em cinco seções. Na seção 2 são apresentados os trabalhos relacionados e são destacadas as principais diferenças identificadas. Na seção 3 a metodologia de ensino proposta para a disciplina de ES é descrita. Na seção 4 os resultados obtidos com o uso da metodologia são avaliados. Na seção 5 as conclusões e os trabalhos futuros do trabalho são apresentados.

## 2 Trabalhos Relacionados

Em Aguiar (AGUIAR, 2004) é apresentado um modelo de processo de desenvolvimento de software denominado *easyProcess*. Este processo tem como objetivo auxiliar na gerência do desenvolvimento de aplicações em disciplinas da Engenharia de Software na graduação do Departamento de Sistemas e Computação da Universidade Federal de Campina Grande (UFCG). Nessa metodologia os conceitos teóricos são explorados por meio do desenvolvimento de um projeto de software. Porém, a principal diferença identificada com relação à metodologia proposta neste artigo está no fato dos alunos terem de utilizar um único modelo de processo de

desenvolvimento. Além disso, os temas das aplicações são previamente selecionados pelos professores do próprio departamento.

Em Alves (ALVES, 2006) uma metodologia de ensino é proposta para as disciplinas de Engenharia de Software da Universidade Federal do Pampa (UNIPAMPA). Nesta metodologia os alunos também praticam os conceitos apresentados através do desenvolvimento de um projeto de software. Entretanto esta metodologia difere da metodologia proposta neste artigo em dois pontos: o modelo de processo de software é definido previamente pelo professor, além disso, o projeto é iniciado na disciplina de Análise e Projeto de Sistemas I e concluído na disciplina Análise e Projeto de Sistemas II, tendo, portanto, uma duração de dois semestres letivos. O objetivo desta estratégia é integrar as duas disciplinas e seus professores.

### 3 A Metodologia de Ensino

Na metodologia proposta, as atividades são divididas em duas categorias: intra e extra-classe. As atividades intra-classe são realizadas em sala de aula e englobam aulas expositivas, incluindo também atividades em classe como exercícios e discussão em grupo. As atividades extra-classe, por sua vez, são aquelas desenvolvidas fora do espaço físico da sala de aula, como forma complementar de ensino. Para realizarem as atividades extra-classe os alunos são organizados em equipes de no máximo quatro alunos e suas execuções contam com o auxílio de monitores. A cada monitor são alocadas três ou quatro equipes.

As atividades realizadas extra-classe têm por objetivo o desenvolvimento de um projeto de software, obtendo como produto final uma aplicação para dispositivos móveis (e.g., celulares, *palm*s, PDAs). O domínio de dispositivos móveis foi selecionado como alvo para o desenvolvimento por tratar-se de uma área da computação que engloba tecnologias recentes, estimulando os alunos para a aplicação dos conceitos apresentados no decorrer da disciplina.

Cada equipe é livre para escolher o modelo de processo de desenvolvimento, as ferramentas de desenvolvimento, a ferramenta de gerência de projeto e as linguagens de programação que mais se adequam às características do projeto e dos membros da equipe.

Os projetos para dispositivos móveis, em geral, envolvem um cliente e um servidor. O cliente pode ser desenvolvido em Java Micro Edition (JME), *Binary Runtime Environment for Wireless* (Brew), SuperWaba, C++, dentre outras, ficando a cargo da equipe selecionar a linguagem que considera mais apropriada. Da mesma forma, a linguagem utilizada no desenvolvimento do servidor é selecionada pela própria equipe (e.g., PHP, JSP/Servlet). A comunicação entre cliente e servidor, por sua vez, pode ser feita por meio das tecnologias *Bluetooth*, *General Packet Radio Service* (GPRS), dentre outras. Além disso, a utilização de padrões de projeto (GAMMA, 1994), bem como o uso de *frameworks* (GAMMA, 1994) é exigida.

Para facilitar o entendimento do domínio e das tecnologias envolvidas, o professor disponibiliza um calendário de aulas extras a serem ministradas pelos monitores e/ou especialistas nas linguagens ou ferramentas utilizadas. Em geral, aulas sobre UML, JME, padrões de projeto, ferramentas de gerência de projeto e *frameworks* são ministradas. Entretanto, de acordo com as necessidades específicas de cada grupo, um acompanhamento mais direcionado é empregado pelos monitores como forma de favorecer o desempenho dos alunos e o desenvolvimento das aplicações em tempo

hábil, uma vez que as mesmas devem ser conduzidas no período de um semestre letivo.

Neste contexto a metodologia de ensino proposta está baseada em cinco características básicas que são descritas a seguir:

1. **Atividades em grupo:** o trabalho em grupo é um dos principais alicerces da metodologia. Os grupos são montados de acordo com o tema selecionado para as aplicações. Cada grupo é composto por quatro integrantes e constitui uma equipe de projeto, que é composto por um líder, que representa o gerente do projeto, e pelos demais integrantes, que representam os demais papéis necessários para o desenvolvimento da aplicação. As atividades de desenvolvimento e gerenciamento do projeto são realizadas pelos alunos, exercitando habilidades como liderança, cooperação e articulação.
2. **Interação com o monitor e com o professor:** a interação dos integrantes do grupo com os monitores e com o professor acontece por meio de reuniões presenciais. Além disso, a disciplina possui uma lista de discussão e um endereço na *web* onde podem ser encontrados: o plano da disciplina com cronograma, uma lista das atividades solicitadas, as notas das avaliações e todas as notas de aula e material didático extra como fonte de pesquisa. Todas as atividades realizadas pelos alunos são acompanhadas pelo professor e pelos monitores, com o objetivo de colaborar com a aprendizagem do grupo e esclarecer dúvidas sobre o conteúdo ou sobre as etapas do processo de desenvolvimento. Além disso, os alunos são orientados com relação às atividades a serem realizadas e sobre a utilização das ferramentas necessárias. Os monitores selecionados são alunos capacitados no conteúdo da disciplina.
3. **Uso de ferramentas:** os alunos são estimulados a utilizarem as ferramentas de gerência de projeto e as ferramentas de desenvolvimento adequadas para a elaboração dos artefatos pertencentes a cada uma das fases do processo de desenvolvimento. As principais ferramentas disponíveis para uso são apresentadas para os alunos em horário complementar (extra-classe) por meio de palestras e seminários realizados por especialistas na área que podem ser monitores e/ou alunos de mestrado e doutorado em Estágio a Docência.
4. **Prática de um modelo de processo de software:** um modelo de processo de software é selecionado por cada equipe de projeto de acordo com a aplicação a ser desenvolvida. No entanto, independente do modelo de processo selecionado, a disciplina possui um conjunto de artefatos básicos que devem ser entregues por todas as equipes nos marcos pré-estabelecidos pelo professor. Esse conjunto de marcos é composto pelos principais artefatos resultantes de cada etapa dos modelos de processo de desenvolvimento existentes na literatura e tem como objetivo controlar o tempo de desenvolvimento das aplicações das equipes e auxiliar nas atividades de gerência da disciplina.
5. **Avaliação equilibrada:** a avaliação é feita por meio de uma média aritmética entre as notas de duas provas e de um projeto de desenvolvimento. As provas buscam avaliar os aspectos teóricos da disciplina, enquanto o projeto tem por objetivo focar na aprendizagem colaborativa entre os membros do grupo, bem como, avaliar a aplicação prática dos conceitos apresentados em sala de aula. Nas provas, os alunos

recebem notas que variam de zero (0) a dez (10), enquanto no projeto, os alunos são avaliados na medida em que os artefatos de cada etapa do modelo de processo de desenvolvimento do projeto são produzidos. A nota para cada artefato é avaliada em dois estágios. No primeiro estágio, uma nota é aplicada pelo monitor a partir da análise dos seguintes elementos: respeito à data de entrega do artefato, uso de ferramenta para geração do artefato, uso de ferramenta para gerência dos recursos e participação da equipe nas reuniões presenciais com os monitores. Em seguida, reuniões são realizadas entre os monitores e o professor com objetivo de repassar a avaliação e o desempenho das equipes. No segundo estágio, a nota atribuída pelo monitor é revisada pelo professor da disciplina. Ao final da disciplina, o projeto final e os artefatos gerados são apresentados pelos alunos em sala de aula. Além dos resultados alcançados em cada uma das etapas, a apresentação inclui a demonstração da aplicação final. Tanto a apresentação quanto a demonstração são avaliadas seguindo o mesmo procedimento dos demais artefatos.

As Tabelas 1, 2, 3, 4, 5 e 6 apresentam o conjunto mínimo de artefatos exigidos e o objetivo de cada um deles, classificados de acordo com as etapas do processo de desenvolvimento.

**Tabela 1. Conjunto de artefatos de Gerência de Projetos**

<i>Gerência de projeto</i>	
<b>Artefato</b>	<b>Objetivo</b>
Plano de Projeto	Estabelece os recursos disponíveis, a estrutura analítica do projeto, os riscos a serem monitorados e os mecanismos de acompanhamento do modelo de processo escolhido
Cronograma	Estabelece as dependências entre as atividades, o prazo estimado necessário para atingir cada marco e a alocação dos recursos nas atividades

**Tabela 2. Conjunto de artefatos de Elicitação e Análise de Requisitos**

<i>Elicitação e Análise de Requisitos</i>	
<b>Artefato</b>	<b>Objetivo</b>
Estudo de Viabilidade	Define se a aplicação contribui para os objetivos gerais da organização e a viabilidade de implementação considerando restrições de tecnologia, custo, prazo e integração com outros sistemas existentes
Especificação de Requisitos	Estabelece o que a aplicação deve implementar. Inclui os requisitos de usuário e uma especificação detalhada dos requisitos do sistema
Especificação de Casos de Uso	Detalha os requisitos da aplicação por meio da descrição de fluxos básicos e fluxos alternativos que descrevem seqüências de eventos de um ator para completar um processo.

**Tabela 3. Conjunto de artefatos de Design**

<i>Design</i>	
<b>Artefato</b>	<b>Objetivo</b>
Especificação de Arquitetura	Apresenta uma visão geral de alto nível da arquitetura prevista do sistema, mostrando a distribuição das funções e componentes nos módulos do sistema.

Projeto da Aplicação	Detalha como as funcionalidades serão fornecidas pelos componentes da aplicação por meio do diagrama de classes, diagramas de seqüência e diagrama de estados
Relatório de Padrões de Software	Identifica os padrões de projeto utilizados durante o desenvolvimento da aplicação

**Tabela 4. Conjunto de artefatos de Implementação**

<i>Implementação</i>	
<b>Artefato</b>	<b>Objetivo</b>
Código	Apresenta o código fonte resultante da implementação da aplicação

**Tabela 5. Conjunto de artefatos de Testes**

<i>Testes</i>	
<b>Artefato</b>	<b>Objetivo</b>
Plano de Testes	Define o nível de cobertura segundo o qual os elementos mais críticos do software serão testados com prioridade e com um nível de cobertura mais amplo (BASTOS <i>et al.</i> ,2007)
Relatório de Testes	Registra os defeitos encontrados durante a realização das atividades de testes (BASTOS <i>et al.</i> ,2007)

**Tabela 6. Conjunto de artefatos de Encerramento**

<i>Encerramento</i>	
<b>Artefato</b>	<b>Objetivo</b>
Apresentação	Resume em uma apresentação aberta para s demais alunos os resultados obtidos com o desenvolvimento do projeto

## 4 Avaliação do Desempenho com a Metodologia Proposta

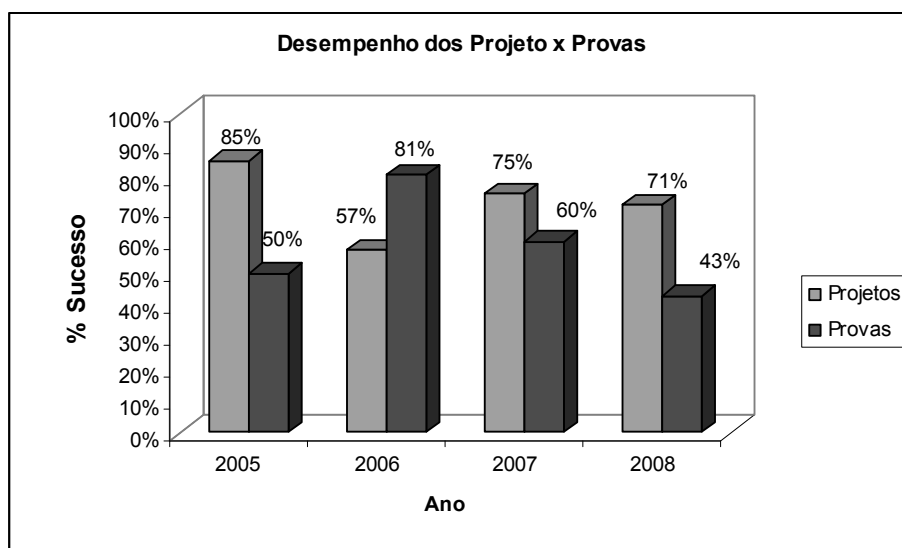
Para validar o uso e avaliar os resultados obtidos, a metodologia de ensino apresentada neste artigo vem sendo aplicada na disciplina de ES do Departamento de Computação da UFC desde 2005. A disciplina de ES é obrigatória e é ofertada anualmente para o Curso de Computação. Ela possui quatro créditos, totalizando sessenta horas-aula por semestre e as turmas têm em média quarenta alunos.

Nos quatro períodos letivos em que a metodologia de ensino foi aplicada, aproximadamente, quarenta projetos foram realizados, abrangendo um total de cento e sessenta alunos. No ano de 2005, somente alunos da graduação cursavam a disciplina de ES, no entanto, a partir de 2006, alunos dos cursos de mestrado e doutorado também passaram a cursar da disciplina.

A metodologia adotada no ano de 2005 contou com um único monitor para dar suporte a todas as equipes e com várias ferramentas de gerência de projeto, pois ficou a critério de cada equipe escolher uma ferramenta de gerência de projeto para acompanhar as suas atividades. Esta estratégia dificultou um acompanhamento criterioso das equipes por parte do monitor e do professor, aumentando, entretanto as notas atribuídas aos projetos, devido à falta de um maior rigor no controle das atividades desempenhadas pelas equipes. Com base no gráfico apresentado na Figura

1, observa-se que no ano de 2005 o desempenho nos projetos foi 35% superior ao desempenho nas provas.

Visando um acompanhamento mais rigoroso dos trabalhos das equipes por parte do monitor e do professor, no ano de 2006 a metodologia foi refinada. O diferencial foi a adoção de uma única ferramenta de gerência de projetos para todas as equipes denominada *dotProject* (DOTPROJECT, 2008). No entanto, as equipes continuavam livres para escolher o modelo de processo de desenvolvimento, as ferramentas de desenvolvimento e tecnologias que julgassem mais adequadas para as características do projeto e dos membros da equipe.



**Figura 1. Análise do desempenho dos alunos nos projetos e provas**

A ferramenta *dotProject* foi selecionada por tratar-se de um gerenciador de projetos web que pode ser acessada via Internet, propiciando o acompanhamento on-line das atividades, reduzindo o esforço necessário para acompanhar o andamento dos vários projetos. Nesse cenário, conforme pode ser verificado na Figura 1 houve uma inversão dos dados em relação ao ano anterior. O desempenho dos projetos foi 24% inferior ao das provas. Em compensação, os estudantes foram mais bem assessorados pelo monitor e professor extra-classe, o que resultou em um maior aprendizado.

No ano de 2007 a estratégia foi mantida. Analisando os dados apresentados no gráfico da Figura 1, pode-se perceber que o desempenho geral dos alunos, envolvendo provas e projetos, ficou mais equilibrado, tendendo ao nivelamento entre o desempenho individual nas provas e o desempenho coletivo no projeto. Entretanto, neste período percebeu-se que o uso de um único monitor para acompanhar todas as equipes não era suficiente para atender às necessidades específicas de aprendizado de todos os alunos, portanto, um atendimento extra-classe mais direcionado tornava-se necessário.

Com o objetivo de suprir não só as necessidades de execução dos projetos, como também refletir uma melhoria no desempenho dos alunos, em 2008 a metodologia foi refinada novamente. Nesse cenário, juntamente ao monitor, foi acrescentada uma equipe de assistentes de ensino composta por três alunos de pós-graduação em Estágio a Docência.

A cada integrante da nova equipe da monitoria foram alocadas apenas três equipes, possibilitando um suporte mais próximo aos alunos durante as atividades práticas.

Com isso, um maior rigor na condução e controle de execução dos projetos foi adicionado. Em uma análise do gráfico mostrado na Figura 1, pode-se verificar que o desempenho nos projetos se manteve satisfatório, porém, o desempenho nas provas diminuiu. A causa identificada para a diminuição do rendimento dos alunos nas provas foi o aumento do grau de dificuldade das mesmas, necessário devido a qualidade do suporte extra-classe dado aos mesmos por parte da monitoria.

## 5 Conclusões e Trabalhos Futuros

De acordo com os resultados obtidos após a implantação da metodologia de ensino proposta podemos concluir que os alunos da disciplina de Engenharia de Software do Departamento de Computação da Universidade Federal do Ceará que utilizaram a metodologia de ensino demonstraram, ao final da disciplina, um preparo e conhecimento maior das práticas de Engenharia de Software.

Com um enfoque maior na gerência dos projetos e utilizando os conceitos na prática, a metodologia permite um aprendizado mais consistente do uso correto das técnicas teóricas e da importância de praticar um modelo de processo de software com todas as etapas incluídas no mesmo durante o desenvolvimento das aplicações.

O sentimento positivo demonstrado pelos alunos, ao final dos projetos, foi fator decisivo para que a metodologia continue sendo utilizada nos semestres seguintes de modo que possa ser continuamente melhorada e refinada. Além disso, como os resultados apresentados neste trabalho não foram comparados com as turmas anteriores a 2005, em que a metodologia ainda não era utilizada, pretende-se fazer essa avaliação como forma de validar a metodologia apresentada neste trabalho.

## 6 Agradecimentos

Agradecemos a colaboração dos alunos de graduação Bruno Sabóia Aragão e Luciana Paiva na elaboração dos modelos dos artefatos utilizados na disciplina. Agradecemos ainda a colaboração da aluna de Doutorado Maria de Fátima Costa Souza pelo auxílio na análise pedagógica dos dados obtidos com a metodologia e também à professora Dra. Bernadette Farias Lóscio, pelas informações dos resultados obtidos com o uso da metodologia no ano de 2007.

## Referências

AGUIAR, Y. P. C., Lima, A. H. G., Leite, F. L. J., et. al. **easYProcess: Um Processo de Desenvolvimento para Uso no Ambiente Acadêmico**. In: XII Workshop de Educação em Informática - XXIV Congresso da Sociedade Brasileira de Computação, Salvador. 2004.

ALVES, A. G., BENITTI, F.B.V. **Processo de Desenvolvimento Integrando Disciplinas de Engenharia de Software** In: XIV Workshop sobre Educação em Computação - XXVI Congresso da Sociedade Brasileira de Computação, Campo Grande. 2006.

BASTOS, A., RIOS, E., CRISTALLI, R., MOREIRA, T. **Base de conhecimento em teste de software**. 2ª. Edição, São Paulo. Editora Martins Fontes, 2007.

DEMO, P. A nova **LDB: Ranços e avanços**. Campinas: Papyrus, 1997. 9ª ed.



**DOTPROJECT. The home of dotProject - the Open Source Project Management tool.**  
Disponível em: <<http://www.dotproject.net>>. Acesso em: 05 ago. 2008.

**FIorentini, L. M. R. Materiais didáticos escritos nos processos formativos a distância.** In: CONGRESSO DE ENSINO SUPERIOR A DISTÂNCIA, I, 2002. Petrópolis. Anais. Petrópolis: EsuD, 2002.

**GAMMA, E., HELM, R., JOHNSON, R., and VLISSIDES, J. Design Patterns, Elements of Object Oriented Software,** Addison Wesley, Reading, MA, 1994.

**PIAGET, J. Psicologia e Pedagogia.** Rio de Janeiro: Forense-Universitária, 1975. 3ª edição.

**PINHEIRO, B. M. e GONÇALVES, M. H. O Processo Ensino-Aprendizagem.** Rio de Janeiro: Editora SENAC Nacional, 2001.

**SOMMERVILLE, I., Engenharia de Software,** 8ª ed.SP: Pearson, 2007.