

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**ESCOLLAB: UMA METODOLOGIA COLABORATIVA VOLTADA
PARA O ENSINO DE ENGENHARIA DE *SOFTWARE***

SEVERINO DO RAMO DE PAIVA

JOÃO PESSOA - PB
ABRIL – 2012

SEVERINO DO RAMO DE PAIVA

ESCollab: Uma Metodologia Colaborativa voltada para o Ensino de Engenharia de *Software*

Dissertação apresentada ao Centro de Informática da Universidade Federal da Paraíba, como requisito parcial para obtenção do título de Mestre em Informática (Sistemas de Computação).

Orientador: Prof. Dr. Álvaro Francisco de Castro Medeiros

JOÃO PESSOA - PB
ABRIL – 2012

*P149e Paiva, Severino do Ramo de.
ESCollab: uma metodologia colaborativa
voltada para o ensino de engenharia de software /
Severino do Ramo de Paiva.-- João Pessoa, 2012.
114f. : il.
Orientador: Álvaro Francisco de Castro
Medeiros
Dissertação (Mestrado) – UFPB/CCEN
1. Informática. 2. Engenharia de Software. 3.
Colaboração. 4. Metodologia de ensino. 5. Redes
Sociais.*

1
2

Ata da Sessão Pública de Defesa de Dissertação de Mestrado de **SEVERINO DO RAMO DE PAIVA**, candidato ao Título de Mestre em Informática na Área de Sistemas de Computação, realizada em 23 de abril de 2012.

3
4

5 Ao vigésimo terceiro dia do mês de abril do ano dois mil e doze, às dez horas, no Auditório
6 do CCEN - da Universidade Federal da Paraíba, reuniram-se os membros da Banca
7 Examinadora constituída para examinar o candidato ao grau de Mestre em Informática, na
8 área de "Sistemas de Computação", na linha de pesquisa "Computação Distribuída", o Sr.
9 **SEVERINO DO RAMO DE PAIVA**. A comissão examinadora foi composta pelos
10 professores doutores: **ÁLVARO FRANCISCO DE CASTRO MEDEIROS** (PPGI-UFPB),
11 Orientador e Presidente da Banca Examinadora, **DANIELLE ROUSY DIAS DA SILVA**
12 (DI-UFPB), como examinadora interna e **JOBERTO SÉRGIO BARBOSA MARTINS**
13 (UNIFACS), como examinador externo. Dando início aos trabalhos, o professor **ÁLVARO**
14 **FRANCISCO DE CASTRO MEDEIROS**, cumprimentou os presentes, comunicou aos
15 mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse,
16 oralmente, a exposição do trabalho de dissertação intitulado "ESCOLLAB: UMA
17 METODOLOGIA COLABORATIVA VOLTADA PARA O ENSINO DE ENGENHARIA
18 DE SOFTWARE". Concluída a exposição, o candidato foi argüido pela Banca
19 Examinadora que emitiu o seguinte parecer: "aprovado". Assim sendo, deve a
20 Universidade Federal da Paraíba expedir o respectivo diploma de Mestre em Informática na
21 forma da lei e, para constar, eu, professor Alisson Vasconcelos de Brito, Vice-
22 coordenador deste Programa, servindo de secretário, lavrei a presente ata que vai assinada
23 por mim mesmo e pelos membros da Banca Examinadora. João Pessoa, 23 de abril de
24 2012.
25

26

Alisson Vasconcelos de Brito

Prof. Dr. Álvaro Francisco De Castro Medeiros
Orientador (PPGI-UFPB)

Profa. Dra. Danielle Rousy Dias Da Silva
Examinadora Interna (DI-UFPB)

Prof. Dr. Joberto Sérgio Barbosa Martins
Examinador Externo (UNIFACS)

27



Dedico este trabalho à minha família e a todos
que contribuíram para sua materialização.

AGRADECIMENTOS

A Deus pelo dom maior da vida.

À minha família pela força nos momentos de dificuldades.

Ao meu orientador, Professor Álvaro Francisco de Castro Medeiros, pela amizade, competência e profissionalismo.

Aos amigos pelas contribuições.

A todos que de alguma forma estiveram presentes e me auxiliaram na elaboração deste trabalho.

RESUMO

Dentre todas as atividades desenvolvidas pelos seres humanos na atualidade, o desenvolvimento de *software* é uma das mais recentes em termos históricos, mas uma das mais importantes devido à crescente informatização da sociedade e seu *modus vivendi*. Por outro lado, a importância crescente dos ambientes colaborativos, no mundo globalizado, tem possibilitado a materialização da “empresa global”, o compartilhamento de informações e viabilizado a interação entre pessoas distantes. Essas mudanças têm potencializado o poder do trabalho em equipe e a interação social. Essa importância social e econômica da indústria do *software* tem motivado vários pesquisadores a buscarem soluções para o aperfeiçoamento da formação de mão de obra especializada na área de Engenharia de *Software*. Este trabalho apresenta uma metodologia de ensino colaborativa voltada para o aperfeiçoamento do ensino das disciplinas da área de Engenharia de *Software*. A metodologia ESCollab incorpora as mais modernas ferramentas pedagógicas, tendo sido aplicada, a título de estudo de caso, em duas instituições de ensino superior na área de computação.

Palavras-chave: Colaboração, Cooperação, Engenharia de *Software*, Sistemas Colaborativos, Metodologia de Ensino, Redes Sociais.

ABSTRACT

Among all the activities performed by human beings today, *software* development is one of the most recent in historical terms, but one of the most important due to the increasing digitalization of the society. On the other hand, the growing importance of collaborative environments, in a globalized world, has enabled the sharing of information and the interaction between people far away. These changes have enhanced the power of teamwork and social interaction. The social and economic importance of the *software* industry has motivated many researchers to seek solutions to improve the training of skilled labor in the area of *Software* Engineering. This dissertation work presents a collaborative teaching methodology aimed at improving the teaching of the disciplines of *Software* Engineering. The ESCollab methodology incorporates the latest teaching tools and has been applied, as a case study, in two institutions of higher education in computing.

Keywords: Collaboration, Cooperation, *Software* Engineering, Collaborative Systems, Teaching Methodology, Social Networks.

LISTA DE FIGURAS

Figura 2.1: Evolução do número de matrículas no período 2001-2009.....	22
Figura 2.2– Distribuição dos cursos da área de computação 2007-2008	24
Figura 2.3 – Evolução das matrículas dos cursos da área de computação 2007-2008.....	24
Figura 2.4 – Evolução dos concluintes dos cursos da área de computação 2007-2008	25
Figura 2.5 – Guia do corpo de conhecimento em Engenharia de <i>Software</i>	30
Figura 3.1 – Espiral do conhecimento.....	38
Figura 3.2– Hierarquia dos 8 tipos de aprendizagens	40
Figura 3.3 – A comunicação como processo de seleção, combinação e intercâmbio de repertórios.	43
Figura 3.4 – Modelo do processo de comunicação	43
Figura 3.5 – Modelo de interação de Moore	45
Figura 3.6 – Modelo simplificado de compartilhamento do conhecimento	48
Figura 4.1: Modelo de colaboração 3C	56
Figura 4.2: Postura Cooperativa	57
Figura 4.3: Metodologia ESCollab e suas fases	60
Figura 4.4: O processo ensino-aprendizagem sob a ótica da Metodologia ESCollab.....	63
Figura 4.5: Protocolo de funcionamento da Metodologia ESCollab	64
Figura 4.6: Representação do funcionamento da Metodologia ESCollab - Notação BPMN	65
Figura 4.7: Resultado Final da Disciplina Engenharia de <i>Software</i> – IESA – 2011.1	69
Figura 4.8: Resultado Final da Disciplina Qualidade de <i>Software</i> – IESA – 2011.2.....	70
Figura 4.9: Resultado Final da Disciplina Sist Inform Inteligentes – IESA – 2011.2	70
Figura 4.10: Resultado Final da Disciplina Desenv. Sist. Distribuídos – IESA – 2011.2.....	71

LISTA DE QUADROS

Quadro 2.1– Evolução do número de instituições de ensino superior no período 2004-2009	21
Quadro 2.2 – Evolução do número de cursos superiores no período 2004-2009.....	22
Quadro 2.3 – Os dez maiores cursos de graduação do Brasil em termos de matrículas em 2009	23
Quadro 2.4 – Distribuição dos cursos da área de computação 2007-2008	23
Quadro 2.5 – Orientação quanto ao Ensino de Engenharia de <i>Software</i>	27
Quadro 3.1 – Teorias da Aprendizagem e suas características	39
Quadro 3.2: Lista de ferramental didático	41
Quadro 3.3: Lista das funcionalidades Web requeridas categorizadas por tipo de interação.....	47
Quadro 3.4 – Principais ambientes de desenvolvimento colaborativo	51
Quadro 3.5 – Principais redes sociais e suas características	52
Quadro 4.1 – Comparação entre a aprendizagem tradicional e a colaborativa.....	58

LISTA DE SIGLAS

ACM	Association for Computing Machinery
ASPER	Associação Paraibana de Ensino Renovado
DEED	Diretoria das Estatística Educacionais do INEP
EaD	Ensino a Distância
ENADE	Exame Nacional de Desempenho dos Estudantes
ES	Engenharia de <i>Software</i>
EUA	Estados Unidos da América
FAQ	<i>Frequently Asked Questions</i> (Perguntas Mais Frequentes)
IBGE	Instituto Brasileiro de Geografia e Estatística
IEEE	Institute of Electrical and Electronics Engineers
IES	Instituição de Ensino Superior
IESA	Instituição de Ensino Superior “A” do Estudo de Caso
IESB	Instituição de Ensino Superior “B” do Estudo de Caso
IFPB	Instituto Federal de Ciência, Educação e Tecnologia da PB
IFRN	Instituto Federal de Ciência, Educação e Tecnologia do RN
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais
LDB	Lei de Diretrizes e Bases da Educação
MEC	Ministério da Educação e Cultura
MSN	Programa de Mensagens On-line
OCDE	Organização para Cooperação e Desenvolvimento Econômico
PBL	<i>Problem Based Learning</i> (Aprendizagem Baseada em Problemas)
PISA	Programme for International Student Assessment
RD	Recurso Didático
SBC	Sociedade Brasileira de Computação
TIC	Tecnologia da Informação e Comunicação
UFMG	Universidade Federal de Campina Grande
UFF	Universidade Federal Fluminense
UFPA	Universidade Federal do Pará
UFPB	Universidade Federal da Paraíba
UFPI	Universidade Federal do Piauí
UNIPE	Centro Universitário de João Pessoa

SUMÁRIO

1.1 INTRODUÇÃO.....	14
1.2 OBJETIVOS.....	16
1.2.1 Geral.....	16
1.2.2 Específicos.....	16
1.3 PROBLEMATIZAÇÃO	17
1.4 HIPÓTESE	17
1.5 RELEVÂNCIA/MOTIVAÇÃO.....	18
1.6 A METODOLOGIA.....	19
1.7 DESENVOLVIMENTO DO TRABALHO	19
CAPÍTULO II.....	20
A EDUCAÇÃO SUPERIOR E O ENSINO DE ENGENHARIA DE <i>SOFTWARE</i>	20
2.1 INTRODUÇÃO.....	20
2.1 A SITUAÇÃO DA EDUCAÇÃO SUPERIOR NO BRASIL	21
2.1.1 A educação superior na área de computação e informática.....	23
2.2 O ENSINO DE ENGENHARIA DE <i>SOFTWARE</i>	26
2.2.1 Trabalhos relacionados	30
2.2.2 Pesquisa de opinião sobre o ensino de engenharia de <i>software</i>	30
2.2.3 Problemas com o Ensino de Engenharia de <i>Software</i>	35
CAPÍTULO III.....	37
FUNDAMENTOS DE ENSINO E APRENDIZAGEM.....	37
3.1 INTRODUÇÃO.....	37
3.2 A IMPORTÂNCIA DA COMUNICAÇÃO NO PROCESSO ENSINO- APRENDIZAGEM	42
3.3 A APRENDIZAGEM COLABORATIVA	47

3.4 REDES SOCIAIS.....	52
CAPÍTULO IV	54
METODOLOGIA <i>ESCOLLAB</i>.....	54
4.1 INTRODUÇÃO.....	54
4.2 ESPECIFICAÇÃO DA METODOLOGIA <i>ESCOLLAB</i>	59
4.2.1 Definições da Metodologia <i>ESCollab</i> :.....	59
CONCLUSÕES E TRABALHOS FUTUROS	72
REFERÊNCIAS	74
APÊNDICE A: Revisão Sistemática das Pesquisas Realizadas Sobre a Melhoria do Ensino da Engenharia de Software	81
APÊNDICE B: Pesquisa de opinião sobre o ensino de Engenharia de Software	96

CAPÍTULO I

CONSIDERAÇÕES INICIAIS AO ESTUDO REALIZADO

1.1 INTRODUÇÃO

Dentre as várias atividades humanas atuais, o desenvolvimento de *software* é uma das mais recentes, contando apenas com cerca de seis décadas de existência. Adicionalmente, registre-se a verdadeira informática-dependência que a humanidade vive no seu atual momento histórico. As aplicações de informática fazem parte de todas as atividades humanas.

Esse cenário da relevância do *software* e da sua indústria faz com que haja nos meios empresariais e acadêmicos uma preocupação com a qualidade da formação de quem desenvolve as aplicações que dão sustentação ao *modus vivendi* da sociedade contemporânea.

Essa importância do *software* no mundo atual fez surgir uma preocupação séria com a qualidade do processo de desenvolvimento de *software*, levando à criação da chamada Engenharia de *Software*. “Engenharia de *Software* (ES) é a aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do *software*” (PRESSMAN, 2010). Por outro lado, Sommerville (2007), define ES como “uma disciplina de engenharia relacionada com todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até sua manutenção, depois que este entrar em operação”.

Outra forma de visualizar a importância da ES é sob o ponto de vista financeiro. No âmbito nacional, a indústria do *software* tem evoluído de forma substancial nas últimas décadas, mas ainda se tem uma balança comercial deficitária nesse segmento, atingindo aproximadamente US\$ 5 bilhões de dólares de déficit anuais (IBGE, 2009). Ainda segundo o IBGE (2009), “a receita de exportação do Brasil pode ser considerada baixa em comparação com a da Índia (maior exportador mundial, com US\$ 50 bilhões), Alemanha (US\$ 17,9 bilhões), EUA (US\$ 13,4 bilhões), Reino Unido (US\$ 13,0 bilhões), Israel (US\$ 6,8 bilhões),

China (US\$ 6,2 bilhões) e Canadá (US\$ 5,0 bilhões). As exportações brasileiras ficaram próximas das do Japão (US\$ 1,0 bilhão)”.

Adicionalmente, a pesquisa de qualidade no setor de *Software* Brasileiro 2009 (MCT, 2010), aponta deficiências na qualidade do processo de *software* por parte do parque nacional de empresas da área de *software*, explicitando a necessidade de investimentos para que se melhore a qualidade e a competitividade do *software* nacional no mercado mundial, passando necessariamente pela melhoria na formação dos nossos engenheiros de *software*.

Outrossim, registre-se que, o número de pessoas que utilizam a *internet* cresce de forma tão rápida que, em 20 de fevereiro de 2011, correspondia a mais de 28,7% da população mundial¹. Entretanto, o número de usuários que utilizam a rede mundial de computadores é mais uma indicação superficial. O que mais mudou não foi a quantidade de internautas e sim o que eles podem realizar com e pela *internet* (PISANI; PIOTET, 2010).

Em função dos avanços significativos alcançados pela educação superior no Brasil nos últimos anos, mais especificamente na área de computação e informática, principalmente no aspecto quantitativo, propor novas abordagens metodológicas e desenvolver ferramentas que possam propiciar a evolução da qualidade da formação dos futuros engenheiros de *software* e, conseqüentemente, a competitividade do *software* nacional, poderá ser uma valiosa contribuição para o desenvolvimento econômico, educacional e social do nosso país.

¹ Disponível em <http://www.internetworldstats.com/stats.htm>, acesso em: 20 Fev. 2011.

1.2 OBJETIVOS

1.2.1 Geral

Propor uma metodologia de ensino colaborativo voltada para a área de engenharia de *software*. A metodologia ESCollab proposta terá ênfase na cooperação e na colaboração entre docentes e estudantes na área de ES no ensino de duas ou mais disciplinas de uma grade curricular de um curso superior da área de Computação. Esta metodologia utilizará conceitos da psicologia cognitiva, de sistemas colaborativos e da teoria da comunicação.

1.2.2 Específicos

Com este objetivo, é proposta a materialização dos seguintes objetivos específicos:

- a) Formatar uma metodologia de ensino colaborativa voltada para o ensino-aprendizagem de engenharia de *software*;
- b) Disponibilizar a metodologia para ser utilizada e realimentada por profissionais, alunos e docentes das instituições envolvidas na pesquisa e demais instituições de ensino do país;
- c) Realizar estudo de caso com instituição de ensino superior da cidade de João Pessoa, envolvendo duas disciplinas (D1 e D2) da grade curricular de um curso superior na área de computação e informática, envolvendo dois professores (P1 e P2), que necessitam colaborar no sentido operacional quanto na abordagem semântica com relação ao estabelecimento de conteúdos programáticos (C1 e C2) que têm relacionamento de pré-requisitos e/ou interseção de assuntos.

Para tanto, foram estudados os seguintes tópicos:

- a) A mudança de paradigmas na educação privilegiando-se o desenvolvimento de competências e atual situação do ensino superior na área de Engenharia de *Software*;
- b) As ferramentas tecnológicas que podem propiciar ganhos na interação entre os agentes envolvidos no processo educacional;
- c) Os conceitos básicos da psicologia cognitiva e da tecnologia da comunicação, que podem ajudar introduzir avanços no ensino de ES.

1.3 PROBLEMATIZAÇÃO

Sem o intuito de tecer meras críticas ao trabalho estruturado em centenas de instituições, cerca de 2000² cursos em funcionamento efetivo na área de Computação e Informática, atendendo cerca de 300³ mil alunos regularmente matriculados, pode-se dizer que, apesar de todos os avanços das últimas décadas na área de engenharia de *software*, observa-se, com base em Paiva (2011a; 2011b), que os docentes enfrentam alguns problemas no seu trabalho cotidiano nesta área, entre eles:

- a) Dificuldades em motivar os alunos quanto ao envolvimento efetivo no processo ensino-aprendizagem;
- b) Aulas excessivamente teóricas e a carência de abordagens práticas que contribuam para a absorção mais eficiente dos conteúdos ministrados;
- c) Falta de domínio das técnicas básicas e ferramentas didáticas por parte de uma parcela significativa dos docentes de ES, trabalhando, na maioria das vezes, na base da tentativa e erro;
- d) Ausência de uma postura colaborativa na condução das disciplinas da área;
- e) Limitações quanto à interação professor-aluno fora da sala de aula, notadamente nas instituições privadas, onde o regime de trabalho dominante é horista⁴;
- f) Dificuldades de fazer a supervisão pedagógica do trabalho docente, principalmente, nas instituições públicas;
- g) Resultados do desenvolvimento dos discentes não satisfatórios em função dos problemas elencados anteriormente.

1.4 HIPÓTESE

Para atingir o objetivo geral, ou seja, de definir uma metodologia de ensino que possa introduzir melhorias no processo ensino-aprendizagem na área de ES, acredita-se que a comunicação eficiente, o domínio de ferramentas didáticas apropriadas, um maior rigor metodológico e a colaboração efetiva entre os agentes do processo ensino-aprendizagem em engenharia de *software* podem potencializar melhorias significativas na aprendizagem nessa área.

² MEC – Censo da Educação Superior 2009

³ MEC – Censo da Educação Superior 2009

⁴ Os docentes são remunerados em função da sua carga horária semanal de aulas

1.5 RELEVÂNCIA/MOTIVAÇÃO

A relevância deste trabalho está explicitada nos seguintes pontos:

- a) O significativo número de pesquisas científicas apontando os problemas e sinalizando soluções para a problemática do ensino-aprendizagem em engenharia de *software*, conforme aponta (PAIVA, 2011a);
- b) A efervescência da Web 2.0, das redes sociais e a crescente utilização dessas ferramentas no processo educacional;
- c) A reformulação das Leis de Diretrizes e Bases (LDB) da Educação, principalmente a profissional, pelo Ministério da Educação e Cultura (MEC) com foco em desenvolvimento de competências;
- d) O significativo *déficit* comercial brasileiro na área de *software* e o parco volume de exportações da indústria brasileira na área (IBGE, 2009);
- e) As inúmeras contribuições que poderão surgir e ajudar os gestores educacionais, docentes e profissionais a refletirem e, se possível, incorporarem melhorias no processo de formação profissional da área.

Outro aspecto motivacional deste trabalho é o de focar no aspecto humano da aprendizagem e na construção colaborativa do conhecimento ancorada nas tecnologias da informação e comunicação (TIC's) e na psicologia cognitiva. Sua aplicação pode ser efetiva na educação em engenharia de *software* presencial, à distância, corporativa e profissional no modelo de competências, caracterizando seu escopo.

1.6 A METODOLOGIA

Esta pesquisa científica utilizou uma abordagem indutiva baseada na observação e experimentação, que compreendeu as seguintes etapas:

- a) A constatação do problema (descrito no Capítulo I);
- b) Investigações bibliográficas e de campo nas áreas da educação, Ciência da Computação e da Psicologia Cognitiva, descritas nos Capítulos II e III;
- c) A formulação da metodologia, utilizando ferramentas da informática e da tecnologia educacional, descrita no Capítulo IV;
- d) O uso de pré-teste com a aplicação da metodologia em cursos superiores da área de informática na cidade de João Pessoa, descrito no Capítulo IV.

1.7 DESENVOLVIMENTO DO TRABALHO

Nas próximas seções, apresenta-se uma abordagem colaborativa para o ensino das disciplinas ligadas à grande área de engenharia de *software*, através da Web 2.0, baseada em uma metodologia própria de ensino-aprendizagem.

No Capítulo 2, apresenta-se um panorama sobre a educação superior e o ensino de engenharia de *software* no Brasil.

No Capítulo 3, aborda-se a fundamentação pedagógica, psicológica e tecnológica da abordagem proposta.

No Capítulo 4, descreve-se a metodologia e detalha-se a aplicação da mesma. Na descrição da aplicação da metodologia, são explicadas suas etapas, mostrados exemplos, realizadas considerações, exibidos resultados parciais e feitas colocações que justificam os procedimentos utilizados. Nesse mesmo capítulo, observam-se os resultados obtidos com a aplicação da metodologia em uma Instituição de Ensino Superior (IES) e as dificuldades encontradas.

Por fim, no Capítulo 5, são relatadas as conclusões, incluindo a delimitação do trabalho e as propostas de pesquisas futuras a serem desenvolvidas.

CAPÍTULO II

A EDUCAÇÃO SUPERIOR E O ENSINO DE ENGENHARIA DE SOFTWARE

2.1 INTRODUÇÃO

O verbo educar é derivado da palavra latina *educere*, que significa literalmente “ajudar a levantar”. Ao longo dos anos, a educação transformou-se em um diferencial competitivo entre as nações, sendo possível estabelecer comparações entre os países com base nos números de patentes registradas, na quantidade de doutores, no índice de trabalhos publicados, nos resultados nas provas do PISA⁵ da OCDE⁶, percentual da população com diploma universitário, dentre outros critérios.

Francis Bacon (1620), afirmou que “a ciência é poder e que o homem pode o quanto conhece”. Nesse mesmo sentido, Paviani (2005, p. 72) afirma “a meta da ciência é a de dominar a natureza e instaurar o reino do homem sobre as coisas. O conhecimento e o avanço tecnológico são modos de dominar a natureza e o próprio homem. Basta observar o poderio científico e tecnológico dos países ricos em relação aos países pobres ou a relação com o poder, de um modo direto ou indireto, que os homens mais instruídos possuem”.

Modernamente, a ideia de uma economia baseada no conhecimento tem suas raízes no trabalho desenvolvido por um grupo de intelectuais, futurologistas e economistas da informação, da década de 1960, incluindo Fritz Machlup (1962), Peter Drucker (1969) e o bem conhecido Daniel Bell (1973) (PEREIRA; CARVALHO; PORTO, 2009, p.79).

Um importante relatório elaborado pela OCDE (OCDE, 2005) apresenta uma análise do trabalho docente em 25 países. As principais conclusões foram:

- a) Escassez de professores;

⁵ PISA – Programme for International Student Assessment – Programa da OCDE para avaliação internacional e alunos do ensino médio

⁶ OCDE – Organização para Cooperação e Desenvolvimento Econômico – www.oecd.org

- b) Falta de qualificação apropriada;
- c) Defasagem salarial;
- d) Baixo reconhecimento público.

Inegavelmente, na atualidade, vivencia-se um cenário de mudança, de desafios e de enaltecimento da educação e dos docentes, no mínimo, no discurso da classe dirigente dos destinos do mundo. Na próxima seção, é traçado um panorama da educação superior nacional.

2.1 A SITUAÇÃO DA EDUCAÇÃO SUPERIOR NO BRASIL

Nos últimos anos, o Brasil tem alcançado um crescimento significativo do número de instituições de ensino superior, do número de cursos superiores e do número de alunos matriculados, conforme se pode ver através do Censo da Educação Superior do Ministério da Educação. O Quadro 2.1 mostra a evolução do número de instituições no período de 2004 a 2009:

Ano	Total	Universidades	%	Centros Universitários	%	Faculdades	%	Institutos Federais e CEFETs	%
2004	2.013	169	8,4	107	5,3	1.703	84,6	34	1,7
2005	2.165	176	8,1	114	5,3	1.842	85,1	33	1,5
2006	2.270	178	7,8	119	5,2	1.940	85,5	33	1,5
2007	2.281	183	8,0	120	5,3	1.945	85,3	33	1,4
2008	2.252	183	8,1	124	5,5	1.911	84,9	34	1,5
2009	2.314	186	8,0	127	5,5	1.966	85,0	35	1,5

Fonte: Censo da Educação Superior / DEED / MEC / INEP

Quadro 2.1– Evolução do número de instituições de ensino superior no período 2004-2009
Fonte: Adaptado do MEC, 2010

Observa-se também uma evolução ainda mais significativa no número de cursos ofertados por essas instituições, conforme demonstra o Quadro 2.2:

Ano	Total	Universidades	%	Centros Universitários	%	Faculdades	%	Institutos Federais e CEFETs	%
2004	18.644	10.475	56,2	2.134	11,4	5.710	30,6	325	1,74
2005	20.407	10.892	53,4	2.542	12,5	6.699	32,8	274	1,34
2006	22.101	11.552	52,3	2.717	12,3	7.541	34,1	291	1,32
2007	23.488	11.936	50,8	2.880	12,3	8.331	35,5	341	1,45
2008	24.719	12.351	50,0	3.238	13,1	8.725	35,3	405	1,64
2009	27.827	13.865	49,8	3.580	12,9	9.897	35,6	485	1,74

Fonte: Censo da Educação Superior / DEED / MEC / INEP

Quadro 2.2 – Evolução do número de cursos superiores no período 2004-2009

Fonte: Adaptado do MEC, 2010

O número de matrículas oscila em torno dos seis milhões de alunos, contabilizando os alunos das modalidades presenciais e de ensino à distância – EAD. A Figura 2.1 mostra essa evolução positiva de quase 100% no número de alunos matriculados nos últimos oito anos, representando um grande avanço em termos quantitativos:

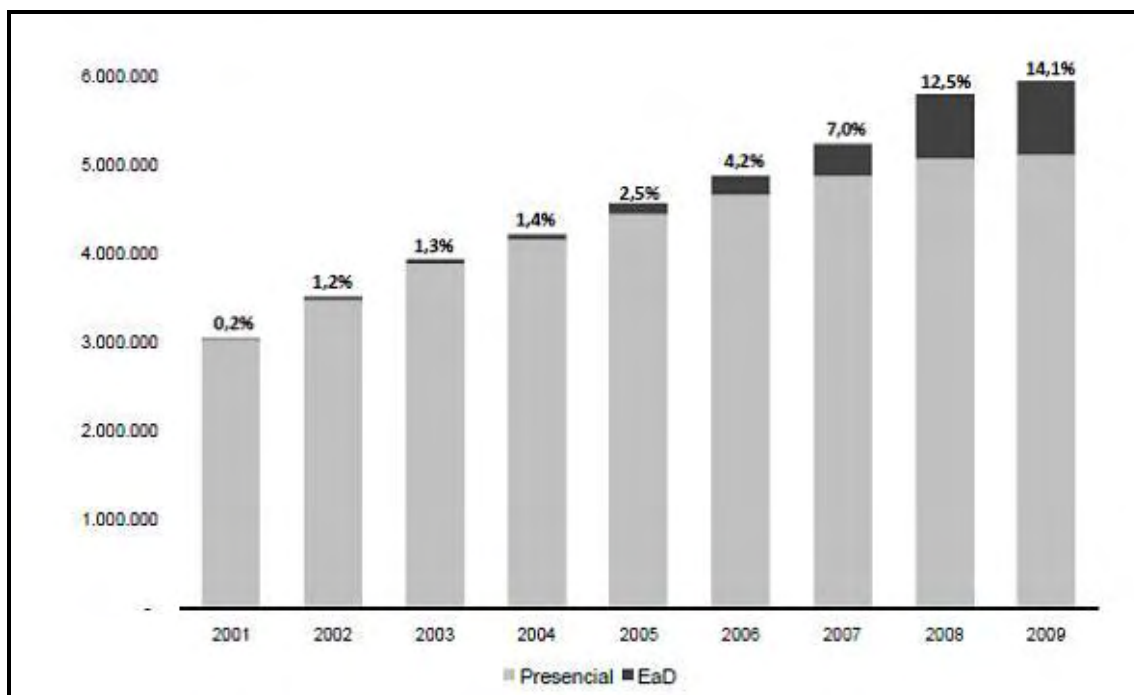


Figura 2.1: Evolução do número de matrículas no período 2001-2009

Fonte: Adaptado do MEC, 2010

A distribuição dessas matrículas pode ser detalhada no Quadro 2.3, onde são apresentados os dez maiores cursos de graduação do Brasil em 2009:

Graduação: Presencial e a distância		
Curso	Matrículas	%
Total	5.954.021	100
1 Administração	1.102.579	18,5
2 Direito	651.730	10,9
3 Pedagogia	573.898	9,6
4 Engenharia	420.578	7,1
5 Enfermagem	235.804	4,0
6 Ciências Contábeis	235.274	4,0
7 Comunicação Social	221.211	3,7
8 Letras	194.990	3,3
9 Educação Física	165.848	2,8
10 Ciências Biológicas	152.830	2,6
Outros Cursos	1.999.279	33,6

Presencial		
Curso	Matrículas	%
Total	5.115.896	100
1 Administração	874.076	17,1
2 Direito	651.600	12,7
3 Engenharia	419.397	8,2
4 Pedagogia	287.127	5,6
5 Enfermagem	235.281	4,6
6 Comunicação Social	205.409	4,0
7 Ciências Contábeis	205.330	4,0
8 Educação Física	163.528	3,2
9 Letras	145.241	2,8
10 Ciências Biológicas	133.204	2,6
Outros cursos	1.795.703	35,1

Educação a Distância		
Curso	Matrículas	%
Total	838.125	100
1 Pedagogia	286.771	34,2
2 Administração	228.503	27,3
3 Serviço Social e orientação	68.055	8,1
4 Letras	49.749	5,9
5 Ciências Contábeis	29.944	3,6
6 Matemática	23.774	2,8
7 Ciências Biológicas	19.626	2,3
8 História	16.864	2,0
9 Comunicação Social	15.802	1,9
10 Ciências ambientais e proteção ambiental	13.091	1,6
Outros cursos	85.946	10,3

Fonte: Censo da Educação Superior/MEC/Inep/Deed

Quadro 2.3 – Os dez maiores cursos de graduação do Brasil em termos de matrículas em 2009
 Fonte: Adaptado do MEC, 2010

Outro dado positivo que pode ser extraído do Censo 2009 é que nos últimos 7 anos, o número de concluintes praticamente dobrou, passando de 467 mil, em 2002, para 959 mil em 2009.

2.1.1 A educação superior na área de computação e informática

No âmbito da área de Computação, o Censo do MEC/INEP apresenta os números constantes no Quadro 2.4 e na Figura 2.2:

Distribuição dos Cursos				
Modalidade de Cursos	2007	2008	Evolução (%)	Panorama 2008 (%)
Ciência da Computação	321	328	2,18	16,06
Engenharia de Computação	86	93	8,14	4,55
Sistemas de Informação	529	538	1,70	26,35
Licenciatura em Computação	65	78	20,00	3,82
Engenharia de Software	0	1	-	0,05
Curso Sequencial de Formação Específica	38	42	10,53	2,06
Cursos de Tecnologia (Todos)	842	907	7,72	44,42
Outros Cursos	54	55	1,85	2,69
Total	1817	2042	12,38	100,00

Quadro 2.4 – Distribuição dos cursos da área de computação 2007-2008
 Fonte: Adaptado do MEC, 2010

Outra forma de visualizar essas mesmas informações é através da Figura 2.2:

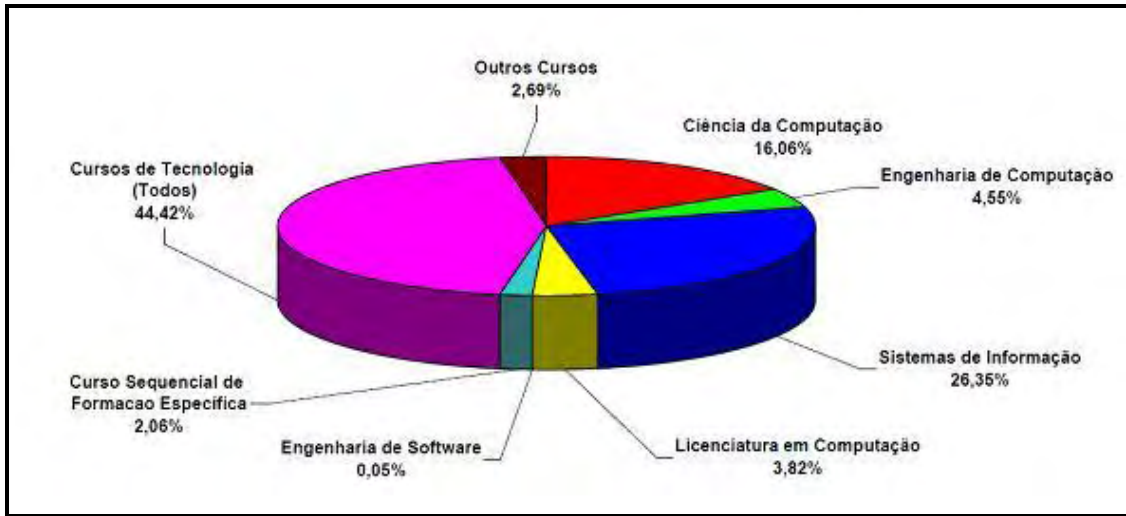


Figura 2.2– Distribuição dos cursos da área de computação 2007-2008
Fonte: Adaptado do MEC, 2010

Da análise do quadro e do gráfico, depreende-se que o número de cursos tem trajetória crescente, sendo predominantemente representados pelos cursos de tecnologia (44,42%), sistemas de informação (26,35%) e ciência da computação (16,06%).

A figura 2.3, a seguir, apresenta a evolução das matrículas na graduação na área de computação no período de 2001 a 2008, exibindo uma curva ascendente no número de estudantes que se matriculam nesses cursos, atingindo um patamar superior a 300 mil alunos matriculados em 2008.

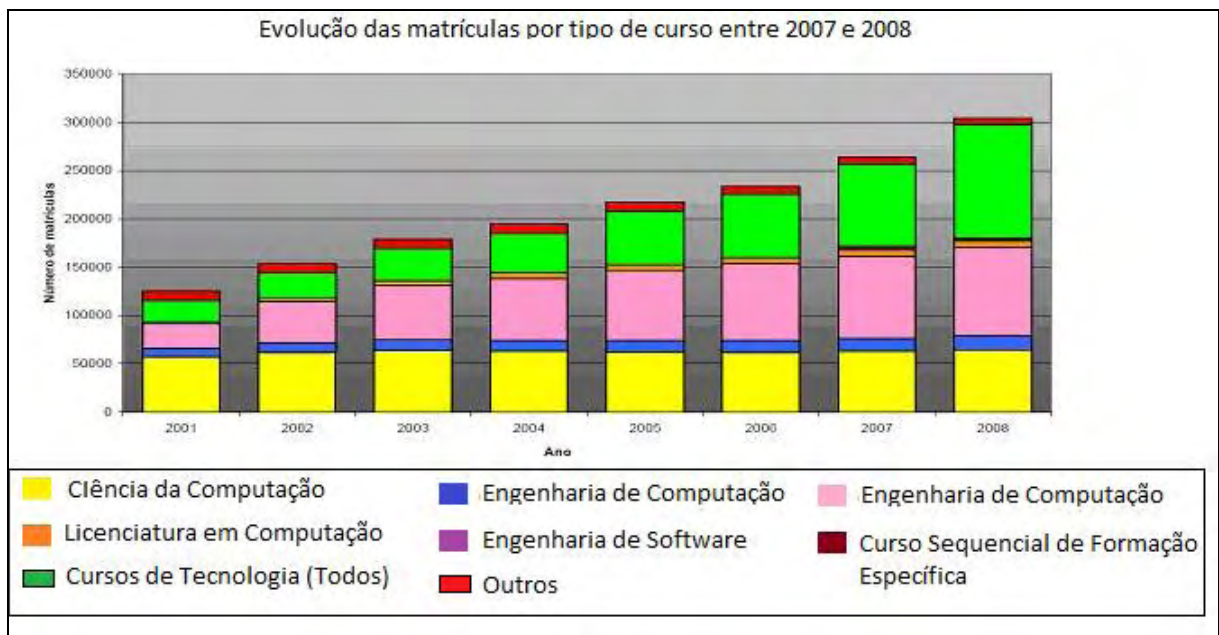


Figura 2.3 – Evolução das matrículas dos cursos da área de computação 2007-2008
Fonte: Adaptado do MEC, 2010

Em seguida, a Figura 2.4 mostra que são colocados no mercado todos os anos quase 40 mil novos profissionais na área de computação. Uma análise comparativa entre os dois gráficos leva à conclusão de que existe uma tendência decrescente no total de alunos matriculados nos cursos da área, sendo detectado um déficit anual da ordem de 5000 alunos a menos na diferença entre os concluintes e ingressantes.

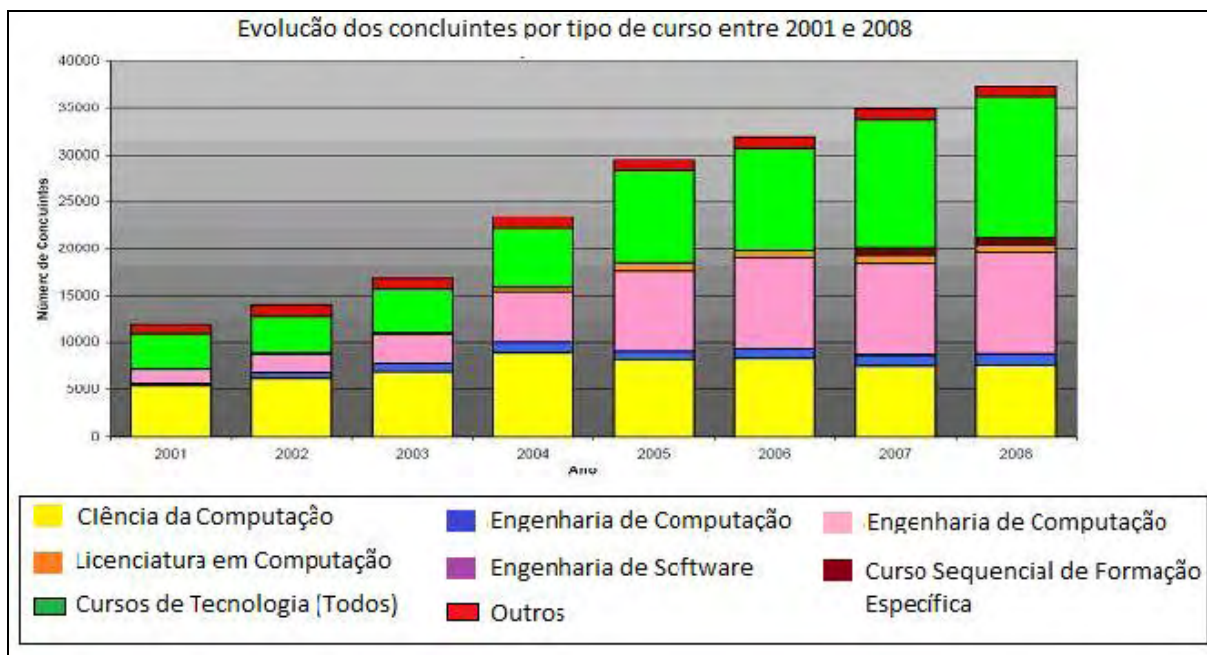


Figura 2.4 – Evolução dos concluintes dos cursos da área de computação 2007-2008
Fonte: Adaptado do MEC, 2010.

Outro aspecto importante a ser ressaltado é a significativa taxa de evasão⁷ dos alunos da área de computação, evidenciando a necessidade urgente da introdução de mudanças nas estratégias de ensino-aprendizagem empregadas nesses cursos.

A partir desse cenário, surge a preocupação com o ensino na área específica e estratégica de engenharia de *software*. Na realidade, a metodologia proposta poderia ser utilizada em outras áreas do conhecimento que tenham disciplinas cujos conteúdos tem um pré-requisito entre elas e que necessitam de uma maior cooperação entre os professores que ministram os conteúdos. A Engenharia de Software foi escolhida por ter disciplinas cujos conteúdos podem ser ministrados com um maior ou menor detalhamento em uma grade curricular nas disciplinas de Especificação de Software, Métodos, Análise e Projeto de Sistemas I e II.

⁷ Censo da Educação Superior MEC/2010

2.2 O ENSINO DE ENGENHARIA DE SOFTWARE

Na busca de um maior rigor para descrever o ensino de ES, buscou-se subsídio na Teoria dos Conjuntos. Assim, pode-se pensar em Instituições de Ensino Superior (IES) que contam com um corpo docente, onde professores $P = \{ p_1, p_2, \dots, p_n \}$ são responsáveis por ministrar disciplinas $D = \{ d_1, d_2, \dots, d_n \}$ que fazem parte de uma Grade Curricular $G = \{ \langle e_1, c_1 \rangle, \langle e_2, c_2 \rangle, \dots, \langle e_n, c_n \rangle \}$ tal que cada disciplina de D tem uma relação $(\langle e_i, c_i \rangle)$ entre uma ementa e_i e o respectivo conteúdo c_i que deve ser ministrado. A forma de se aferir se o conteúdo c_i foi realmente ministrado por p_i é realizado por uma avaliação $A = \{ \langle t_1, t_2, \dots, t_n \rangle, \langle s \rangle \}$ que é composta por diversas ferramentas avaliativas, destacando-se dentre elas um projeto obrigatório, onde um *software* S é elaborado.

O ensino de engenharia de *software* exige um grau muito grande de abstração e conceitos teóricos que, muitas vezes, torna o ensino dos seus conteúdos um verdadeiro desafio para quem leciona e para quem aprende.

Nos últimos anos, pesquisadores e órgãos governamentais preocupados com o ensino de engenharia de *software* têm apresentado várias iniciativas que visam melhorar o processo de ensino-aprendizagem nesta área que são resumidas por Paiva (2011a).

Pioneiros da área de engenharia de *software*, a exemplo de Parnas (1998), advogam o estabelecimento de fronteiras bem claras entre o corpo de conhecimento da ciência da computação e de engenharia de *software*, inclusive sendo o primeiro a propor a criação de cursos específicos na área de ES, o que começou a ocorrer, no Brasil, em meados da década passada.

Entidades como o MEC (2005), a ACM (2001) e a SBC - Sociedade Brasileira de Computação (2005), tentaram estabelecer padrões e parâmetros norteadores para o ensino de engenharia de *software*.

A comissão de especialistas de ensino de computação e informática do MEC publicaram as diretrizes curriculares de cursos da área de computação e informática, dentro das quais se definem regras para o ensino de Engenharia de *Software* nos seguintes termos “Engenharia de Software compreende um conjunto de disciplinas matemáticas, técnicas (em computação), sociais e gerenciais que sistematizam a produção, a manutenção, a evolução e a recuperação de produtos intensivos em software”. Isso ocorre dentro de prazos e custos estimados, com progresso controlado e utilizando princípios, métodos, tecnologias e processos em contínuo aprimoramento. Os produtos desenvolvidos e mantidos segundo os preceitos de Engenharia de Software asseguram, por construção, qualidade satisfatória,

apoiando adequadamente os seus usuários na realização de suas tarefas, operam satisfatória e economicamente em ambientes reais e podem evoluir continuamente, adaptando-se a um mundo em constante evolução. Segundo as diretrizes (MEC, 2005, p.10) “O ensino de Engenharia de Software em cursos de graduação pode dar origem a várias disciplinas com diferentes ênfases. A origem dessas disciplinas pode ter como motivação diferentes classificações didáticas: aspectos gerenciais, aspectos técnicos, aspectos teóricos e aspectos experimentais. A ênfase pode se dar em diferentes etapas do processo de desenvolvimento e manutenção de software: engenharia de requisitos, análise, arquitetura e projeto, programação, testes, manutenção, garantia de qualidade e gestão do processo de software. É importante notar que esses aspectos devem estar integrados em outras disciplinas, como por exemplo: bancos de dados, interface homem-máquina, sistemas de informação, redes e laboratórios diversos”. O documento cita ainda que deve-se assegurar que o estudante adquira experiência na aplicação destes conceitos através da prática em laboratórios e estágios. É fortemente recomendado que o estudante seja exposto a uma variedade de sistemas operacionais, sistemas de gerenciamento de bancos de dados, linguagens e paradigmas de programação, plataformas de operação, e de ferramentas de apoio ao desenvolvimento de software e documentação.

Através dessas diretrizes curriculares, o MEC orienta como cada disciplina/matéria deve ser lecionada. No Quadro 2.5, são exibidas as orientações dadas para o ensino de Engenharia de *Software* em cada um dos cursos da área de computação:

CURSO	Disciplina Engenharia de <i>Software</i>
Bacharelado em Ciência da Computação	As disciplinas devem cobrir os fundamentos / estruturas de todas as tecnologias e pelo menos uma delas (ênfase) com profundidade com vistas à realização de projetos. A Prática do Ensino de Computação deve ser dispensada.
Engenharia de Computação	As disciplinas devem cobrir os fundamentos/estruturas de todas as tecnologias e pelo menos uma delas (ênfase) com profundidade com vistas à realização de projetos. A Prática do Ensino de Computação deve ser dispensada.
Bacharelado em Sistemas de Informação	As disciplinas devem cobrir os fundamentos e/ou estruturas de todas as tecnologias, sem a necessidade, contudo, de capacitar os alunos ao projeto das mesmas. Devem-se usar intensivamente, em laboratório, as tecnologias correntes: banco de dados, engenharia de <i>software</i> , redes de computadores, entre outras. A Prática do Ensino de Computação deve ser dispensada.
Licenciatura em Computação	As disciplinas devem cobrir os fundamentos/estruturas de todas as tecnologias, sem a necessidade, contudo, de capacitar os alunos ao projeto das mesmas. A matéria "Prática do ensino de Computação" deve ser coberta com abrangência e profundidade, totalizando horas, conforme determina a LDB (Art. 65), e a aplicação da pedagogia voltada para o "como ensinar em geral".

Quadro 2.5 – Orientação quanto ao Ensino de Engenharia de *Software*

Fonte: Adaptado de MEC, 2005

Através da análise das recomendações das Diretrizes Curriculares, constantes no Quadro 2.5, pode-se observar um tratamento diferenciado para cada perfil de egressos em cada um dos cursos superiores da área de Computação, como também uma atenção especial e explícita que é dada à adoção de projetos ao longo das disciplinas da área de Engenharia de *Software*.

Outro exemplo de uma contribuição por parte dessas entidades é a definição do Currículo de Referência da área de computação da SBC (2005), explicitando que a ES faz parte do grupo de matérias específicas da computação, pertencendo ao núcleo de Tecnologia da computação, incorporando os seguintes tópicos ou disciplinas: processo de desenvolvimento de *software*; ciclo de vida do desenvolvimento de *software*; qualidade de *software*; técnicas de planejamento e gerenciamento de *software*; gerenciamento de configuração de *software*; engenharia de requisitos; métodos de análise e projeto de *software*; garantia de qualidade de *software*; verificação, validação e teste; manutenção; documentação; padrões de desenvolvimento; reuso; engenharia reversa; reengenharia; ambientes de desenvolvimento de *software*.

A ACM/IEEE (ACM, 2001) publicou no ano de 2001 o seu Currículo de Referência para a área de computação no qual explicitou os conteúdos essenciais para a subárea de Engenharia de *Software*, estabelecendo disciplinas nucleares e a carga horária mínima semestral para cada uma:

SE. Engenharia de *Software* (31 horas):

SE1: Projeto de *Software* (8);

SE2: Usando API (5);

SE3: Ferramentas de *Software* e ambientes (3);

SE4: Processo de *Software* (3);

SE5: Requerimentos e especificação de *Software* (5);

SE6: Validação de *Software* (3);

SE7: Evolução de *software* (3);

SE8: Gerenciamento de Projeto de *Software* (3);

SE9: Computação baseada em componentes;

SE10: Métodos formais;

SE11: Confiabilidade de *Software*;

SE12: Desenvolvimento de Sistemas Especializados.

Observe-se que as disciplinas SE9 a SE12 são opcionais, podendo ser ofertadas em função de uma necessidade específica do perfil do curso ou interesse particular. Através da observação e vivência no mundo acadêmico, pode-se afirmar que estas informações associadas às disciplinas da área de ES e especialmente no caso daquelas em que há uma relação de pré-requisitos, nem sempre são repassadas para que os docentes tenham conhecimento integral da área, gerando, muitas vezes problemas de comunicação ao longo do trabalho docente.

Outro trabalho nesta mesma linha de tentar estabelecer parâmetros sobre o ensino e o conhecimento na área de engenharia de *software* é o “Guia do corpo de conhecimento em engenharia de *software*” publicado pelo IEEE em 2004 na tentativa de estabelecer as fronteiras para o corpo de conhecimento nesta área, tendo ficado conhecido como projeto Swebok (IEEE, 2004). O projeto Swebok enquadrava o conhecimento de ES em 10 áreas distintas:

- SE1: Requerimentos de *Software*;
- SE2: Design *Software*;
- SE3: Construção de *Software*;
- SE4: Teste de *Software*;
- SE5: Manutenção de *Software*;
- SE6: Gerenciamento de Configuração de *Software*;
- SE7: Gerenciamento de Engenharia *Software*;
- SE8: Processo de Engenharia de *Software*;
- SE9: Métodos e Ferramentas em Engenharia de *Software*;
- SE10: Qualidade de *Software*.

A Figura 2.5 ilustra alguns ramos dessa árvore do conhecimento em ES:

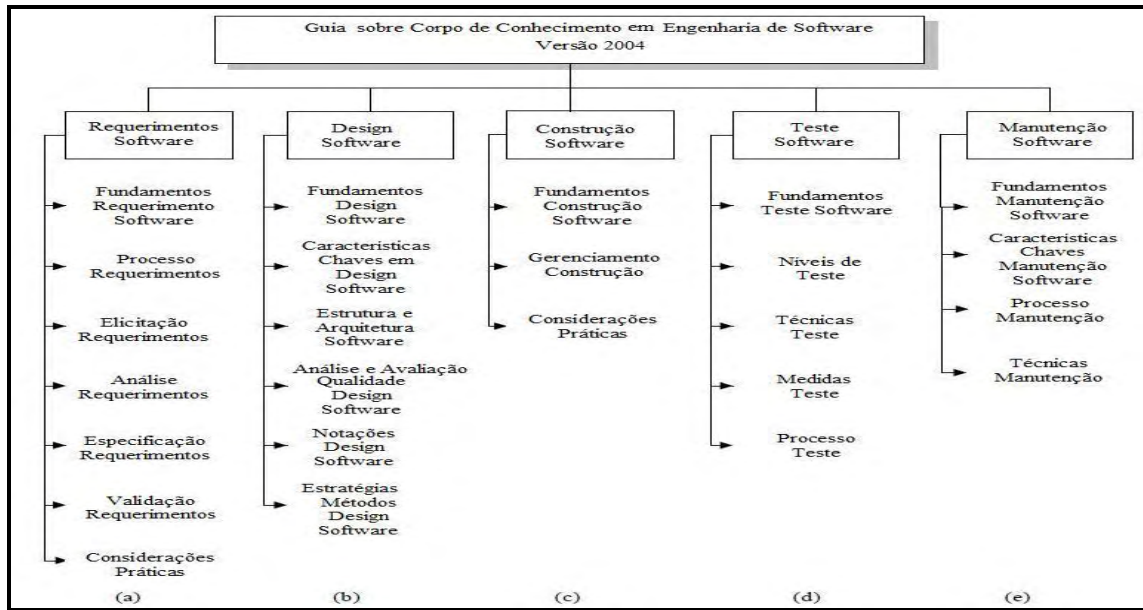


Figura 2.5 – Guia do corpo de conhecimento em Engenharia de *Software*
Fonte: Adaptado de IEEE, 2004

2.2.1 Trabalhos relacionados

De uma maneira geral, registra-se em anos recentes, um grande número de pesquisas científicas focadas na análise do processo ensino-aprendizagem das disciplinas ligadas à grande área de Engenharia de *Software* (PAIVA, 2011a).

Vários autores escreveram sobre metodologias para facilitar o aprendizado de conceitos que podem ser aplicadas ao ensino da Engenharia de *Software*. Relacionamos alguns trabalhos interessantes e colocamos no Apêndice A. Nesta relação, a abordagem que mais se aproxima de nossa proposta é a de Dekhane & Tsoi (Dekhane, S. & Tsoi, M. Y.,2010) pela forma como trata a interdisciplinaridade e a importância dada aos conceitos ministrados e materializados nos projetos.

2.2.2 Pesquisa de opinião sobre o ensino de engenharia de *software*

No sentido de coletar mais subsídios para analisar o ensino de ES, foi realizada uma pesquisa de opinião com entrevistas sendo executadas através de visitas *in loco* nas instituições de ensino superior na cidade de João Pessoa, como também via formulários disponibilizados através da *internet* e entrevistas feitas no Congresso da Sociedade Brasileira de Computação 2011 em Natal-RN, conforme mostram os apêndices B e C (PAIVA, 2011c).

Outra fonte de coleta dessas informações foi a visita a algumas empresas da área de *software* na cidade de João Pessoa com a aplicação dos referidos questionários aos profissionais que se enquadravam no perfil desejado.

As respostas aos questionários somaram 142 respondentes oriundos de mais de 30 instituições do Brasil, totalizando 1,34% do total das IES brasileiras. Foram 30 questionários de alunos ou ex-alunos de instituições da Paraíba e 112 de outros estados.

Do universo de alunos/profissionais entrevistados, podem-se inferir as seguintes conclusões:

- a) Os entrevistados são oriundos de 30 instituições de ensino superior, representando as cinco regiões do país, predominando a UFPI (12%), IESA (9%), UFCG (6%), UFF (6%), UFPA (6%), IFRN (6%), IESB (4%) e UNIPÊ (4%);
- b) Os cursos predominantes entre os participantes da pesquisa são Ciência da Computação (50%), Sistemas de Informação (24%) e cursos tecnológicos (15%);
- c) Cerca de 85% dos entrevistados não possuíam nenhuma experiência em desenvolvimento de *software* antes de cursar o seu curso superior na área de computação;
- d) Os dois principais livros textos utilizados no ensino das disciplinas da área de Engenharia de *Software* são o de Roger Pressman (41%) e o de Ian Sommerville (51%), sendo seguidos por apostilas próprias elaboradas pelos docentes (30%);
- e) As disciplinas da área de Engenharia de *Software* cursadas pelos entrevistados têm carga horária de até 60 horas (49%) e de até 75 horas (23%), sendo afirmado que na grade dos seus cursos têm, no máximo, 2 disciplinas (35%) ou 1 disciplina (26%). A maioria dos entrevistados considera a carga horária das disciplinas insuficiente (52%) para um bom aprendizado;
- f) As principais ferramentas didáticas utilizadas pelos docentes para ministrarem os seus cursos são aulas expositivas (82%), aulas práticas de laboratório (61%), elaboração de projetos (56%), apresentação de seminários (45%), estudo de casos (40%) e ensino baseado em problemas (27%). Com relação aos projetos desenvolvidos durante as disciplinas, os entrevistados afirmam que em apenas 53% das disciplinas cursadas houve o desenvolvimento de projetos; os entrevistados avaliam ainda que, em sua grande maioria (55%), não houve a implementação de projetos ou os projetos não tiveram nenhuma aplicação prática, sendo utilizados apenas para atribuir notas aos discentes nas disciplinas;

- g) Os mecanismos de avaliação utilizados nas disciplinas cursadas foram as provas subjetivas (72%), implementação de projetos (59%), seminários (51%) e provas objetivas (44%);
- h) Com relação ao conhecimento prévio do programa e do planejamento pedagógico das disciplinas no início das mesmas, os entrevistados afirmam que tomaram conhecimento (70%) e que os mesmos teriam sido entregues pelos docentes em sala (45%);
- i) Após cursar as disciplinas da área de ES, os entrevistados julgam que suas habilidades na área de desenvolvimento de *software* são apenas regulares (30%), pouca (16%) ou nenhuma (4%), avaliando que os seus conhecimentos na área estão na média das suas turmas (61%), embora a maioria (59%) considere-se apta a entrar no mercado de trabalho; perguntados adicionalmente sobre o domínio dos conhecimentos de ES, os entrevistados afirmam terem domínio apenas regular (46%), pouco (11%) ou nenhum (3%);
- j) Ao avaliarem a qualidade do atendimento docente extra-sala de aula, os entrevistados julgaram ser ótimo (25%) ou bom (35%), afirmando que foram utilizados como mecanismos de comunicação adicional as listas de discussão (45%), ferramentas do Google Docs (37%) e MSN (11%);
- k) Perguntados sobre a ocorrência de uma efetiva cooperação entre os docentes da área na elaboração de conteúdos, na continuidade dos assuntos e nos projetos das disciplinas, os entrevistados avaliam que perceberam um trabalho em equipe (44%), que cada docente fazia o seu trabalho independente (27%), que todas as disciplinas foram ensinadas por apenas um docente (21%) e que não houve nenhuma cooperação, havendo até críticas ao trabalho dos outros docentes (7%);

Outro questionário foi elaborado visando a aplicação restrita aos professores e coordenadores de cursos com disciplinas na área de engenharia de *software*. Observou-se uma maior dificuldade em obter participantes com esse perfil, tendo sido observada até certa resistência à participação por parte de alguns. Ao todo, foram 13 respondentes de 9 instituições superiores das regiões Nordeste, Centro-Oeste e Sul, dos quais 15% são da IESB, 8% da IESA, 8% do UNIPÊ e 69% de outras IES, sendo destes, 3 respondentes não quiseram se identificar. Com base nas respostas, pode-se afirmar:

- a) 54% dos entrevistados são formados em ciência da computação, 23% em sistemas de informação, 15% em licenciatura em computação e 15% em cursos tecnológicos;
- b) 46% dos entrevistados têm mais de 5 anos de atividade docente, 31% têm mais de 3 anos e menos de 5 anos de experiência acadêmica e 23% têm menos de 3 anos de experiência profissional como docente;
- c) 38% dos entrevistados afirmam terem como a sua mais alta titulação o título de mestre, 23% são doutores, 23% são especialistas e 15% são graduados;
- d) Sobre o planejamento pedagógico das disciplinas ministradas, os docentes afirmam que foi feito individualmente por eles (47%), foi feito de forma colegiada pelos docentes da área de ES (37%), foi feito de forma colegiada pelos docentes do curso (8%) e 8% afirmam não ter havido nenhum planejamento;
- e) Perguntados sobre o planejamento pedagógico envolvendo duas disciplinas (d1 e d2) por dois docentes (p1 e p2), os docentes afirmaram que o planejamento ocorre de forma isolada por cada um dos docentes (54%), é feito em equipe ajustando parcialmente conteúdos e estratégias (23%) e que é feito em equipe ajustando totalmente conteúdos e estratégias (23%);
- f) Indagados sobre a forma de condução das disciplinas da área de ES, os docentes afirmaram que o fazem de forma livre e independente (92%);
- g) Com relação ao cumprimento do conteúdo programático com a qualidade desejada, os docentes afirmam que o fazem bem (69%);
- h) Com relação à carga horária das disciplinas, 62% afirmam que a carga horária majoritária das disciplinas é de 60 horas (62%), existindo nas grades dos seus cursos 3 disciplinas (38%), 2 disciplinas (31%), e mais de 3 disciplinas (23%), sendo avaliado pelos docentes que a carga horária é suficiente para fazer um bom trabalho (85%);
- i) Perguntados sobre as ferramentas didáticas utilizadas para ministrar as suas aulas, os docentes responderam que usam aulas expositivas (92%), apresentação de seminários pelos alunos (69%), aulas práticas de laboratório (62%), adoção de projetos (62%), ensino baseado em problemas (38%) e elaboração de artigos (38%);
- j) Indagados sobre os mecanismos utilizados para avaliação dos alunos em suas disciplinas, os docentes responderam que usam a apresentação de seminários

(77%), provas subjetivas (69%), implementação de projetos (69%), confecção de artigos (38%) e provas objetivas (31%);

- k) Quanto aos projetos desenvolvidos ao longo das suas disciplinas, os professores afirmaram que eles não tinham utilidade outra senão o uso como mecanismo de avaliação (38%), não houve a realização de projetos em suas disciplinas (23%), os projetos serviram para viabilizar um produto que foi para o mercado (15%) e que os projetos foram utilizados como ferramenta na própria instituição de ensino (15%);
- l) Questionados sobre a qualidade do atendimento docente para dúvidas, os professores afirmaram ser regular (77%), bom (15%) e ótimo (15%), sendo utilizadas algumas ferramentas de comunicação extra-classe tais como: listas de discussão (54%), ferramentas do Google (46%), sistema EAD (Moodle ou outra ferramenta) (23%), Twitter (15%) e MSN (15%);
- m) Perguntados sobre a crença na viabilidade dos mecanismos de comunicação via Web, os docentes responderam que acreditam, mas têm usado pouco, pretendo incrementar o seu uso (54%); acreditam e têm usado com bons resultados (31%); outros 8% afirmaram não acreditar e não pretenderem usar tais ferramentas;
- n) Com relação aos livros textos utilizados como referências em suas aulas, afirmaram adotar Ian Sommerville (85%), Roger Pressman (77%), apostila elaborada pelos docentes (46%);
- o) Indagados sobre considerarem os seus alunos aptos para o mercado de trabalho, responderam que ainda não, mas com um pouco mais de aperfeiçoamento estarão em condições (69%);
- p) Com relação à aceitação de algum trabalho de supervisão pedagógica, os respondentes afirmaram receberem muito bem esse tipo de trabalho (46%) e terem boa aceitação ao acompanhamento (46%).

2.2.3 Problemas com o Ensino de Engenharia de *Software*

Conforme pode ser obtido da análise dos trabalhos relacionados no Apêndice A (PAIVA, 2011a) e das respostas à pesquisa de opinião, os principais problemas no processo de ensino-aprendizagem em ES são:

- 1) A existência de uma certa distância entre o que se ensina e a realidade existente no mercado de trabalho;
- 2) Os discentes têm pouco interesse pelas aulas maciçamente teóricas;
- 3) Em geral, os cursos de ES têm significativas taxas de evasão e reprovação;
- 4) Índices de aprendizagem efetiva questionáveis, principalmente quando os alunos são confrontados com situações reais do mercado de trabalho;
- 5) É dada pouca ênfase ao trabalho em grupo;
- 6) Pouco estímulo à interdisciplinaridade;
- 7) Em geral, o que se ensina e aprende em um semestre ou disciplina, não é apropriado pela comunidade como um patrimônio intelectual para uso futuro;
- 8) O planejamento dos conteúdos e a prática pedagógica são ditados pelo interesse de cada docente sem passar, muitas vezes, pelo crivo institucional;
- 9) Há pouca ou nenhuma interação entre professores e alunos fora da sala de aula;
- 10) O conhecimento produzido em um curso ou disciplina não é compartilhado com a comunidade interna ou externa à IES;
- 11) Cada Instituição/Docente segue uma linha de acordo com a sua visão/concepção.

Com base nessas constatações, pode-se afirmar que, os resultados parciais obtidos encorajaram no sentido da continuação da investigação sobre o impacto no ensino de ES utilizando a abordagem da metodologia ESCollab, proposta neste trabalho.

Baseado no estudo de caso realizado nas instituições IESA e IESB podemos justificar os pontos acima: 1) as grades curriculares dos cursos não acompanham o mercado; 2) há uma carência na formação dos mestres com relação às práticas atuais de mercado e a falta de estrutura laboratorial para oferecer práticas atuais casadas com a teoria; 3) geralmente as disciplinas de ES são no final do curso e que já acumulam desistências dos períodos anteriores; 4) foi constatado um número de sete ou mais disciplinas o que dificulta a execução de projetos mais abrangentes; 5) há uma resistência de cooperação entre disciplinas diferentes

pela própria logística relacionada a montagem de horários em comum de turmas diferentes e de horários diferentes o que dificulta o trabalho em grupo; 6) e 7) não era prática comum a realização de trabalho cooperativo. Esta proposta de trabalho motivou uma cooperação pioneira entre professores distintos; 8) Apesar de a IESB ter em sua organização uma proposta de planejamento anual na prática os professores trabalhavam de forma isolada; 9) As redes sociais e a WEB 2.0 serviram como forma de motivar o trabalho nos itens 10 e 11 de forma cooperativa.

CAPÍTULO III

FUNDAMENTOS DE ENSINO E APRENDIZAGEM

3.1 INTRODUÇÃO

Este capítulo visa caracterizar os principais elementos pertencentes à teoria educacional e ao processo ensino-aprendizagem, estabelecendo a sua inter-relação com as tecnologias da informação e comunicação (TIC's). Como podemos ver em Libâneo (LIBÂNEO, 2002, p. 57) há uma complexidade da vida social contemporânea e a conseqüente diversificação das atividades educativas. Estas últimas resultam, ao mesmo tempo, em ampliação das ações pedagógicas com uma necessidade de ampliação das visões de novos horizontes integrados com outros segmentos sociais não apenas fechado ao ambiente acadêmico mas também nos meios profissionais, político, sindical, empresarial, nos meios de comunicação social, nos movimentos da sociedade civil, assiste-se a uma redescoberta da pedagogia. Como Libâneo enfatiza “Estamos diante de uma sociedade pedagógica” em movimento.

Essa importância do processo ensino-aprendizagem tem gerado, ao longo do tempo, vários estudos envolvendo pedagogos, psicólogos e outros profissionais na tentativa de entenderem os segredos da cognição⁸ humana. Esta perspectiva nos remete ao pensamento de Grinspun (GRINSPUN, 2001, p. 38) onde a “...educação é co-extensiva ao ato de viver. Todo homem em qualquer lugar, em qualquer circunstância, está envolto pelo processo educativo. Não podemos pensar em fazer educação desvinculada do processo de produção e das relações sociais, ou mais precisamente, sem uma estreita relação com o projeto de sociedade. Assim, é que a educação precisa estar voltada para a realidade, mais exatamente para transformá-la”

⁸ Ato ou processo de conhecer, envolvendo atenção, percepção, memória, razão, raciocínio, imaginação, pensamento e linguagem. O termo teve origem nos escritos de Aristóteles e Platão.

As principais correntes de estudo sobre os enfoques teóricos de ensino-aprendizagem, suas abordagens e principais teóricos pode ser visto através da Figura 3.1:

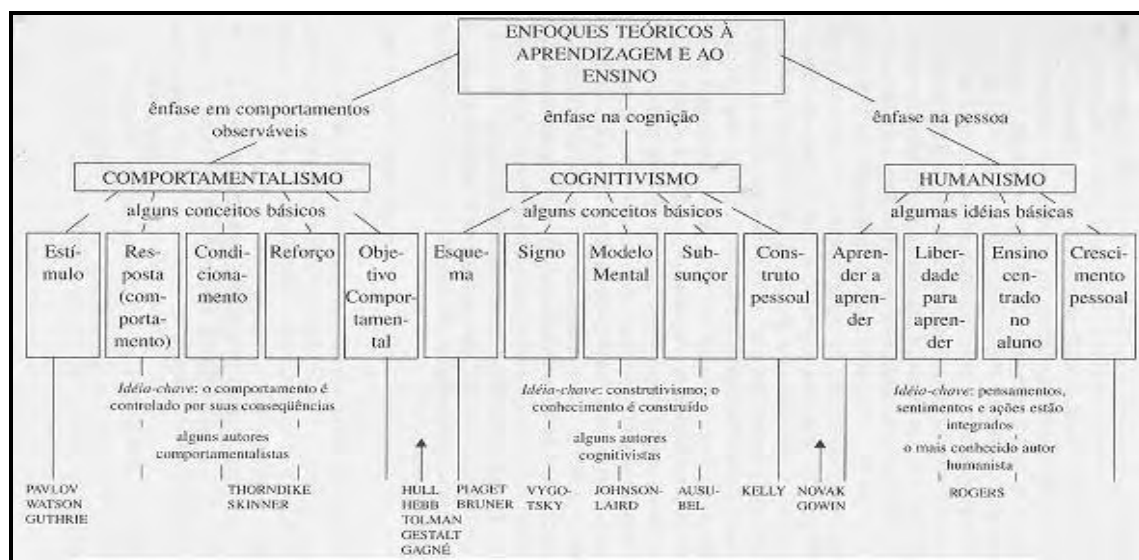


Figura 3.1 – Espiral do conhecimento
 Fonte: Adaptado de Moreira (1999, p. 18)

A Figura 3.1 apresenta as principais correntes de estudiosos da aprendizagem humana, merecendo destaque às idéias do construtivismo e do humanismo. Para subsidiar a fundamentação da discussão pedagógica sobre o ensino de ES, o Quadro 3.1, a seguir, apresenta as características básicas das principais teorias da aprendizagem que embasam o presente trabalho:

TEORIAS DA APRENDIZAGEM	CARACTERÍSTICAS
1) Epistemologia Genética de Piaget	a) O ponto central é a estrutura cognitiva do indivíduo; b) Níveis diferentes de desenvolvimento cognitivo; c) Desenvolvimento facilitado pela oferta de atividades e situações desafiadoras; d) A interação social e a troca entre os indivíduos estimulam o processo de aquisição do conhecimento.
2) Construtivista de Bruner	a) O aprendiz é participante ativo do processo de aquisição do conhecimento; b) Instrução relacionada a contextos e experiências pessoais; c) Determinação de seqüências mais efetivas de apresentação do material; d) Teoria atual com ênfase na criação de comunidades de aprendizagem mais próximas das práticas colaborativas do mundo contemporâneo.
3) Teoria Sócio-Cultural de Lev Vygotsky	a) Desenvolvimento cognitivo é limitado a um determinado potencial para cada intervalo de idade (Zona Proximal de Desenvolvimento);

	b) Desenvolvimento cognitivo completo requer interação social.
4) Aprendizagem baseada em problemas / Instrução ancorada	a) A aprendizagem é iniciada com a proposta de um problema a ser resolvido (âncora ou foco); b) A aprendizagem é centrada no indivíduo e contextualizada.
TEORIAS DA APRENDIZAGEM	CARACTERÍSTICAS
5) Cognição Distribuída	a) Interação entre indivíduo, ambiente e artefatos culturais; b) Ensino recíproco; c) Importante papel para a tecnologia.
6) Teoria da Flexibilidade Cognitiva	a) Reestruturação do conhecimento como resposta a demandas situacionais; b) Revisitação ao material instrucional; c) Atividades devem conter múltiplas representações do ensinado.
7) Cognição Situada	a) Aprendizagem ocorre em função da atividade, contexto, cultura e ambiente social na qual está inserida; b) Interação social e colaboração são componentes críticos para a aprendizagem (comunidade de prática).
8) Aprendizagem Auto-regulada / Metacognição	a) Controle e monitoramento da própria cognição pelo sujeito;
9) Aprendizagem por observação	a) Auto-observação, auto-julgamento, auto-reação. b) Não ação, não participação; c) Ouvir ou assistir libera a realização de outros processos mentais; d) Não ação, não participação; e) Ouvir ou assistir libera a realização de outros processos mentais; f) Utilização da memória; g) Obtenção de informação através de diálogos.

Quadro 3.1 – Teorias da Aprendizagem e suas características

Fonte: Adaptado de Santos; Santoro e Borges, 1999

Para cada uma das teorias apresentadas no Quadro 3.1, tem-se aspectos relevantes que podem ser apropriados na construção de uma metodologia voltada para o ensino das disciplinas da área de Engenharia de *Software*. Pode-se observar que os aspectos mais relevantes, no ponto de vista do presente trabalho, em cada teoria estão destacados em negrito.

Moreira (1999) apresenta um esquema representando os oito tipos de aprendizagem, segundo o qual qualquer habilidade intelectual pode ser analisada em termos da combinação de habilidades mais simples para produzir a sua aprendizagem.

A Figura 3.2 ilustra essa hierarquia da aprendizagem:

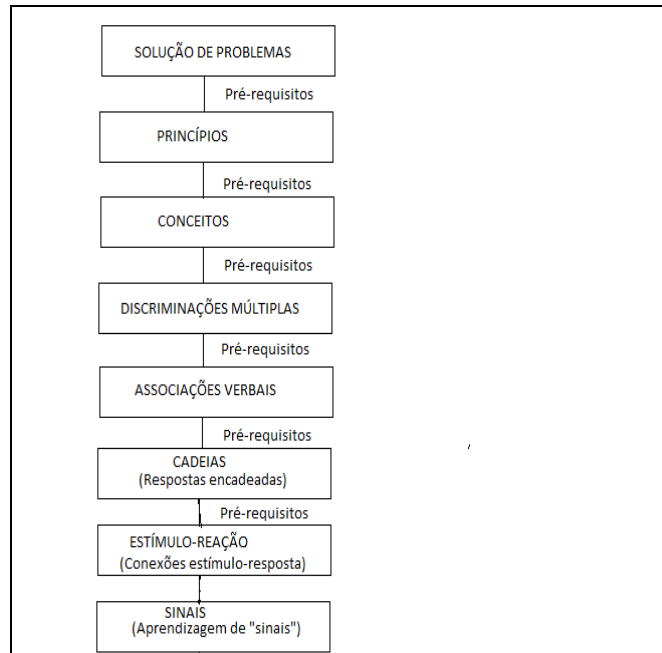


Figura 3.2– Hierarquia dos 8 tipos de aprendizagens
Fonte: Adaptado de Moreira (1999, p.73)

Os tipos de aprendizagem apresentados por Moreira (1999) podem ser explicados da seguinte forma:

- Tipo 1: Sinais - corresponde à aprendizagem de “sinais”, sendo ligadas às respostas gerais, difusas, emocionais, caracterizando-se por serem respostas que não se acham sob controle voluntário;
- Tipo 2: Estímulo-Reação – corresponde à aprendizagem associada a uma resposta precisa a um dado estímulo;
- Tipo 3: Cadeias – são as respostas encadeadas de duas ou mais conexões de estímulo-resposta;
- Tipo 4: Associações verbais – é um subtipo da aprendizagem em cadeia, sendo associado às cadeias verbais;
- Tipo 5: Discriminações Múltiplas – associado à capacidade de responder de forma diferenciada a diferentes estímulos dentro de um conjunto;
- Tipo 6: Conceitos – corresponde a uma resposta a uma classe de estímulos como um todo, caracterizando-se por uma generalização do caso 5;
- Tipo 7: Princípios – está associado a cadeias de conceitos e regras;
- Tipo 8: Solução de Problemas – corresponde à elaboração de novos princípios que combinem com outros já absorvidos anteriormente.

Merece destaque neste esquema da Figura 3.2 a posição no topo da sua hierarquia para a resolução de problemas, que é, naturalmente, o mais complexo e de longe o mais utilizado no ensino de Engenharia de *Software*, demonstrando a complexidade exigida no processo ensino-aprendizagem nesta área.

Normalmente, na condução do processo de ensino-aprendizagem, segundo Bordenave (2007:152), o docente utiliza métodos de ensino e algumas ferramentas didáticas, visando facilitar a aprendizagem:

TÉCNICA	DESCRIÇÃO
Aula Expositiva	Aula ministrada nos moldes tradicionais, normalmente, conduzida pelo docente responsável pela disciplina.
Aula Prática	Aula realizada em ambiente de laboratório utilizando alguma ferramenta computacional
Desenvolvimento de Projetos	Um ou mais projetos são propostos à turma para serem desenvolvidos ao longo da disciplina. O projeto pode ser trabalhado individualmente ou em grupo dependendo da sua complexidade e do tempo disponível, havendo casos em que um projeto transcende o tempo de vida de uma disciplina.
Estudo de Caso	Uma situação real é trazida para o ambiente acadêmico no sentido de discutir e vivenciar a experiência original ocorrida no caso em estudo.
Mini-teste	Testes mínimos feitos ao longo da disciplina, possibilitando a realização de um modelo de avaliação contínua, sem marcação prévia, com o intuito de forçar a turma a estudar continuamente os conteúdos ministrados.
PBL – Aprendizagem baseada em Problemas	Um problema é proposto à turma e o docente faz a mediação em busca da solução mais apropriada. A técnica PBL tem semelhanças com o estudo de caso.
Trabalho em Grupo	Técnica que pode ser combinada com outras, visando dar experiência em questões específicas do trabalho em equipe, permitindo vivenciar a prática no gerenciamento de tarefas, cumprimento de prazos, liderança de equipes, gestão de conflitos etc. Os grupos podem ser misturados ao longo das várias etapas de um trabalho, forçando assim a convivência entre alunos que não têm maior aproximação no dia a dia.
Seminário	Temas pertencentes à disciplina são distribuídos entre os membros da turma e eles fazem apresentações no âmbito da turma ou mais geral sobre o assunto, podendo ser uma atividade individual ou em grupo.
Experimentação	Os alunos são envolvidos com projetos de pesquisas científicas, visando introduzi-los no mundo científico através dessa integração entre o ensino e a pesquisa.

Quadro 3.2: Lista de ferramental didático

Fonte: adaptado de Bordenave, 2007

Cada uma dessas ferramentas tem a sua especificidade de aplicação, cabendo ao docente a escolha durante a fase de planejamento pedagógico e o seu uso, normalmente,

combinado de várias delas durante o processo ensino-aprendizagem. Essas técnicas serão apropriadas na formatação da metodologia ESCollab.

Dentre essas ferramentas didáticas discutidas anteriormente, merece destaque, no âmbito do ensino de engenharia de *software*, o projeto, pois na abordagem de desenvolvimento de projetos, as atividades são centradas no aluno que, individualmente ou em pequenos grupos, produzem, documentam e avaliam artefatos intermediários para a conclusão de um projeto. Os alunos são estimulados a refletir, formular questões e expressar dúvidas sobre os artefatos produzidos e o professor atua como um orientador dos projetos (FAGUNDES, 1999; SANTORO et al, 2003; BITTENCOURT et al, 2006).

A cada etapa de produção do projeto, os alunos produzem artefatos intermediários que são insumos do projeto como um todo. A turma colabora através da troca de contribuições, conforme são entregues os resultados da produção de cada etapa. Os alunos discutem e refletem sobre os seus próprios artefatos produzidos e sobre os artefatos produzidos pelos demais alunos, o que possibilita desenvolver competências e habilidades, como a autocrítica, negociação, compartilhamento de entendimentos e capacidade argumentativa, aspectos desejáveis na aprendizagem colaborativa (FAGUNDES, 1999, STAHL et al, 2006).

3.2 A IMPORTÂNCIA DA COMUNICAÇÃO NO PROCESSO ENSINO-APRENDIZAGEM

A comunicação é um dos elementos essenciais ao processo ensino-aprendizagem. Nas dez grandes famílias de competências necessárias a(o) educador(a), citadas por Perrenoud (2000, p. 14) e elencadas na lista abaixo, pode-se observar, que várias delas estão intimamente relacionadas à capacidade de comunicação:

- 1) Organizar e dirigir situações de aprendizagem;
- 2) Administrar a progressão das aprendizagens;
- 3) Conceber e fazer evoluir os dispositivos de diferenciação;
- 4) Envolver os alunos em suas aprendizagens e em seu trabalho;
- 5) Trabalhar em equipe;
- 6) Participar da administração da instituição de ensino;
- 7) Informar e envolver a comunidade;
- 8) Utilizar novas tecnologias;
- 9) Enfrentar os deveres e dilemas éticos da profissão;
- 10) Administrar a sua própria formação contínua.

A comunicação professor-aluno dar-se-á pela combinação e seleção dos vários repertórios existentes simultaneamente no emissor (professor) e no receptor (aluno). A Figura 3.3, a seguir, demonstra essa relação entre esses repertórios:

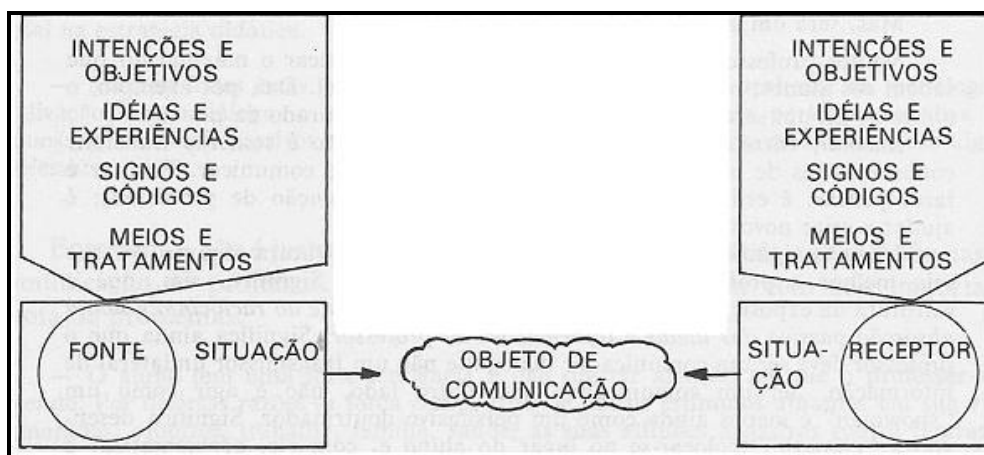


Figura 3.3 – A comunicação como processo de seleção, combinação e intercâmbio de repertórios.
 Fonte: Adaptado de Bordenave e Pereira, (2007, p. 186)

Cada um desses repertórios é composto por conjuntos de intenções, objetivos, idéias, experiências, signos, códigos, meios e tratamentos existentes de cada lado do processo de comunicação.

O significado de uma mensagem emitida pelo emissor (docente) é afetado pela dinâmica interna dos receptores (alunos), sofrendo a influência das experiências, conhecimentos, valores, atitudes e grupos de referência, conforme mostra a Figura 3.4:

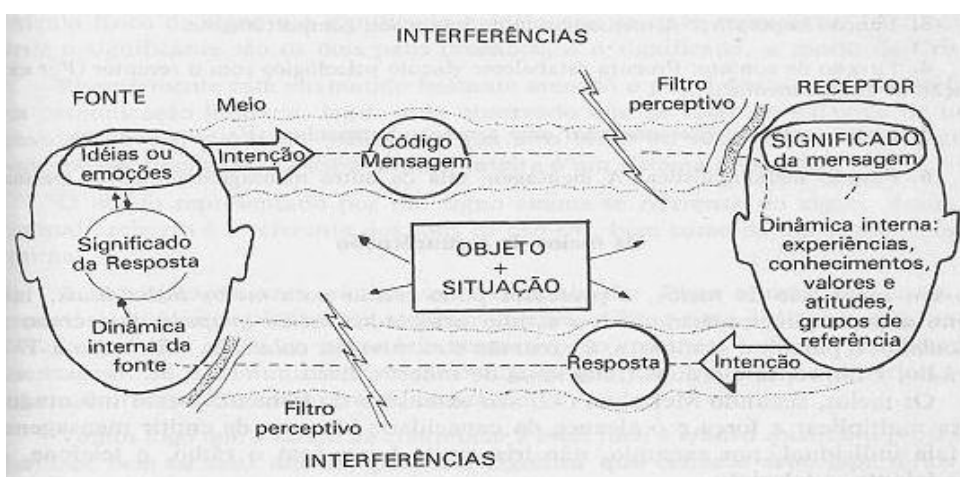


Figura 3.4 – Modelo do processo de comunicação
 Fonte: Adaptado de Bordenave e Pereira (2007, p. 187)

“A emissão, transmissão e recepção de informação, entretanto, é apenas uma das funções da comunicação entre o professor e alunos. Da boa comunicação, dependem não só a aprendizagem, mas também o respeito mútuo, a cooperação e a criatividade” (BORDENAVE e PEREIRA, p. 183).

Segundo Bordenave e Pereira (2007, 186), os problemas de comunicação professor-aluno giram em torno dos seguintes eixos:

- a) Problemas psicológicos relacionados com a percepção, atenção, motivação, atitudes, memória, hábitos de pensamento;
- b) Problemas semiológicos relacionados com o emprego de signos e códigos para comunicar: palavras, gestos, tom de voz, coisas escritas no quadro ou na apresentação;
- c) Problemas semânticos relacionados com o significado das palavras, dos objetos e das pessoas, e sua interpretação;
- d) Problemas sintáticos relacionados com a retro-informação e ao diálogo, com a quantidade de idéias transmitidas por diversos canais e com a capacidade deste para levar sinais.

Bordenave e Pereira (2007) sustentam ainda que esta lista de problemas vem demonstrar a complexidade do processo da comunicação, mas também vem oferecer um caminho para uma solução, que é apelar às ciências básicas: Psicologia, Semiologia, Semântica, Sintática, Cibernética, na procura de subsídios para melhorar nossa ação de comunicar, sendo esta uma questão importante tanto para o ensino presencial quanto para o EaD⁹.

Nesse sentido, Michael Moore (1989), uma das maiores referências mundiais em EaD, desenvolveu um importante conceito de “distância transacional”, estabelecendo que a separação entre professores e alunos, na EaD, tem influência significativa no processo de ensino e aprendizagem. Todavia, a partir dessa distância física convencional e até mesmo temporal, abre-se uma nova dimensão pedagógica e psicológica, na qual ocorre uma forma diferenciada de comunicação, uma nova transação. Essa nova dimensão, surgida a partir do ensino à distância, Moore (1989) convencionou chamar de “distância transacional”.

Sob o ponto de vista da distância transacional, não interessa, portanto, a distância física entre professor e aluno, nem mesmo entre os próprios alunos, e sim as relações pedagógicas e psicológicas que se estabelecem no ensino à distância. Contudo, independente da distância convencional ou temporal, os docentes e os alunos podem estar mais ou menos

⁹ EaD – Ensino à Distância

distantes em EaD, sob o ponto de vista transacional. Para Moore (1989), a distância transacional é inversamente proporcional ao envolvimento dos estudantes e dependente do diálogo, da estrutura e do grau de autonomia do discente.

Essas interações são claramente observadas através de um modelo elaborado por Moore (1989) para explicar essas interações existentes no ensino à distância, podendo ser vista na Figura 3.5:

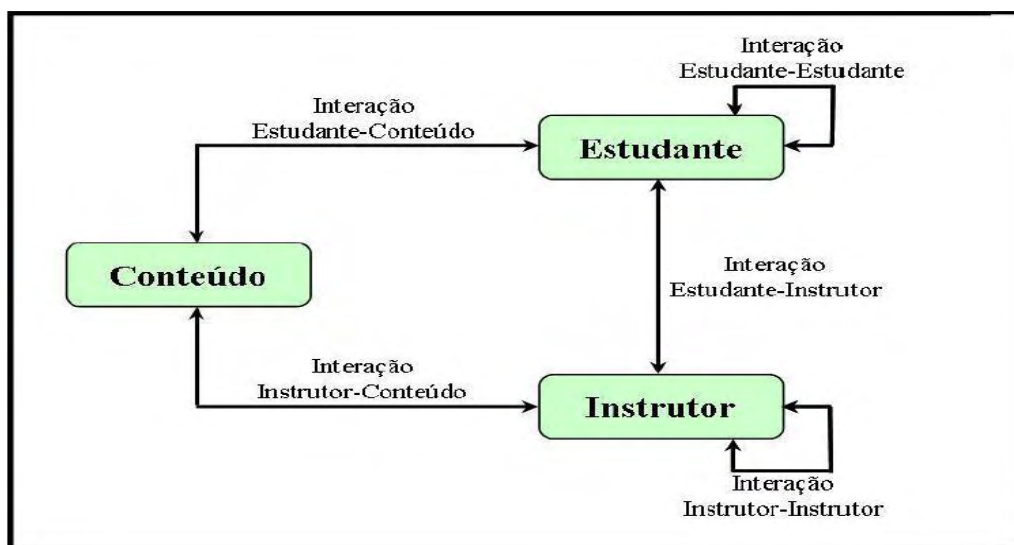


Figura 3.5 – Modelo de interação de Moore
Fonte: Adaptado de Moore, 1989

Essas interações observadas no modelo de Moore (1989), apresentado na Figura 3.5, especificamente para o EaD, também estão presentes no ensino presencial, sendo de fundamental importância para o sucesso do processo ensino-aprendizagem.

Essas interações e suas principais características são resumidas no Quadro 3.3, que segue:

Subsistema	Tipo de interação	Funcionalidades Web
Corresponde às atividades de aprendizado realizadas pelo estudante. Engloba o processo de transmissão de conteúdo e o ferramental de suporte ao aprendizado	Estudante-Conteúdo	<ul style="list-style-type: none"> • Atalhos para sites e materiais educacionais • Apresentação multimídia do conteúdo • Auto-avaliação • Avaliação automatizada (múltipla escolha) • Avaliação Alternativa (discursiva, participativa, colaborativa, etc)

<p>Corresponde à coordenação das atividades de aprendizado dos diversos estudantes, incluindo a provisão de canais de comunicação entre os mesmos</p>	<p>Estudante-Estudante</p>	<ul style="list-style-type: none"> • Individualização da instrução • Individualização da avaliação • Compilação de perguntas mais freqüentes (FAQ) • Ajuda on-line para o conteúdo • Ajuda on-line para o uso do sistema • Guia de estudos • Ferramenta de contribuição para o conteúdo por parte do estudante • Jogos educacionais • Listagem de alunos (acesso a dados pessoais, perfis de preferências, atividades concluídas, cursos matriculados, etc) • Ferramenta de correio eletrônico • Sala de bate-papo (moderada ou não) • Fórum de discussão (moderada ou não)
<p>Corresponde ao gerenciamento e supervisão do aprendizado</p>	<p>Estudante-Conteúdo (Supervisão)</p>	<ul style="list-style-type: none"> • Monitoramento de avaliação de satisfação • Monitoramento de uso do sistema (tarefas concluídas, páginas visitadas, avaliação realizadas, tempo das atividades, etc) • Monitoramento da performance acadêmica
<p>Corresponde às atividades de autoria de novos conteúdos, de melhoria contínua do conteúdo existente e de adaptação do mesmo às necessidades individuais dos estudantes.</p>	<p>Instrutor-Conteúdo</p>	<ul style="list-style-type: none"> • Ferramenta de autoria de conteúdo • Ferramenta de atualização e manutenção do conteúdo • Monitoramento de avaliação de satisfação • Monitoramento de uso do sistema • Monitoramento de performance acadêmica • Lista de verificação de atividades (interações instrutor-instrutor envolvendo múltiplos desenvolvedores)

Quadro 3.3: Lista das funcionalidades Web requeridas categorizadas por tipo de interação
Fonte: Adaptado de Anderson, 2003

Observa-se que esses conceitos, que originalmente foram voltados para o ensino à distância, também têm validade e aplicabilidade na sala de aula convencional.

3.3 A APRENDIZAGEM COLABORATIVA

Essa busca de melhorias na comunicação tem levado os educadores à apropriação de recursos tecnológicos na tentativa de incrementar a qualidade da interação dos membros do processo educacional. O professor, em geral um migrante digital, precisa transformar-se para potencializar a comunicação com os seus alunos, originalmente, nativos digitais.

Segundo Paulo Freire (2005 p. 34), “o professor que realmente ensina, quer dizer, que trabalha os conteúdos no quadro da rigorosidade do pensar certo, nega, como falsa a fórmula farisaica do faça o que mando e não o que eu faço. As palavras às quais faltam a força do exemplo, pouco ou quase nada valem. Pensar certo, é fazer certo”. Neste sentido, vale salientar que a formação crítica pode se apoiar na tecnologia ou no novo como descreve Libâneo (LIBÂNEO, 2002, p. 67) “É certo que as práticas docentes recebem o impacto das novas tecnologias da comunicação e da informação, provocando uma reviravolta nos modos mais convencionais de educar e ensinar. (...) a utilização pedagógica das tecnologias da informação pode trazer efeitos cognitivos relevantes, estes todavia não podem ser atribuídos apenas à tecnologia”.

Perrenoud (2000, p.128) observa que, formar para novas tecnologias é formar o julgamento, o senso crítico, o pensamento hipotético e dedutivo, as faculdades de observação e de pesquisa, a imaginação, a capacidade de memorizar e classificar, a leitura e a análise de textos e de imagens, a representação de redes, de procedimentos e de estratégias de comunicação.

É preciso combater a ideia do professor como o único irradiador de conhecimentos apregoado pela educação bancária. O professor ainda é um ser superior que ensina a ignorantes. Isto forma uma consciência bancária. O educando recebe passivamente os conhecimentos, tornando-se um depósito do educador. Educa-se para arquivar o que se deposita. Mas o curioso é que o arquivado é o próprio homem, que perde assim o seu poder de criar, se faz menos homem, é uma peça. O destino do homem deve ser criar e transformar o mundo, sendo sujeito de sua ação (FREIRE, 1979, p. 38).

Segundo Perrenoud (2000, p.14) “É natural pensar que com o trabalho colaborativo pode-se obter melhores resultados, pelo menos potencialmente, do que se os integrantes de um grupo atuassem individualmente. Um aspecto interessante em um grupo é o fato de que pode haver uma complementação de habilidades, conhecimentos e esforços, além da interação entre indivíduos que possuem capacidades, entendimentos e pontos de vista complementares. Através da colaboração, os membros de uma equipe podem identificar mais rapidamente possíveis inconsistências e falhas durante o processo de realização de uma tarefa e, conjuntamente, podem buscar idéias e informações que ajudem no processo de resolução de problemas”.

Fuks *et all* (FUKS; RAPOSO; GEROSA, 2002) reforça a necessidade de uma postura colaborativa por parte dos docentes e nos lembra que o “individualismo dos professores começa, de algum modo, com a impressão de que cada um tem uma resposta pessoal e original a questões como: O que é ensinar? O que é aprender?”.

Essa visão pode ser consubstanciada na Figura 3.6 extraída da tese de Domenico Gendarmi (2010), demonstrando, simplificada, os mecanismos de compartilhamento do conhecimento de forma colaborativa e social.

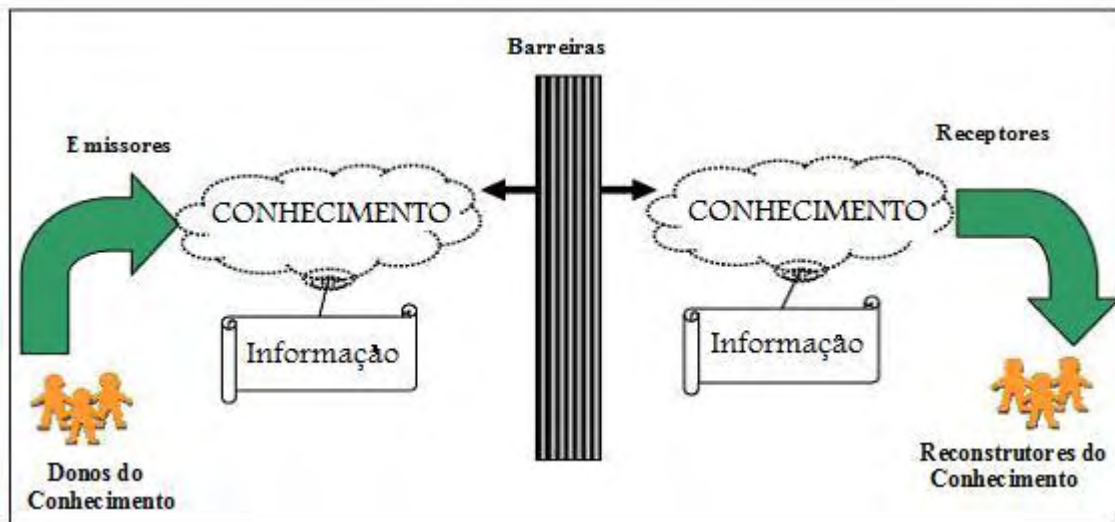


Figura 3.6 – Modelo simplificado de compartilhamento do conhecimento

Fonte: adaptado de Gendarmi (2010, p.16)

Os envolvidos no processo de comunicação enfrentam barreiras para conseguirem estabelecer a troca de conhecimento, permitindo a democratização do processo de produção e disseminação da informação na sociedade contemporânea.

A aprendizagem colaborativa é um processo compartilhado de construção de significados a partir das interações entre os participantes que podem ser mediadas por sistemas computacionais (DILLENBOURG, 1996, STAHL et al, 2006). Na aprendizagem colaborativa, os alunos constroem conhecimento a partir da reflexão das discussões em grupo, que estimulam o interesse e o pensamento crítico, o que possibilita alcançar melhores resultados que no aprendizado isolado (SANTORO E PIMENTEL, 2009).

Aprendizagem colaborativa com Suporte Computacional (CSCL) é a área que investiga como as pessoas podem aprender em grupo com o auxílio do computador através de ambientes interativos para aprendizagem em grupo em que sejam realizadas atividades tanto com objetivos cognitivos como interação social (STAHL et al, 2006)

Segundo Ivan e Ciurea (2009), um sistema colaborativo é aquele onde um grupo de usuários trabalha envolvido em uma atividade compartilhada, geralmente a partir de localidades remotas. Em meio à grande atualidade e capilaridade das aplicações distribuídas, os sistemas colaborativos assumem um papel de relevância cada vez maior.

Dessa forma, os sistemas colaborativos ganharam importância quando se passou a entender que o ambiente de trabalho exige a realização articulada de tarefas por pessoas que integram uma equipe. Com a evolução dos sistemas colaborativos, definidos ainda como ferramentas de software usadas em redes de computadores com o objetivo de tornar mais fácil a execução de tarefas em grupos, a colaboração em equipes distribuídas geográfica e temporalmente tornou-se possível, tendo a contribuição também da popularização da Internet e dos dispositivos móveis (ALMEIDA; BARANAUSKAS, 2008).

Queremos enfatizar que há uma necessidade maior de integração em todas as fases do processo de desenvolvimento de software. Tanto no nível operacional, onde as ferramentas devem prover alguma forma de troca de informações, até o nível conceitual onde há um modelo ou metamodelo que estabeleça diretrizes para o efetivo compartilhamento e colaboração. No caso específico do desenvolvimento de software, apenas para citar um exemplo concreto não basta existir um conjunto de ferramentas, como ilustra o Quadro 3.4, as funcionalidades tais como Controle de Versão, Gerenciamento de Bugs, Gerador de Código, etc, precisam ser especificadas em modelo ou metamodelo para que a cooperação realmente aconteça.

Exemplos de sistemas colaborativos largamente utilizados no desenvolvimento de *software* podem ser vistos no Quadro 3.4:

CDE	Controle Versão	Gerenciamento de “bugs”	Ferramentas de Construção	Gerenciador Modelos	Gerenciador Conhecimento	Ferramentas Comunicação	Aplicações Web 2.0
SourceForge (www.sourceforge.net)	CVS ¹⁰ , SVN ¹¹ , Git ¹² , Mercurial ¹³ , Bazaar ¹⁴	Bugs, controle requisições, caminhos, solicitações de suporte	Não	Não	Não	Listas e fóruns	Blogs, micro-blogs, Wiki e Feed
GForge (Gforge.org)	CVS, SVN e Perforce ¹⁵	Tarefas, Bugs, controle requisições, caminhos, solicitações de suporte	Integração c/ CruiseControl ¹⁶	Não	Gerenciador Documentos	Listas e fóruns	Feed e Wiki
Trac (Trac.edgewall.org)	SVN, plugins p/ integração com Git, Perforce, Mercurial, Darcs ¹⁷ , Bazaar	Tarefas, requisições, bugs e suporte à edição	Plug-ins p/ integração: Continuum ¹⁸ , CruiseControl, Hudson ¹⁹	Project Roadmap ²⁰	Como Wiki	Plugins para fóruns	Wiki, Feed, plugins para sistemas de TAggs e Wikis
Google Code (code.google.com)	SVN, Git, Mercurial	Defeitos, melhorias e tarefas	Não	Não	Como Wiki	Integração com Google Groups	Wiki e Feed
Assembla (www.assembla.com)	SVN, Mercurial e Git	Defeitos, melhorias, tarefas e idéias	Não	Milestones e Agile Planner ²¹	Como Wiki	Chat e Mensagem board	Wiki, microblog e Feed

¹⁰ nongnu.org/cvs

¹¹ tortoise.svn.net

¹² git-scm.com

¹³ mercurial.selenic.com

¹⁴ bazaar.canonical.com

¹⁵ perforce.com

¹⁶ cruisecontrol.sourceforge.net

¹⁷ darc.net

¹⁸ continuum.apache.org

¹⁹ www.hudson-ci.net

²⁰ project-roadmap.com

²¹ agileplanner.codeplex.com

CDE	Controle Versão	Gerenciamento de “bugs”	Ferramentas de Construção	Gerenciador Modelos	Gerenciador Conhecimento	Ferramentas Comunicação	Aplicações Web 2.0
Rational Team Concert (jazz.net/projects/rational-team-concert)	Próprio	Defeitos, melhorias, planejamento de itens, retrospectivas, riscos, casos de uso, itens construídos	Própria e integração com Ant ²² e Maven ²³	Process Template	Como Wiki	Instant Messaging	Feed, wiki e taggs
GitHub (github.com)	Git	Edições	Não	Não	Integração com MS Share-Point ²⁴ e Lotus Quickr Document ²⁵	Não	Feed, wiki e redes sociais
Lauchpad(launchpad.net)	Bazaar	Bugs; integração com rastreadores externos	Não	Blueprints ²⁶ (características e processos)	Perguntas e respostas	Listas de e-mail	Não
CodePlex(www.codeplex.com)	Próprio	Características, tarefas e edições	CruiseControl.net	Documentation Tab	Como Wiki	Listas de e-mail e listas de discussão	Feed e Wiki

Quadro 3.4 – Principais ambientes de desenvolvimento colaborativo

Fonte: Adaptado de Lanubille et al. (2010, p.54)

²² ant.apache.org

²³ maven.apache.org

²⁴ sharepoint.microsoft.com

²⁵ www.ibm.com/developerworks/lotus/products/quickr

²⁶ www.blueprintcss.org

Essas ferramentas têm sido utilizadas por times de desenvolvimento distribuídos geograficamente ao redor do mundo para otimizar o processo de construção de aplicações. A comunidade acadêmica tem também feito uso desses recursos na tentativa de aproximar os discentes da realidade do mercado de desenvolvimento de *software*.

3.4 REDES SOCIAIS

Uma rede social é uma estrutura social constituída por pessoas ou organizações, representada na *internet*, apresentando diversos tipos de relações com a finalidade de compartilhar valores e/ou objetivos em comum.

Facebook²⁷, Orkut²⁸, Google+²⁹ e LinkedIn³⁰ são exemplos de redes sociais que aglutinam milhões de pessoas em todo o mundo, facilitando os contatos com propósitos diversos, que variam desde o mero conhecimento às atividades profissionais.

O Quadro 3.5 apresenta as principais características, os números estimados de usuários e o posicionamento em um serviço de classificação:

Rede Social	Fundação	País de Origem	Número Usuários (Milhões)	Posição no Alexa ³¹	URL
Facebook	2004	EUA	750	2	www.facebook.com
Orkut	2004	EUA	30	59	www.orkut.com
LinkedIn	2003	EUA	20	16	www.linkedin.com
Google+	2011	EUA	- ³²	- ³³	www.plus.google.com
Twitter	2006	EUA	20	8	www.twitter.com

Quadro 3.5 – Principais redes sociais e suas características

Fonte: Adaptado de www.wikipedia.org. (Acesso 08 jul. 2011).

Segundo Costa (2008), as redes sociais representam os relacionamentos sociais como nós (ou elos) e ligações entre estes nós, de modo que cada nó representa um ator no ambiente e as ligações representam relações entre esses atores, as quais podem se consolidar de várias

²⁷ www.facebook.com

²⁸ www.orkut.com

²⁹ www.plus.google.com

³⁰ www.linkedin.com

³¹ Alexa – Serviço que avalia a popularidade de websites: www.alexa.com

³² Não há dados disponíveis devido ao seu lançamento recente

³³ Idem

formas, cada uma representando diferentes contextos com relação à rede social. As redes sociais também podem ser entendidas como grandes depósitos de dados que são responsáveis pelo armazenamento de informações sobre cada usuário que faz parte da rede.

Já Torquato (TORQUATO, 2009) enxerga as redes sociais como “uma das formas de representação dos relacionamentos entre seres humanos, sejam afetivos ou de caráter profissional, ou entre grupos de interesses comuns. Esses ambientes funcionam por meio da interação social objetivando a conexão e comunicação entre as pessoas, podendo inclusive ser uma ferramenta muito importante para o desenvolvimento de comunidades científicas”.

É possível, nesse tipo de ambiente, a criação de comunidades virtuais através das quais as pessoas são conectadas para aprender e criar mais e interagir com pessoas que apresentem interesses afins. As redes sociais facilitam a interação social, a aprendizagem colaborativa e o trabalho cooperativo, demonstrando ser uma importante contribuição para o desenvolvimento da pesquisa científica (ARCOVERDE; MELO; FRANCO, 2008).

No contexto atual do ensino básico e universitário, é bastante perceptível a utilização das redes sociais como ferramentas didáticas no processo ensino-aprendizagem, permitindo a melhoria da comunicação entre alunos versus alunos e professores versus alunos.

A abordagem de Arcoverde, Melo e Franco será utilizado como âncora nesta dissertação, potencializando a interação, a aprendizagem colaborativa e o trabalho cooperativo, no âmbito do ensino-aprendizagem em ES.

CAPÍTULO IV

METODOLOGIA *ESCollab*

4.1 INTRODUÇÃO

A OCDE (2001) analisa que os docentes não possuem um corpo de conhecimento científico codificado sobre o ensinar e aprender. Ao invés disso, os docentes trabalham em contextos individualizados e adquirem seu conhecimento por tentativa e erro. Dessa forma, seu conhecimento é pessoal ao invés de coletivo, tácito ao invés de explícito e baseado em matérias/conteúdo ao invés de ser baseado em processos.

“Professores (...) precisam agora ensinar aos estudantes a apreender como aprender e isso requer a produção e aplicação de novos conhecimentos pedagógicos em grande escala” (OCDE, 2001:71).

De um modo geral, as dificuldades atribuídas ao ensino de Engenharia de *Software* não são fáceis de serem suplantadas, mas certamente podem ser amenizadas com o uso de práticas metodológicas adequadas.

A expressão "Metodologia de Ensino" está atrelada à área de ensino e busca descrever os melhores métodos e técnicas para que o processo ensino-aprendizagem possa ser desenvolvido com maior eficiência.

Os métodos podem ser usados em várias áreas, podendo ter ajustes específicos para casos particulares. As técnicas, comumente, estão relacionadas com a prática, a ação na qual o objetivo maior é a compreensão do ensino.

Na nossa visão, metodologia é um conjunto de técnicas, métodos e ferramentas utilizados de forma racional e seqüenciada. Esse conceito também pode ser visto como uma subárea da pedagogia que procura descrever, pesquisar e justificar os melhores métodos e técnicas de determinada área de ensino. Sendo assim, cada área pode ter a sua metodologia específica.

Sobre a aplicação de uma nova visão metodológica, podemos buscar em Gasparin (2005, p.3) “essa nova forma pedagógica de agir exige que se privilegiem a contradição, a dúvida, o questionamento; que se valorizem a diversidade e a divergência; que se interroguem certezas e as incertezas, despojando os de sua forma naturalizada, pronta, imutável (...) deste enfoque, defende-se o caminhar da realidade social, como um todo, para a especificidade teórica da sala de aula e desta para a totalidade social novamente, tornando possível um rico processo de trabalho pedagógico (...) para fundamentar a concepção metodológica e o planejamento do ensino-aprendizagem, como a ação docente-discente”.

Neste capítulo, propõe-se uma metodologia de ensino que possa nortear os processos de elaboração, manutenção e oferta de cursos voltados para ministrar os conteúdos programáticos da área de engenharia de *software*. Essa metodologia tem o seu foco na colaboração, embasando-se nos princípios da cooperação, comunicação e controle.

A abordagem proposta tem a preocupação de introduzir práticas didáticas com estratégias multifacetadas para tornar o ensino desses conteúdos da engenharia de *software* mais atraentes e interativos de tal forma que os alunos possam, envolvendo-se desde o início do curso, aprender mais e melhor.

A metodologia de ensino proposta incorporará aulas expositivas, aulas práticas de laboratório, sessões de soluções de problemas, discussões abertas e interativas através das redes sociais, implementação de mini-projetos, aplicação de mini-testes, discussões de estudos de casos, promoção de concursos, uso de analogias e o compartilhamento de código fonte, dentre outras ferramentas didáticas.

Para formatar a metodologia norteadora do presente trabalho, utiliza-se como balizamento programático as diretrizes curriculares do MEC (2005), as referências curriculares do IEEE/ACM (2001), as referências curriculares da SBC (2005) e as definições do Guia de Conhecimento em Engenharia de *Software* (IEEE, 2004). Do ponto de vista da fundamentação em termos de teorias da aprendizagem, a ESCollab ancora-se em aspectos do Construtivismo de Bruner (1996), da Teoria Sócio-Cultural de Vygotsky (DANIELS, 2003) e na Teoria da Cognição Distribuída de Hutchins (1995) e Rogers (1997).

De acordo com Lanubile et al. (2010), “a engenharia de *software* envolve pessoas colaborando para desenvolver *softwares* melhores. A colaboração é um desafio, especialmente por envolver pessoas localizadas em locais diferentes e sem a possibilidade de encontros presenciais”.

A metodologia de ensino proposta ancora-se no modelo de colaboração 3C ilustrada na Figura 4.1:

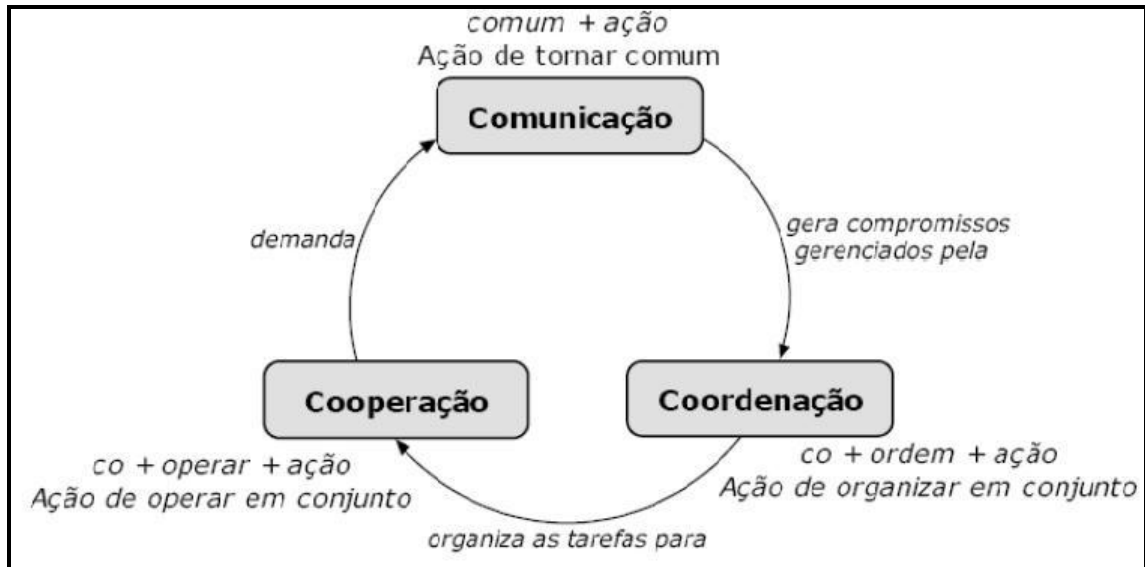


Figura 4.1: Modelo de colaboração 3C
Fonte: Adaptado de Pimentel, 2006; Fillippo et. al., 2007

Sobre o modelo 3C, explica Mangueira (2009), “no processo de colaboração, os indivíduos têm que intercambiar informações (comunicação), organizar-se em uma estrutura bem definida (coordenação) e trabalhar em conjunto dentro de um espaço compartilhado (cooperação)”.

A metodologia em questão, especificamente, para o ensino de engenharia de *software* baseia-se em sete princípios básicos:

- i. **Simplicidade** – buscar soluções simples e de fácil implementação para facilitar o trabalho no processo ensino-aprendizagem;
- ii. **Comunicação** – a interação permanente entre os diversos agentes envolvidos nas disciplinas integrantes da área de Engenharia de *Software*, tendo como suportes básicos as redes sociais e um ambiente *Web* voltado para tanto. Os agentes participantes do processo poderiam ser: professores de ES, alunos atuais, ex-alunos, professores de outras instituições e profissionais da área.
- iii. **Cooperação** – aproveitamento do conhecimento produzido ao longo do processo ensino-aprendizagem em cada disciplina atualmente sendo ministrada e passadas, dentro da IES e nas demais instituições participantes do processo, materializando o conceito de inteligência múltipla e cooperativa; máxima: tudo é público, nada é privado;

- iv. **Coordenação** - está relacionada à ação de organizar, arranjar, dispor algo sob determinada ordem e método. A atividade de coordenação inclui a identificação de objetivos, o mapeamento destes em tarefas e sua distribuição entre os participantes, sendo essas tarefas realizadas na ordem e tempo previstos e em conformidade com os objetivos e restrições determinados (RIQUE, 2011);
- v. **Transparência** – os cursos deixam de ser mundos fechados restritos apenas ao binômio professor-aluno e passam a estar abertos à supervisão, ao acompanhamento e à participação da comunidade interna e externa da IES;
- vi. **Responsabilidade social** – o processo ensino-aprendizagem deve estar conectado com as demandas sociais, produzindo necessariamente retorno para a sociedade, notadamente para aqueles que mais precisam, direcionando a ação e os subprodutos do ensino para atender às necessidades da sociedade;
- vii. **Reutilização** – os artefatos produzidos por um docente poderão ser utilizados pelos demais docentes da área dentro da própria IES ou por professores de outras instituições, permitindo o compartilhamento de materiais didáticos e conseqüentemente a redução do esforço na montagem dos cursos.

Para viabilizar esses princípios, propõe-se o que se convencionou chamar de “postura cooperativa” (SILVA, 2003), conforme ilustrado na Figura 4.2:

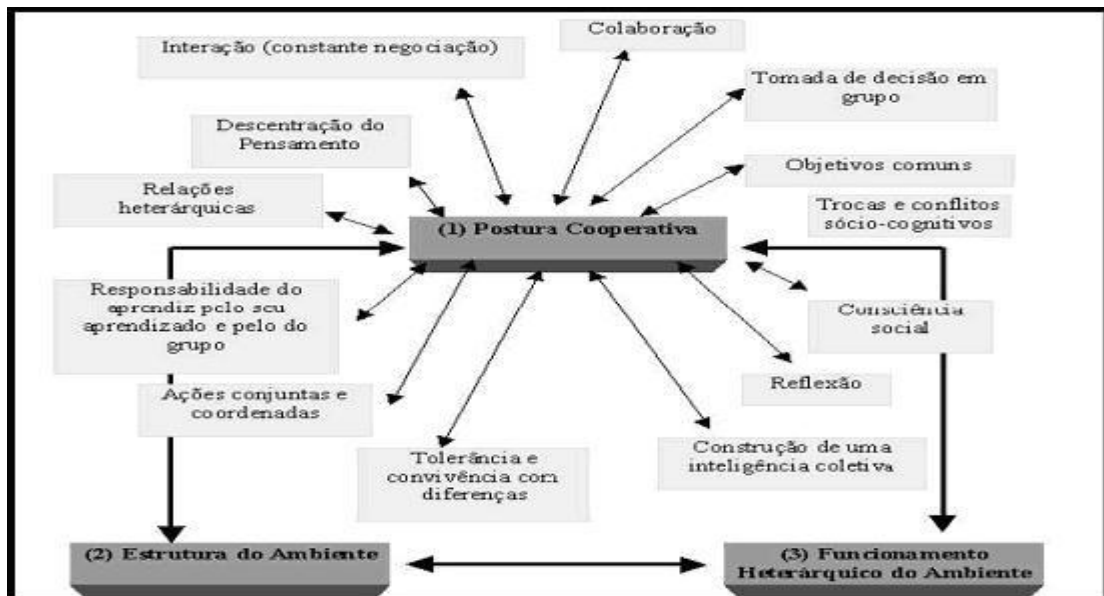


Figura 4.2: Postura Cooperativa
Fonte: Adaptado de Silva, 2003

Pode-se observar na Figura 4.2 a presença de elementos relevantes para o presente trabalho, tais como: como a “construção de uma inteligência coletiva”, “colaboração”, “responsabilidade do aprendiz pelo seu aprendizado e pelo aprendizado do grupo” e “trocas e conflitos sócio-cognitivos”.

A postura colaborativa no âmbito da educação introduz novos comportamentos para o ambiente de aprendizagem. O Quadro 4.1 confronta a aprendizagem tradicional com a aprendizagem colaborativa:

CARACTERÍSTICA	APRENDIZAGEM TRADICIONAL	APRENDIZAGEM COLABORATIVA
Localização	Sala de Aula	Ambiente de aprendizagem
Postura do Professor	Autoridade	Orientador
Referencial	Professor	Aluno
Visão do aluno	“Uma garrafa a encher”	“Uma lâmpada a iluminar”
Postura do educando	Reativa, passiva	Proativa, investigativa
Ênfase Maior	No produto	No processo
Forma de aprendizagem	Isolada	Grupal
Forma de aprendizagem com foco	Memorização	Transformação

Quadro 4.1 – Comparação entre a aprendizagem tradicional e a colaborativa
Fonte: Adaptado de Silva, 2006

Percebe-se, pelos elementos contrastados na comparação feita no Quadro 4.1, que o professor deixa de ser o único irradiador do conhecimento e que o aluno passa a ser um elemento ativo na construção do conhecimento grupal e no processo de aprendizagem individual e grupal.

Em suma, levando em conta a concepção de Nébias (s.d.), “a metodologia de ensino não se resume ao passo que devem ser dados pelo docente em sala de aula, nem aos meios ou ferramentas que são necessários para que o aluno se aproprie do conhecimento. É preciso apreendê-la como a orientação global que permeia o trabalho pedagógico, dando-lhe coerência, sentido e perspectiva. Nessa linha, o planejamento será o elemento mediador para a obtenção de uma ação consciente, intencional e eficiente na busca do objetivo maior que é a aprendizagem efetiva”.

4.2 ESPECIFICAÇÃO DA METODOLOGIA ESCOLLAB

Neste ponto, passa-se a formatar a metodologia ESCollab e suas fases.

4.2.1 Definições da Metodologia ESCollab:

A metodologia ESCollab é subdividida em 3 fases distintas nas quais são definidas as entradas, saídas e participantes de cada etapa:

- 1) **Planejamento:** é a fase inicial da metodologia correspondendo ao planejamento da(s) disciplina(s) a ser(em) ministrada(s). O planejamento deve acontecer entre os docentes pertencentes à subárea de ES.
 - a) **Entradas:** PPC – Projeto Pedagógico de Curso; Recomendações IEEE/ACM e SBC; Diretrizes Curriculares do MEC; calendário escolar; levantamento bibliográfico na área/disciplina.
 - b) **Saídas:** Ementa(s), programa(s) da(s) disciplina(s), Bibliografia básica/complementar, planejamento de curso, formulário de auto-avaliação discente e avaliações diagnósticas inicial (a ser aplicada no início das aulas) e final (a ser aplicada no final das aulas). No planejamento de curso devem ser explicitados os recursos didáticos e estratégias a serem utilizadas ao longo da disciplina, devendo ser respeitadas as determinações quanto à estratégia e ferramental obrigatório da metodologia.
 - c) **Participantes:** docente titular da disciplina, docente(s) voluntário(s), coordenação de curso, supervisão pedagógica, membro(s) da sociedade civil.
- 2) **Execução: é a fase na qual as aulas e atividades didáticas ocorrem.**
 - a) **Entradas:** Ementa, programa da disciplina, plano de curso, Bibliografia básica/complementar, avaliação diagnóstica.
 - b) **Saídas:** Resultado da avaliação Diagnóstica Inicial, Plano de curso ajustado, registro das atividades didáticas, controle de frequência, resultado das avaliações, artigos, projetos concluídos, seminários apresentados, problemas resolvidos, Estudos de casos, listas de exercícios resolvidas, avaliações aplicadas, correções das avaliações aplicadas.
 - c) **Participantes:** docente titular da disciplina, docente(s) voluntário(s), membro(s) da sociedade civil e aluno(a)s.

3) Conclusão: corresponde à fase final da disciplina.

- a) **Entradas:** Resultado da avaliação Diagnóstica Inicial, Plano de curso ajustado, registro das atividades didáticas, controle de frequência, resultado das avaliações, artigos, projetos concluídos, seminários apresentados, problemas resolvidos, Estudos de casos, listas de exercícios resolvidas, avaliações aplicadas, correções da avaliações aplicadas.
- b) **Saídas:** Avaliação diagnóstica final, análise comparativa da evolução da aprendizagem da turma, avaliação da disciplina/docente, relatório da auto-avaliação dos discentes. Relatório final da disciplina
- c) **Participantes:** docente titular da disciplina, docente(s) voluntário(s), membro(s) da sociedade civil, aluno(a)s, coordenação de curso, supervisão pedagógica.

Esquemáticamente, a metodologia ESCollab pode ser visualizada através da Figura 4.3:

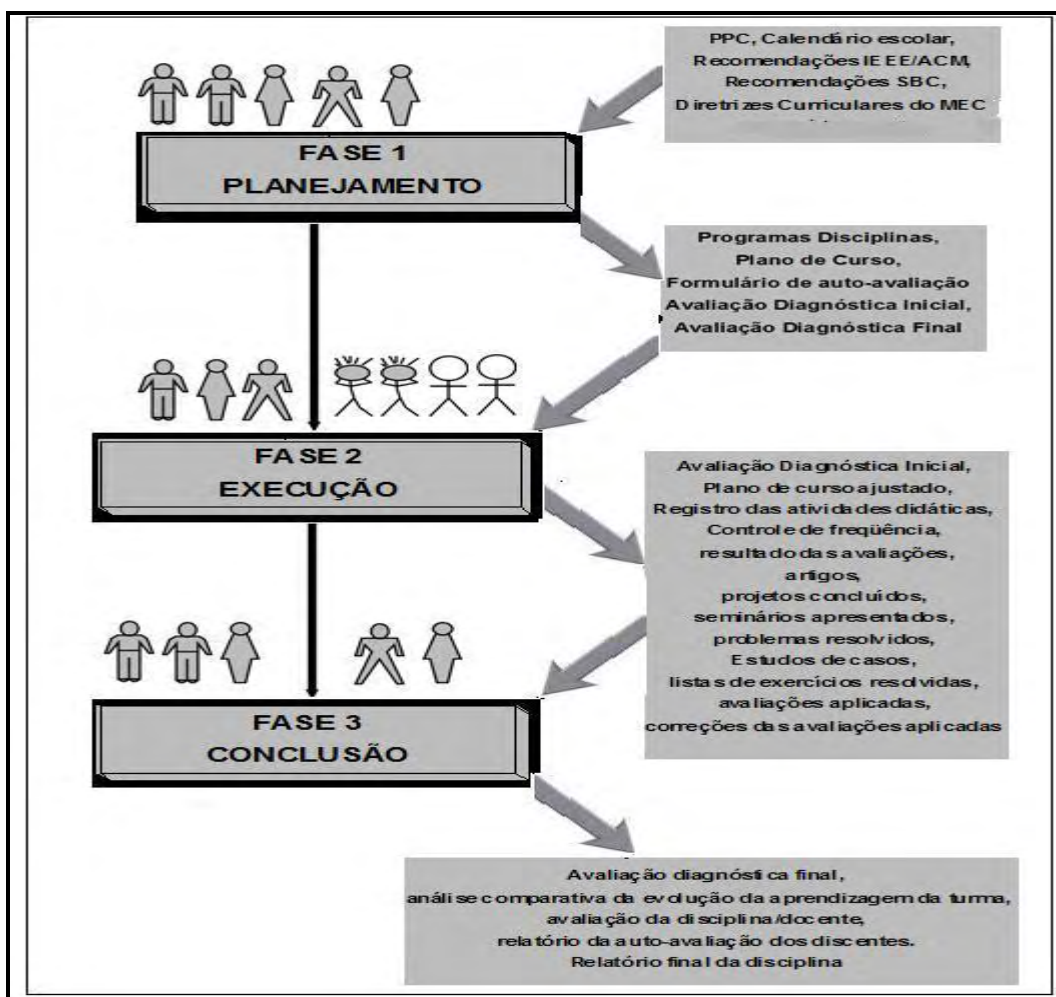


Figura 4.3: Metodologia ESCollab e suas fases

Através da Figura 4.3, pode-se ver graficamente cada uma das fases da metodologia ESCollab com as suas entradas, saídas e participantes envolvidos em cada uma dessas fases.

A ESCollab está pautada em dois pilares: o primeiro é a necessidade crescente de cooperação entre agentes p_i e p_j que ministram disciplinas $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$ que tem relação de pré-requisitos entre $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$; o outro é o apelo tecnológico para reaproveitar artefatos no processo de forma que crie uma cultura de reaproveitamento, e conseqüente, cooperação entre os agentes.

A ESCollab elenca um conjunto de conselhos para que os agentes possam planejar suas atividades de forma cooperativa, utilizando os recursos da web 2.0, proporcionando um repositório de objetos de aprendizagem voltado para o ensino da Engenharia de *Software*.

O fundamental é que p_i e p_j sintam a necessidade de fazerem um planejamento em conjunto ao ministrarem as disciplinas $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$. Neste sentido, esses agentes devem especificar um mesmo conteúdo semântico para alguns artefatos:

- a) Toda Aula AL é composta por um conjunto de Recursos Didáticos - RD; que deve ser de domínio público; e deve estar na Web;
- b) Um recurso pode ser um vídeo, um áudio, um texto, uma apresentação, um artigo científico, uma aplicação, ou um objeto de aprendizagem disponível na Web;
- c) Os recursos associados às disciplinas de G, tais como $\langle e_i, c_i \rangle$, devem ser públicos, possuir uma página web (contendo a e_i -Ementa; e o c_i – Conteúdo Programático, Plano de aulas a ser executado ao longo do período letivo; Links ou referências para os recursos associados; Links para os mecanismos de avaliação; Informações sobre Avaliações (Contínua e Somativa) e controle de frequência e/ou participação; Informações sobre os projetos, Problemas e Estudos de casos associados à disciplina;
- d) O conjunto de alunos matriculados em uma disciplina $\langle e_i, c_i \rangle$ compõe o corpo discente, onde: cada membro do corpo discente deve possuir endereço eletrônico (e-mail) de domínio público; ser membro de uma rede social comum aos demais discentes e docentes; possuir acesso a uma página da disciplina que deve ser atualizada coletivamente por alunos e docentes;
- e) Cada disciplina $\langle e_i, c_i \rangle$ deve possuir no mínimo, um colaborador voluntário da sociedade civil organizada, atuando tanto como especialista que vai trazer as histórias do usuário para determinação do problema como também, podendo atuar na comunidade de desenvolvedores colaboradores com o projeto a ser desenvolvido;

- f) A disciplina deve adotar uma abordagem didático-pedagógica semelhante às melhores propostas encontradas na pesquisa feita por Paiva (2011a):
 - i. Empregar a técnica PBL (aprendizagem baseada em problemas);
 - ii. Realizar Estudos de Caso – EC sobre os conteúdos da disciplina;
 - iii. Produzir Artigo(s) Científico(s) – ARTC como mecanismo de ensino-aprendizagem;
 - iv. Apresentação de seminários sobre temas relevantes à disciplina;
- g) Os mecanismos de avaliação devem fazer parte do planejamento e podem ser: mini-testes, avaliações objetivas e/ou avaliações subjetivas, estudo dirigido, projeto, estudos de caso, artigo científico, seminários, problemas, conforme (BORDENAVE, 2007);
- h) Um projeto deve estar associado a cada disciplina $\langle ei, ci \rangle$ com o objetivo de materializar conceitos abordados ao longo do curso. O domínio do problema do projeto deve ter um usuário real (que pode ser entidades públicas (creches, hospitais, orfanatos, escolas, sindicatos, cooperativas, associações de moradores, como sugestão para ficar mais próximo do lado social); ter o seu código fonte disponível em um ambiente colaborativo de desenvolvimento com escolha livre pelo grupo (CDE - Collaborative Development Environment) para receber a participação/contribuição das comunidades de desenvolvedores de *software*;
- i) É fundamental na ESCollab que os agentes pi e pj planejem conjuntamente os seus cursos e as formas de avaliar o conteúdo trabalhado com os alunos das disciplinas $\langle ei, ci \rangle$ e $\langle ej, cj \rangle$, que têm relação de pré-requisitos entre $\langle ei, ci \rangle$ e $\langle ej, cj \rangle$;
- j) A cada hora aula semanal presencial da disciplina deve haver a publicação de uma contribuição ou Recurso Didático (resolução de exercício, comentário sobre alguma atividade etc), via redes sociais, por parte dos alunos e docentes envolvidos no processo ensino-aprendizagem, totalizando um mínimo de 20% da carga horária total da disciplina.

A Figura 4.4, em seguida, apresenta de forma simplificada a representação do processo ensino-aprendizagem conduzido sob a ótica da metodologia ESCollab:

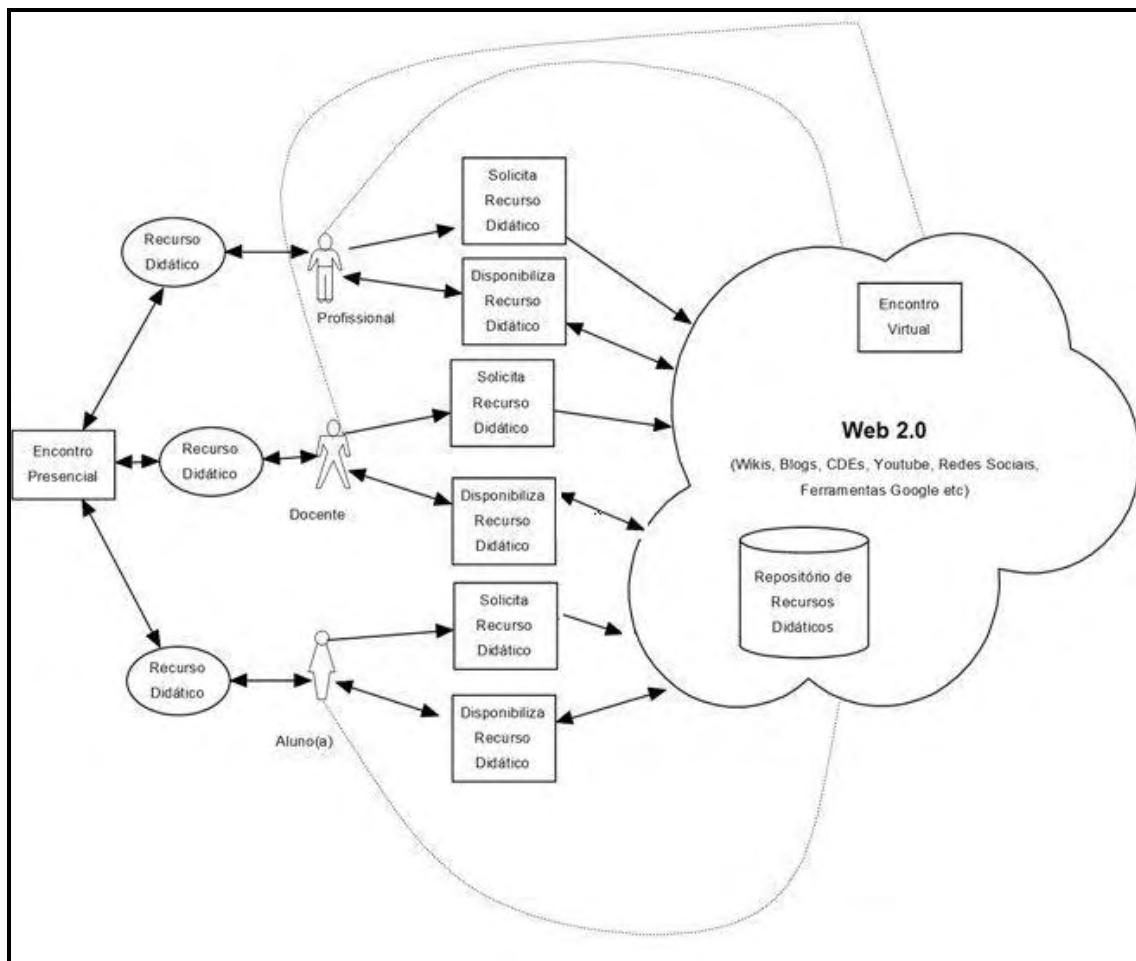


Figura 4.4: O processo ensino-aprendizagem sob a ótica da Metodologia ESCollab

Através da Figura 4.4, pode-se observar a ocorrência dos encontros presenciais entre docente e alunos, sendo possível também a participação eventual do(s) profissional(ais) representante(s) da sociedade civil. Além dos encontros presenciais, devem ocorrer encontros virtuais através das redes sociais, correspondendo a um mínimo de 20% (vinte por cento) da carga horária da disciplina. Registre-se o armazenamento dos recursos didáticos exclusivamente na Internet, utilizando recursos da Web 2.0 (wikis, CDEs, Youtube, redes sociais, ferramentas Google etc).

A Figura 4.5 exhibe o protocolo de funcionamento da Metodologia ESCollab:

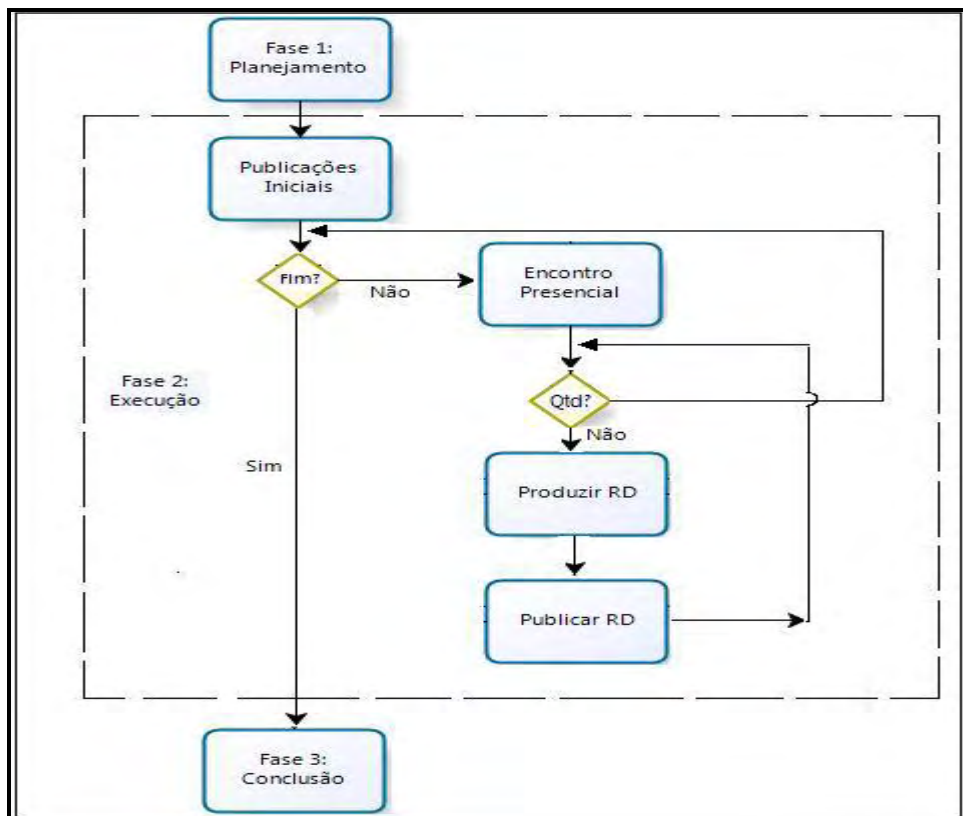


Figura 4.5: Protocolo de funcionamento da Metodologia ESCollab

De acordo com a Figura 4.5, enquanto a disciplina estiver em funcionamento, existirão encontros presenciais, que devem ser seguidos da produção e/ou pesquisa de recursos didáticos, que devem ser publicados por alunos e professores na Web 2.0 na quantidade proporcional à carga horária semanal da disciplina.

Visando dar uma representação mais formal à descrição da metodologia proposta, a Figura 4.6 apresenta o processo de funcionamento da metodologia ESCollab, utilizando a notação BPMN³⁴ (OMG, 2009):

³⁴ Business Process Model and Notation

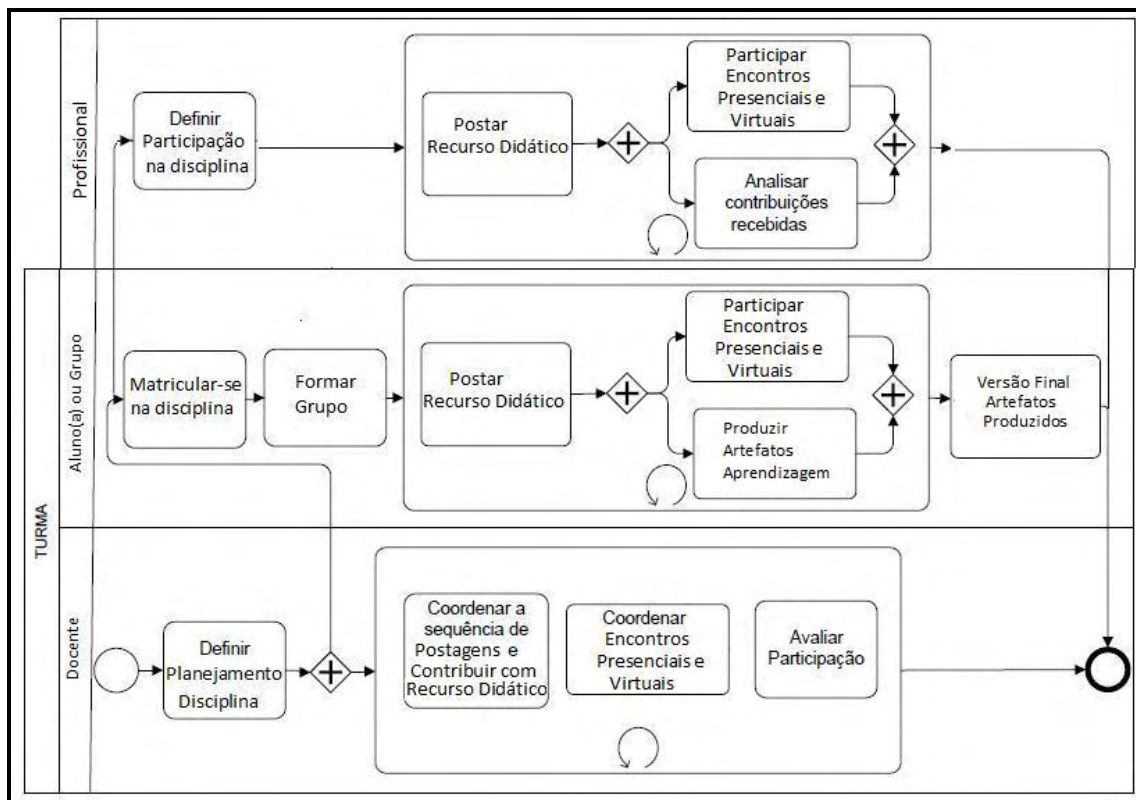


Figura 4.6: Representação do funcionamento da Metodologia ESCollab - Notação BPMN

Na Figura 4.6, podem ser observados os papéis dos agentes envolvidos no processo ensino-aprendizagem sob a ótica da metodologia ESCollab, sendo visualizadas as contribuições esperadas da parte de cada perfil envolvido no processo.

A ESCollab pode ser aplicada a outras áreas do conhecimento onde existam disciplinas com pré-requisitos na grade curricular. Do ponto de vista conceitual espera-se uma maior ênfase dos agentes no planejamento e execução de atividades cooperativas para maximizar o rendimento e minimizar o trabalho repetitivo. O protótipo ENGESOFT (que automatiza o processo usando a ESCollab) utilizado como *plugin* do *moodle* é uma alternativa que está sendo testada e pode ser uma inovação interessante para quem utiliza este último ambiente de educação a distância para cursos que tem disciplinas interligadas em seu programa.

Assim como o Unix quando surgiu com sua mensagem “small is beautiful” (uma alusão a simplicidade) a ESCollab busca lembrar que se os agentes envolvidos simplesmente combinarem os conceitos/conteúdos podem ter excelentes resultados. Simples assim: “preciso que meus alunos tenham determinada aptidão” e que a comunicação entre os docentes envolvidos no processo seja feita de forma efetiva.

4.3 APLICAÇÃO DA METODOLOGIA ESCOLLAB: UM ESTUDO DE CASO

A aplicação da metodologia obedeceu às três fases propostas: planejamento, execução e conclusão. A seguir, será feita a descrição do processo de aplicação da metodologia ESCollab, as observações realizadas e os resultados obtidos.

A metodologia ESCollab foi utilizada parcialmente pela primeira vez no semestre 2010.1 na turma do 4º período do curso de Ciência da Computação da IESB, na disciplina Análise e Projeto de Sistemas I (60h).

No semestre 2010.2, a mesma turma cursou a disciplina Análise e Projeto de Sistema II (60h) também sob a ótica parcial da metodologia ESCollab, tendo sido registrado um ganho de cerca de 15% da carga horária em termos da não necessidade de revisão dos conteúdos da disciplina anterior, diminuição do índice histórico de evasão e aumento do interesse da turma pelas aulas.

No semestre 2011.1, foi introduzida outra IES no experimento, a Faculdade IESA, que possui o mesmo conceito no ENADE (4) que a IESB. A turma pioneira da IESB, então no 6º período, cursou a disciplina Especificação de *Software* I (60 h), sob as recomendações da metodologia ESCollab. Na IESA, propositadamente, de forma a permitir fazer uma análise comparativa, não foi aplicada a metodologia ESCollab na disciplina Engenharia de *Software* (60 h) no semestre 2011.1. Para melhor aplicação do estudo de caso, acordou-se, antes do início das aulas, uma interseção de conteúdos programáticos superior a 90% entre as ementas das disciplinas.

Na IESB, no semestre 2011.1, foi elaborado o planos de aulas, avaliações diagnósticas inicial e final, durante a fase de planejamento, que culminou com uma reunião final com os membros da equipe que conduziria o experimento de aplicação integral da metodologia ESCollab no semestre.

No primeiro dia de aula, os alunos foram submetidos à avaliação diagnóstica inicial com a intenção de levantar o grau de conhecimento prévio dos mesmos com relação aos conteúdos a serem abordados ao longo da disciplina. Esse diagnóstico foi feito utilizando a ferramenta Google Form, propiciando a coleta de dados a partir da página Web da disciplina e o acesso imediato aos resultados pelo docente. Nesse primeiro momento, também foi feita

uma explicação detalhada do funcionamento da metodologia ESCollab e o “novo” papel a ser desempenhado pelos alunos dentro da mesma.

Os alunos foram divididos em 2 grupos rotativos, em cada atividade, compostos de 4 alunos e trabalharam ao longo do semestre no desenvolvimento do sistema de controle de aluguel por temporada. Os recursos didáticos ou objetos de aprendizagem (ementa, apresentações, textos, listas de exercícios, estudos de casos, problemas etc) foram colocados à disposição da turma na página Web do professor, conforme preconiza a metodologia ESCollab. As ferramentas de interação foram o Google Groups e o Twitter, sendo usada como principal ferramenta de compartilhamento o Google Docs.

Durante o semestre, foram registradas, em média, 20 interações semanais entre os alunos e entre o professor e os alunos. A cada 15 dias foram aplicados mini-testes para aferir a aprendizagem dos conteúdos teóricos lecionados em sala. No final do semestre, os alunos apresentaram seminários sobre temas diversos ligados à disciplina. O curso foi acompanhado por um agente externo ao curso de graduação, que foi um aluno do mestrado em informática, participando das atividades didáticas como um consultor externo.

Na última semana de aula, foi aplicada a avaliação diagnóstica final, utilizando o Google Forms. Os resultados da avaliação inicial e final foram confrontados, permitindo a verificação de ganhos de aprendizagem incremental média nos alunos da turma da ordem de 85% em relação ao estágio inicial de conhecimentos do conteúdo programático da disciplina aferido pela avaliação diagnóstica inicial.

Na Faculdade IESA, utilizou-se a abordagem tradicional. O docente fez o planejamento isoladamente, os alunos não desenvolveram projetos, a principal ferramenta didática foi a aula expositiva, foram avaliados apenas em duas provas bimestrais durante o curso e os objetos de aprendizagem foram sendo disponibilizados ao longo do semestre. A turma era composta por 16 alunos e não foram compostos grupos de trabalho. A comunicação foi feita através de uma lista de discussão da turma, sendo registradas, em média, 4 interações semanais entre os alunos e entre alunos e docente. Na aplicação da avaliação diagnóstica final, verificou-se um ganho incremental de aprendizagem médio da ordem de 64% em termos de aprendizagem ao longo da disciplina.

Na comparação realizada entre os ganhos de aprendizagem aferidos pelas avaliações diagnósticas inicial e final em cada uma das turmas, verificou-se uma absorção maior de conteúdos da ordem de 21% a mais para a turma da IESB, que adotou a metodologia ESCollab.

Uma prova objetiva sobre os assuntos comuns às duas disciplinas foi elaborada conjuntamente pelos docentes das duas turmas e aplicada no final do semestre aos alunos de ambas as turmas, mostrando, em média, uma diferença de aprendizagem da ordem de 40% a mais em favor da turma da IESB, que utilizou a metodologia ESCollab.

Quando a comparação entre as turmas foca as interações entre os membros do processo ensino-aprendizagem durante o semestre, registrou-se uma diferença a maior no número de interações da ordem de 400% (quatrocentos por cento) para a turma da IESB.

Com os resultados promissores dos semestres anteriores, no semestre 2011.2, a metodologia ESCollab foi aplicada na íntegra na turma do 7o. período, na disciplina métodos computacionais da IESB e na Faculdade IESA, a aplicação efetiva foi feita de forma integrada em 3 disciplinas (Qualidade de *Software*, Sistemas de Informação Inteligentes e Desenvolvimento de Aplicações Distribuídas), no 8º. período do curso de ciência de computação, percebendo-se uma boa receptividade por parte da comunidade da IES, embora se tenha registrado alguma resistência dos alunos da IESA em relação à obrigatoriedade de realização de um projeto integrado entre as três disciplinas citadas.

Essa resistência deve-se ao fato de que nos semestres anteriores não havia a obrigatoriedade da implementação de projetos, conforme preconiza a metodologia ESCollab.

A título de permitir uma análise da aprendizagem dos alunos envolvidos no experimento, apresenta-se, na Figura 4.7, o resultado final da disciplina engenharia de *software* cursada pela turma da IESA no período 2011.1:

CURSO: CIENCIA DA COMPUTACAO		TURMA: CC7P01-3601						
Disciplina: 713J ENG SOFT (ENGENHARIA DE SOFTWARE)								
RA - Aluno	TP	B1	B2	PS	NC	MA	MF	E
6027917333	GR		5.00	8.0		6.5	5.8	AP
6028424306	GR	8.2	6.0			7.1	7.1	AP
6028425000	GR	8.2	6.0			7.1	7.1	AP
6024923588	GR	6.4	6.3			6.35	6.2	AP
6024924185	GR	6.9	6.6			7	7	AP
6027920059	GR		5.0	9.0		7	7	AP
6029166912	GR		5.20	2.0		3.6	3.3	RN
6028424519	GR		2.50	6.0		4.25	5.6	AP
6027917066	GR		5.70	4.5		5.1	5.3	AP
6028425124	GR		4.50	4.3		4.4	6.2	AP
6020001312	GR	6.4		3.7		5.05	4	RN
6029253360	GR		3.20	2.5		2.85	3.4	RN
6027917449	GR					0	0	RN
6029253300	GR	5.5	4.70			5.1	5.1	AP
6029166750	GR	4.6	5.0			4.8	5	AP

Figura 4.7: Resultado Final da Disciplina Engenharia de *Software* – IESA – 2011.1
 Fonte: Pesquisa Direta, 2011

Através da listagem constante na Figura 4.7, pode-se observar que 4 alunos foram reprovados e que a nota final média na disciplina engenharia de *software* foi 5,21.

As Figuras 4.8, 4.9 e 4.10, apresentam o resultado final das disciplinas subseqüentes na grade curricular do curso de ciência da computação na IESA (Qualidade de *Software*, Sistemas de Informação Inteligentes e Desenvolvimento de Aplicações Distribuídas), tendo sido ministradas sob a ótica da Metodologia ESCollab.

CURSO: CIENCIA DA COMPUTACAO TURMA: CC8P01-3601
Disciplina: 982T QUAL SOFT QUALIDADE DE SOFTWARE

RA - Aluno	TP	B1	B2	PS	NC	MA	MF	E
6027917333	GR		6.6	6.1		6.35	7.2	AP
6028424306	GR	9.2	8.0			8.6	8.6	AP
6028425000	GR	9.0	8.0			8.5	8.5	AP
6024923588	GR		7.1	8.6		7.85	7.9	AP
6024924185	GR	8.0	7.1			7.55	7.6	AP
6027920059	GR	8.4	8.5			8.45	8.5	AP
6029166912	GR	8.3	9.3			8.8	8.8	AP
6028424519	GR		8.6	6.8		7.7	7.7	AP
6027917066	GR	9.0	8.7			8.85	8.9	AP
6028425124	GR	5.4	9.0			7.2	7.2	AP
6020001312	GR	7.1	7.2			7.15	7.2	AP
6029253360	GR	7.0	5.3			6.15	7.1	AP
6027917449	GR	9.4	4.0			7	7	AP
6020001696	GR		4.6	4.5		4.55	5.3	AP
6029253300	GR	6.4	6.0			6.2	6.4	AP
6029166750	GR	8.5	9.1			8.8	8.8	AP

Figura 4.8: Resultado Final da Disciplina Qualidade de *Software* – IESA – 2011.2
 Fonte: Pesquisa Direta, 2011

Disciplina: 992T SII SIST INFORM INTELIGENTES

RA - Aluno	TP	B1	B2	PS	NC	MA	MF	E
6027917333	GR		7.6	9		8.3	8.3	AP
6028424306	GR	10	9.8			9.9	9.9	AP
6028425000	GR	8.6	9.3			8.95	9	AP
6024923588	GR	7.5	6.5			7	7	AP
6024924185	GR	9.2	10			9.6	9.6	AP
6027920059	GR		8.8	7.8		8.3	8.3	AP
6029166912	GR	7.7	6.6			7.15	7.2	AP
6028424519	GR	7.1	6.9			7	7	AP
6027917066	GR		7	7.1		7.05	7.1	AP
6028425124	GR	8.8	8.5			8.65	8.7	AP
6020001312	GR		8.1	4.8		6.45	6.2	AP
6029253360	GR	8.3	6.8			7.55	7.6	AP
6027917449	GR	4.4	6.3			5.35	5.7	AP
6020001696	GR		7.3	1.8		4.55	6	AP
6029253300	GR	9	9			9	9	AP
6029166750	GR	9	7.8			8.4	8.4	AP

Figura 4.9: Resultado Final da Disciplina Sist Inform Inteligentes – IESA – 2011.2
 Fonte: Pesquisa Direta, 2011

Disciplina: 792J DESENVOLV SIST DISTRIBUIDOS								
RA - Aluno	TP	B1	B2	PS	NC	MA	MF	E
6027917333	GR	9	7.25			8.12	8.1	AP
6028424306	GR	9	8.5			8.75	8.8	AP
6028425000	GR	8.4	7.25			7.62	7.8	AP
6024923588	GR	7.5	8			7.75	7.8	AP
6024924185	GR	8.5	7.75			8.12	8.1	AP
6027920059	GR	8	7.25			7.62	7.6	AP
6029166912	GR	8	8.25			8.12	8.1	AP
6028424519	GR		7.25	10		8.62	8.6	AP
6027917066	GR	9	8			8.5	8.5	AP
6028425124	GR	9	8.5			8.75	8.8	AP
6020001312	GR		7	8		7.5	7.5	AP
6029253360	GR	8.9	7			7.95	8	AP
6027917449	GR		7	7.5		7.25	7.3	AP
6020001696	GR							
6029253300	GR	8.8	7			7.9	7.9	AP
6029166750	GR	8.5	8.5			8.5	8.5	AP

Figura 4.10: Resultado Final da Disciplina Desenv. Sist. Distribuídos – IESA – 2011.2
 Fonte: Pesquisa Direta, 2011

A partir da análise dos relatórios apresentados, conclui-se que houve ganhos no desempenho individual de cada aluno e que a nota final média da turma em cada uma das disciplinas (7,85) também foi superior em 51% à nota média final na disciplina engenharia de *software* (5,21), ministrada no semestre anterior, utilizando a abordagem de ensino convencional, visualizando-se também que nenhum aluno foi reprovado com a adoção da metodologia ESCollab.

CONCLUSÕES E TRABALHOS FUTUROS

A proposta desta dissertação, a ESCollab, é uma metodologia colaborativa de ensino que visa construir conhecimento e práticas de ensino voltadas para a área de Engenharia de *Software* com o propósito central de contribuir com a melhoria da formação dos profissionais envolvidos no desenvolvimento de *software*. Obviamente, a delimitação deste trabalho restringe-se ao âmbito específico do ensino das disciplinas da área de Engenharia de *Software*.

Com a adoção da metodologia ESCollab, vislumbra-se a possibilidade real de ampliação e democratização do conhecimento na área de Engenharia de *Software* através das contribuições de alunos, professores, gestores de cursos e profissionais, materializando a concepção de uma inteligência coletiva sobre o domínio de conhecimento de Engenharia de *Software*.

A título de registro das dificuldades encontradas no processo, deve-se considerar a cultura institucional pré-existente nas duas IES nas quais foram realizados os experimentos como um certo fator de dificuldade na realização dos mesmos, havendo alguma resistência à mudança e um certo descompromisso da parte de alguns com a discussão de temas como planejamento e didática. Uma justificativa plausível é a de que a maioria dos cursos de graduação e de pós-graduação na área de informática, de uma forma geral, dão pouca importância à prática docente e aos aspectos de metodologia de ensino em suas grades curriculares, seguindo à risca as Diretrizes Curriculares do MEC, sob esse aspecto.

Com base na aplicação da metodologia ESCollab em uma ambiente real de ensino em disciplinas da área de engenharia de *software*, avalia-se que, de uma forma geral, a hipótese enunciada inicialmente nesta dissertação pode ser comprovada, sendo alcançados também os objetivos propostos. Os resultados da avaliação da aplicação sinalizam no sentido de ganhos reais significativos na aprendizagem dos alunos envolvidos no estudo de caso em comparação aos que não utilizaram a metodologia ESCollab.

O direcionamento futuro desta pesquisa aponta no sentido do aprofundamento de como o uso das redes sociais, do trabalho colaborativo, da responsabilidade social e de um maior rigor metodológico nos aspectos didáticos podem contribuir para a melhoria do ensino na área da Ciência da Computação. Como contribuições futuras na Educação em

Computação, podendo advir desta dissertação, sugere-se algumas propostas de pesquisas nesta área:

1. Aprimorar o protótipo ENGSOFT e desenvolver de um ambiente computacional para automatizar as interações intrínsecas ao processo educacional em engenharia de *software* sob a ótica da metodologia ESCollab (PAIVA, 2011b);
2. Aprofundar a discussão sobre abordagens didáticas mais rigorosas do ponto de vista metodológico no ensino na área da ciência da computação;
3. Adaptar a metodologia ESCollab para todas as áreas da ciência da computação;
4. Adaptar a metodologia ESCollab para demais áreas do conhecimento;
5. Desenvolver ferramental para integrar a comunidade científica na área de engenharia de *software* e ciência da computação.

A elaboração desse trabalho foi uma experiência bastante enriquecedora, sendo possível verificar, na qualidade de educador, o desenvolvimento de várias competências que são motivadoras para a busca contínua do crescimento profissional, crendo firmemente que a metodologia Escollab pode contribuir com a melhoria da qualidade do ensino na área da ciência da computação, estimulando os agentes envolvidos no processo de ensino-aprendizagem a colaborarem e a adotarem maior rigor metodológico em suas atividades didáticas.

REFERÊNCIAS

ASSOCIATION FOR COMPUTING MACHINERY - ACM/IEEE.. **Computing curricula 2001 computer science**. Disponível em: <<http://www.acm.org>>. Acesso em: 29 abr. 2011.

ALMEIDA, L. D. A.; BARANAUSKAS, M. C. C. Um prospecto de sistemas colaborativos: Modelos e Frameworks. In: VIII SIMPÓSIO SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 21-24 Outubro 2008, 204-213.

ANDERSON, T. **Modes of interaction in distance education**: recent developments and research questions. In: MOORE, Michael Grahame; ANDERSON, William G. (Ed.). *Handbook of distance education*. Mahwah, NJ: Lawrence Erlbaum, 2003. p. 129-144.

ANISSETY, P.; YOUNG, P. **Collaboration problems in conduction a group project in a software engineering course**. *Journal of Computing Sciences in Colleges*, v. 26 Issue 5, Mai. 2011.

ARCOVERDE, D. F.; MELO, C. A.; FRANCO, R. O. S. **Labbos**: uma rede social para pesquisa colaborativa. IHC 2008. Demonstrações e Pôsteres. 21-24 Outubro, Porto Alegre – RS, Brasil. pp 326-327.

BACON, F. **Novum organum**. 1620. Disponível em: <http://p.download.uol.com.br/cultvox/livros_gratis/novum_organum.pdf>. Acesso em: 31 mar. 2011.

BARBOSA, R. M. **Ambientes virtuais de aprendizagem**. Porto Alegre: Artmed, 2005

BITTENCOURT, J.de V, GOMES, J Z, JUNQUEIRA, L M, et. al. **Criando uma plataforma para projetos de aprendizagem**: desafios e reflexões no desenvolvimento do AMADIS. *RENOTE. Revista Novas Tecnologias na Educação*, v. 4, pp. 1-13, 2006.

BORDENAVE, Juan D.; PEREIRA, Adair M. **Estratégias de ensino-aprendizagem**. 28. ed. Petrópolis:Vozes, 2007

BRASIL. Ministério da Educação e Cultura - MEC. **Censo da Educação Superior 2009**. Brasília, 2010. Disponível em: <<http://www.portalmec.gov.br>>. Acesso em: 30 de mai. 2011.

BRASIL. Ministério da Educação e Cultura - MEC. **Diretrizes curriculares de cursos da área de computação e informática**. 2005. Disponível em: <<http://www.portalmec.gov.br>>. Acesso em: 30 mar.2011.

BRASIL. Ministério da Ciência e Tecnologia - MCT . Secretaria de Política de Informática. **Pesquisa de qualidade no setor de software Brasileiro, 2009**. Brasília, 2010. Disponível em: <<http://www.mct.gov.br>>. Acesso em: 01 fev. 2011

BRUNER, J. **The culture of education**. Cambridge, MA: Harvard University Press, 1996.

CHEN, J.; LU, H.; AN, L. & ZHOU, Y. **Exploring teaching methods in software engineering education.** *Proc. 4th Int. Conf. Computer Science & Education ICCSE '09*, 2009, 1733-1738

COCCOLI, M.; STANGANELLI, L. & MARESCA, P. **Computer supported collaborative learning in software engineering.** *Proc. IEEE Global Engineering Education Conf. (EDUCON)*, 2011, 990-995

COSTA, R. *et al.* **A.M.I.G.O.S:** uma plataforma para gestão de conhecimento através de redes sociais. SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, 2008. IEEE Xplore, p. 192-203.

COSTA-SORIA, C.; LLAVADOR, M. & DEL CARMEN PENADES, M. **An approach for teaching software engineering through reverse engineering.** *Proc. EAEEIE Annual Conf*, 2009, 1-6

DANIELS, H. **Vygotsky e a pedagogia.** São Paulo:Edições Loyola, 2003.

DEKHANE, S. & TSOI, M. Y. **Work in progress:** inter-disciplinary collaboration for a meaningful experience in a *software* development course. *Proc. IEEE Frontiers in Education Conf. (FIE)*, 2010.

DILLENBOURG, P., BAKER, M., BLAYE, A. & O'MALLEY. ,1996. **Evolution of research on collaborative learning.** In *E. Spada & P. Reiman, (Eds.) Learning in Humans and Machine: Towards an interdisciplinary learning science.* pp. 189-211, Oxford, Elsevier.

ENDLER, A. B.; FUCKS, M.; FILIPPO, H.; RAPOSO, D. **Ambientes colaborativos de realidade virtual e aumentada.** In: Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações. Livro do Pré-Simpósio IX SVR, pp 168191, 2007. x, 27, 28, 29

FAGUNDES, L. C. ; SATO, L. S. ; MAÇADA, D. L., 1999 . **Aprendizes do futuro:** as inovações começaram. 1. ed. Brasília: PROINFO/SEED/MEC, 1999. v. 19. 95 pág.

FILIPPO, D. *et al.* **Ambientes colaborativos de realidade virtual e aumentada.** In: Realidade Virtual e Aumentada - Conceitos, Projeto e Aplicações, Cláudio Kirner e Robson Siscoutto (eds), Editora SBC–Sociedade Brasileira de Computação, Porto Alegre, 2007, ISBN 85-7669-108-6, Cap. 9, pp 168-191.

FREIRE, P. **Educação e mudança.** Rio de Janeiro: Paz e Terra, 1979.

_____. **Pedagogia da autonomia.** 31. ed. São Paulo:Paz e Terra, 2005.

FUJIOKA, R. C. **SCollab core:** um framework para desenvolvimento e integração de aplicações JEE a redes sociais. 2011. Dissertação (Mestrado em Informática) – UFPB/CCEN. João Pessoa, UFPB

FUKS, H.; RAPOSO, A. B.; GEROSA, M. A. **Engenharia de groupware:** desenvolvimento de aplicações colaborativas. XXI Jornada de Atualização em Informática, Anais do XXII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, v2, Cap.3, 2002, ISBN 85-88442-24-8, pp.89-128.

GAO, Y.; FENG, X. **Incorporating innovation-oriented education into the software engineering course.** *Proc. IEEE Int. Conf. Systems, Man and Cybernetics SMC 2009*, 2009.

GARG, K.; VARMA, V. **Case studies as assessment tools in software engineering classrooms.** *Proc. 22nd Conf. Software Engineering Education and Training CSEET '09*, 2009, 8-11

GAROUSI, V. **Applying peer reviews in software engineering education: an experiment and lessons learned.** *#IEEE_J_EDU#*, 2010, 53,182-193.

GASPARIN, J. L. **Uma didática para a pedagogia histórica-crítica.** 3. ed. Campinas: Autores Associados, 2005.

GERDAMIN, Domenico. **Collaborative tagging as a community-driven approach to knowledge sharing.** 2010. 161 f. Tese (Doutorado) - Universidade de Bari, Bari, 2010.

GIRALDO, F. D.; COLLAZOS, C. A.; OCHOA, S. F.; ZAPATA, S. & DE CLUNIE, G. T. **Teaching software engineering from a collaborative perspective: some Latin-American experiences.** *Proc. Workshop Database and Expert Systems Applications (DEXA)*, 2010, 97-101.

GOTEL, O.; KULKARNI, V.; SAY, M.; SCHARFF, C. & SUNETNANTA, T. **A Global and competition-based model for fostering technical and soft skills in software engineering education.** *Proc. 22nd Conf. Software Engineering Education and Training CSEET '09*, 2009, 271-278.

GRINSPUN, M. P. S. Z. (org.). **Educação tecnológica: desafios e perspectivas.** 2. ed. São Paulo: Cortez, 2001.

HUTCHINS, E. **Cognition in the Wild.** Cambridge, MA: MIT Press, 1995.

IVAN, I.; CIUREA, C. **Quality characteristics of collaborative systems.** 2009 Second International Conferences on Advances in Computer-Human Interactions. IEEE Computer Society, 2009.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Pesquisa de Serviços de Tecnologia da Informação 2009.** Disponível em: <<http://www.ibge.gov.br/home/estatistica/economia/psit/2009/default.shtm>>. Acesso em: 20 mar. 2011.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS - IEEE. **Guide to the software engineering body of knowledge : 2004 version.** Los Alamitos: IEEE Publisher, 2004.

JIA, Y. **Improving software engineering courses with case study approach.** *Proc. 5th Int Computer Science and Education (ICCSE) Conf*, 2010, 1633-1636

JIANGUO, L. **Combination of Research and Teaching in Software Engineering Education.** *Proc. WASE Int. Conf. Information Engineering ICIE '09*, 2009, 2, 437-440.

LANUBILE, F.; EBERT, C.; PRIKILADNICKI, R.; VÍZCAINO, A.. **Collaboration tools for global software engineering**. *IEEE Software*. 2010.

LI, Z.; LI, J.-M.; GUAN, Y.-C. & YI, K. **Practice of actual project driven teaching model on software engineering major**. *Proc. Int. Conf. Information Engineering and Computer Science ICIECS 2009*,

LIBÂNEO, J. C. **Didática**. São Paulo: Cortez, 1994. Coleção magistério.

LIBÂNEO, J. C.. **Adeus professor, adeus professora?: novas exigências educacionais e profissão docente**. 6. ed. São Paulo:Cortez Editora, 2002.

LU, X.; XUE, H.; OUYANG, H. & TU, M. **Practice of using suite of teaching to build the students' forming capacity of software engineering**. *Proc. Second Int Communication Systems, Networks and Applications (ICCSNA) Conf, 2010, 1*, 308-311.

MANGUEIRA, Jorge A. et al. **JCollab: uma ferramenta para produção e distribuição de telejornais no contexto da Web 2.0**. Disponível em: <http://jcollab.lavid.ufpb.br/?page_id=15>. Acesso em: 30 mai.2011.

MEDEIROS, Álvaro Francisco de Castro. **Procedimentos Automáticos para Gerência de Comunidades no Orkut**. Relatório Técnico, Workshop – Labes, out. 2009.

MIRIAN-HOSSEINABADI, S.-H.; AGHAKASIRI, Z.; SADEGHI, A.;DELFIANI, P. & GHANDEHARI, M. **Emphasizing experiences in teaching software engineering courses**. *Proc. 2nd Int Education Technology and Computer (ICETC) Conf, 2010, 2*.

MOORE, M. **Three types of interaction**. *American Journal of Distance Education*. 3 (2), p. 1-6, 1989.

MOREIRA, M. A. **Teorias de aprendizagem**. São Paulo: EPU, 1999.

MOTTOK, J. & GARDEIA, A. **The Regensburg Concept of P-Seminars --- How to organize the interface between secondary school and university education to create a didactic cooperation between teaching and learning of Software Engineering with Lego Mindstorms NXT Embedded Robot Systems**. *Proc. IEEE Global Engineering Education Conf. (EDUCON)*, 2011, 917-920

NÉBIAS, C. M. **Metodologia de Ensino**. (material de curso), s/d.

OCDE. **Knowledge management for the learning society**. Paris: OCDE, 2001

OCDE. **Teachers Matter: Attracting , Developing and Retaining Effective Teachers**. Paris: OCDE, 2005

OBJECT MANAGEMENT GROUP - OMG. **Business Process Model and Notation-BPMN**, 2009. Disponível em: <<http://www.omg.org/spec/BPMN/1.2.>>. Acesso em: 05 mai. 2011.

O'REILLY, T. **What Is Web 2.0**. Setembro de 2005. Disponível em: <<http://oreilly.com/web2/archive/what-is-web-20.html>>. Acesso em: 28 jul. 2010

PAIVA, S. R. **Introdução à programação: do algoritmo às linguagens atuais**. Rio de Janeiro: Ciência Moderna, 2008

_____. **Uma revisão sistemática das pesquisas realizadas sobre a melhoria no ensino de engenharia de *software***. João Pessoa, UFPB. Relatório Técnico – UFPB/CCEN, 2011a.

_____. **Pesquisa nacional sobre ensino de engenharia de *software***. João Pessoa, UFPB. Relatório Técnico – UFPB/CCEN, 2011b.

_____. **SOFTENG: um ambiente para dar suporte a uma metodologia voltada para o ensino de Engenharia de *Software***. João Pessoa, UFPB. Relatório Técnico – UFPB/CCEN, 2011c

PARNAS, David Lorge. ***Software engineering programs are not computer science programs***. *Ann. Software Eng.*, 1998: 19-37.

PAVIANI, J. **Problemas de filosofia da educação**. 7. ed. Caxias do Sul: Educs, 2005.

PEREIRA, M. Z. C.; CARVALHO, M. E. P. de; PORTO, R. C. C. **Globalização, interculturalidade e currículo na cena escolar**. São Paulo: Editora Alínea, 2009.

PERRENOUD, P. **10 Novas competências para ensinar**. Porto Alegre: ArtMed Editora, 2000.

PIAGET, J. **Psicologia e pedagogia**. Rio de Janeiro: Forense Universitária, 1982.

PIMENTEL, M., **RUP-3C-Groupware: um processo de desenvolvimento de groupware baseado no Modelo 3C de Colaboração**. Tese (Doutorado) - Departamento de Informática, PUC-Rio, Rio de Janeiro, RJ, Brasil, 2006.

PISANI, P; PIOTET, D. **Como a Web transforma o mundo: a alquimia das multidões**. 1. ed, Editora Senac, 2010.

PRESSMAN, R. S. **Engenharia de *software***. 6. ed. Porto Alegre: AMGH, 2010.

PRIKLADNICKI, R. et al. **Ensino de engenharia de *software*: desafios, estratégias de ensino e lições aprendidas**. In: XII SIMPÓSIO BRASILEIRO DE ENGENHARIA DE *SOFTWARE*. Fortaleza, 2009.

QIU, M. & CHEN, L. **A Problem-Based Learning approach to teaching an advanced *software engineering* course**. *Proc. Second Int Education Technology and Computer Science (ETCS) Workshop*, 2010, 3, 252-255.

RICHARDSON, I. & DELANEY, Y. **Problem-Based Learning in the *software engineering* classroom**. *Proc. 22nd Conf. Software Engineering Education and Training CSEET '09*, 2009, 174-181.

RICO, D. F. & SAYANI, H. H. **Use of Agile Methods in Software Engineering Education.** *Proc. AGILE '09. Agile Conf*, 2009, 174-179.

RIQUE, T. P. **WebQDA:** uma ferramenta Web Colaborativa para apoiar a análise qualitativa de dados. 2011. Dissertação (Mestrado em Informática) – UFPB/CCEN, João Pessoa, UFPB

ROGERS, Y. **A brief introduction to distributed cognition.** UK: University of Sussex, Aug 1997. School of Cognitive and Computing Sciences at Brighton-University of Sussex. Disponível em: <<http://www.cogs.susx.ac.uk/users/yvonner/dcog>>. Acesso em: 10 mar. 2011.

SANTORO, F.M., BORGES, M.R.S., SANTOS,N, 2003. **Learning through collaborative projects: the architecture of an environment.** In: *International Journal of Computer Applications in Technology*, v 16 , Issue 2-3 (July 2003), ISSN:0952-8091., pp. 127-141

SANTORO, F. M. ; PIMENTEL, M., 2009. **Tecnologias computacionais para educação.** In: *Chronos (UNIRIO)*, v. 1, pp. 83-91, 2009.

SANTOS, M. S.; SANTORO, N.; BORGES, F. M. **Um framework para estudo de ambientes de suporte à aprendizagem cooperativa.** Revista Brasileira de Informática na Educação, UFSC, Florianópolis, (4), 1999. x, 36

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO - SBC. **Currículo de referência para os cursos de graduação em bacharelado em Ciência da Computação e Engenharia da computação.** 2005. Disponível em: <<http://www.sbc.org.br>>. Acesso em: 05 mai.2011.

SILVA, A. L. C. **Ambientes virtuais de aprendizagem:** uma experiência no ensino presencial de graduação. 2003. Dissertação (Mestrado) Universidade do Vale do Itajaí, São Paulo: UVI pg. 40, 41, 42

SILVA, K.X.S. **Webquest:** uma metodologia para pesquisa escolar por meio da internet. UCB, 2006. x, 33 100.

SOMMERVILLE, Ian. **Engenharia de software.** 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

SONG, H.; Li, X.; WANG, P. & ZHAO, D. **Improving personalized talent education by strengthening specialty direction of information system in software engineering.** *Proc. Second Int. Conf. Education Technology and Training ETT '09*, 2009, 16-18

SOUZA, Mariane et al. **SPARSE:** uma abordagem para o ensino de engenharia de *software* baseada em jogos e simulação. Anais do XXI SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. João Pessoa, 2010.

STAHL, G., KOSCHMANN, T., e SUTHERS, D., 2006. **Computer-supported collaborative learning: An historical perspective.** In: *R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences*, Cambridge University Press, Cambridge, UK, pp 409-426.

TIEJUN, P.; YUEFENG, F.; LEINA, Z.; XIAOYAN, Y. & YUNPENG, L. **Research of high-quality innovative and pioneering undergraduate training model of Software Engineering.** *Proc. 2nd Int Education Technology and Computer (ICETC) Conf.*, 2010, 3

TORQUATO, J. R. C. **Ubiproject**: uma infra-estrutura para redes sociais de projetos compatível com o OAI-PMH. João Pessoa: UFPB, 2009. Dissertação (Mestrado) – UFPB/CCEN.

VIGOTSKY, L. **A formação social da mente**: o desenvolvimento dos processos psicológicos superiores. São Paulo: Martins Fontes, 1988.

YADAV, S. S. & XIAHOU, J. **Integrated project based learning in software engineering education**. *Proc. Int Educational and Network Technology (ICENT) Conf.*, 2010, 34-36.

YANG, C. & LIU, Y. **Teaching reform and practice on the software engineering course**. *Proc. 1st Int Information Science and Engineering (ICISE) Conf.*, 2009, 3470-3473.

ZHANG, J. & LI, J. **Teaching software engineering using case study**. *Proc. Int Biomedical Engineering and Computer Science (ICBECS) Conf.*, 2010.

APÊNDICE A

Revisão Sistemática das pesquisas realizadas sobre a melhoria do ensino de Engenharia de *Software*



RELATÓRIO TÉCNICO

UMA REVISÃO SISTEMÁTICA DAS PESQUISAS REALIZADAS
SOBRE A MELHORIA DO ENSINO DE ENGENHARIA DE
SOFTWARE

**MESTRANDO:
SEVERINO DO RAMO DE PAIVA**

**ORIENTADOR:
ÁLVARO MEDEIROS**

JOÃO PESSOA
JUNHO DE 2011

UMA REVISÃO SISTEMÁTICA SOBRE AS PESQUISAS VOLTADAS PARA A
MELHORIA DO ENSINO DE ENGENHARIA DE SOFTWARE

RESUMO

O ensino de engenharia de software vem assumindo uma importância cada vez maior dentro da sociedade moderna devido à onipresença dos sistemas de informação em todas as áreas de atividades dos seres humanos. Tanto assim, que na lista das dez maiores empresas do mundo, a maioria é da área de TI - Tecnologia da Informação. Levando-se em consideração a importância da boa formação dos futuros engenheiros de software, registram-se várias iniciativas visando aperfeiçoar o ensino de engenharia de software no mundo todo. Esse texto descreve o processo de realização de uma revisão sistemática da literatura sobre as iniciativas realizadas para aperfeiçoar o ensino de engenharia de software, apresentando os resultados de forma sucinta e organizada.

Palavras-Chaves: Engenharia de software, Sistemas de informação, Tecnologia da Informação.

UMA REVISÃO SISTEMÁTICA SOBRE AS PESQUISAS VOLTADAS PARA A MELHORIA DO ENSINO DE ENGENHARIA DE SOFTWARE

ABSTRACT

The teaching of software engineering is assuming a growing importance in modern society due to the ubiquity of information systems in all areas of human activities. So much so that in the ten largest companies in the world, most are in IT- Information Technology. Taking into account the importance of proper training of future software engineers, register several initiatives to improve the teaching of software engineering worldwide. This text describes the process of conducting a systematic review of literature on their initiatives to improve teaching software engineering, presenting the results in a succinct and organized.

1. INTRODUÇÃO

A Engenharia de Software é uma das subáreas da Ciência da Computação sobre a qual se tem gerado muitas discussões em torno da qualidade e da eficiência das atuais estratégias de ensino-aprendizagem voltadas para a formação profissional dos futuros desenvolvedores de software. Em geral, é dada muita atenção aos conteúdos teóricos e pouca importância às atividades práticas, sendo atribuídas, na montagem das grades curriculares dos cursos de graduação, muitas vezes, cargas horárias insuficientes para o cumprimento de todo o conteúdo programático da área. Os registros de baixo rendimento e alta desmotivação nas salas de aula são comumente citados em trabalhos acadêmicos sobre o assunto [3], [4], [5], [6], [8], [9]. Adicionalmente, em pesquisas do MCT/SEPIN sobre a qualidade de software, a indústria nacional tem apresentado resultados que deixam a desejar em comparação com outros países, necessitando urgentemente aprimorar a qualidade dos profissionais que chegam ao mercado de trabalho [1].

A pesquisa bibliográfica constitui-se em um procedimento formal para a aquisição de conhecimento sobre a realidade [2]. Nesse sentido, pretende-se investigar, na literatura técnica da área, o material relevante sobre iniciativas que visem melhorar o processo de ensino-aprendizagem na área de Engenharia de Software. Para qualificar os resultados obtidos, gerando mais valor científico, optou-se por realizar uma revisão sistemática na qual, diferentemente da revisão informal da literatura, o pesquisador segue um roteiro de revisão previamente estabelecido.

O presente relatório apresenta essa revisão, e faz a discussão dos resultados obtidos. O trabalho está subdividido nas seguintes seções: a seção 2 descreve o objetivo do trabalho; a seção 3 apresenta o planejamento da revisão sistemática e o roteiro utilizado na mesma; a seção 4 descreve o processo utilizado na revisão sistemática e apresenta os resultados obtidos; a seção 5 apresenta os resultados da revisão sistemática, categorizando-os pela natureza da iniciativa; a seção 6 realiza a discussão dos resultados e faz as considerações finais.

2. OBJETIVO DO TRABALHO

Em função da importância da área de Engenharia de Software na formação dos futuros profissionais da área de desenvolvimento de software, o LABES - Laboratório de Engenharia de Software – deu início, no ano de 2010, a uma linha de pesquisa que visa contribuir com uma nova abordagem metodológica no ensino das disciplinas dessa área, visando tornar o ensino das mesmas mais atrativo e a aprendizagem dos educandos mais efetiva.

3. PROCESSO UTILIZADO NA REALIZAÇÃO DA REVISÃO SISTEMÁTICA DA LITERATURA

Esta etapa tem o objetivo sistematizar o processo utilizado para identificar as iniciativas, práticas ou experiências voltadas para contribuir com a melhoria do ensino de Engenharia de Software.

3.1. Fontes

A pesquisa foi realizada nas seguintes bases de dados:

- a) IEEE (<http://ieexplore.ieee.org>);
- b) ACM (<http://portal.acm.org/dl.cfm>);
- c) Science Direct (<http://www.sciencedirect.com>)

3.2. Termos utilizados na busca

- a) software engineering;
- b) education
- c) training
- d) teaching
- e) mentoring
- f) course
- g) tool
- h) method
- i) curriculum
- j) collaborative
- k) Web

3.3. “Strings” usados nas buscas

("software engineering") AND (education OR training OR teaching OR mentoring OR course) AND (tool OR method OR curriculum) AND (collaborative OR web)

3.4. Critérios para seleção dos trabalhos:

- os trabalhos devem estar disponíveis na Web e sem custo;
- os trabalhos devem ter menos de 3 (anos) anos de publicação, isto é, os textos devem ter sido publicados nos anos de 2009, 2010 e 2011;
- os trabalhos devem tratar de iniciativas voltadas para o ensino de Engenharia de Software;
- os trabalhos devem ter um bom nível em termos de teor científico.

4. ANÁLISE E SELEÇÃO DOS DOCUMENTOS RECUPERADOS NA BUSCA

No Quadro 2, pode-se observar a quantidade de documentos recuperados em cada fonte de pesquisa a partir das *strings* de busca utilizadas:

BASE DE DADOS	ARTIGOS ENCONTRADOS
IEEE	396
ACM	257
Science Direct	172

Quadro 2 - Quantidade de documentos recuperados por base de pesquisa
Fonte: Elaboração própria

Feita a análise dos documentos encontrados utilizando os critérios de aceitação elencados no subitem 3.4, obteve-se a relação dos documentos selecionados, conforme o **Erro! Fonte de referência não encontrada.:**

#	REFERÊNCIA	PAÍS DE ORIGEM	ABORDAGEM UTILIZADA	AVALIAÇÃO
01	(PRIKLADNICKI, Rafael et al., 2009)	Brasil	Metodologia baseada em dinâmicas de grupo, EAD, experimentos, projetos, atividades	Utiliza algumas ferramentas didáticas interessantes, mas ignora outras possibilidades como colaboração, produção de artigos científicos e seminários, dando pouca ênfase ao planejamento

			lúdicas e jogos.	prévio da disciplina. As experiências foram feitas em quatro instituições diferentes, não havendo a integração das propostas em uma única IES.
02	SILVA, Marco Aurélio Gaciotto et al., 2009)	Brasil	Ensino utilizando EAD, TV Digital e colaboração	Foca o modelo educacional apenas no desenvolvimento de projetos voltados para ensino à distância através da TV Digital.
03	ANDRADE, Rossana Maria de Castro et al., 2009)	Brasil	Metodologia de ensino baseada em projetos	A metodologia utiliza a abordagem de ensino tradicional, acrescentando o desenvolvimento de um projeto de <i>software</i> .
04	(Zhang, J. & Li, J, 2010)	China	Ensino com ênfase em estudo de casos	A proposta enfatiza apenas o estudo de caso como a única ferramenta didática
05	(Yang, C. & Liu, Y.,2009)	China	Orientação por projetos, estímulo à competição e mapas de conceitos	A proposta concentra-se apenas em dois pilares básicos: projetos e mapas de conceitos.
06	Tiejun, P. et al., 2010)	China	Metodologia de ensino voltada para a introdução de conceitos inovadores	A proposta concentra-se em trabalhar temas como TSP (processo de <i>software</i> em Equipe), PSP (processo de <i>software</i> pessoal), CSS (Sistemas de estudo Co-operativos), tentando formar profissionais conectados com a modernidade da área de ES. A viabilização torna-se difícil para a grande maioria das disciplinas ensinadas na graduação das IES do Brasil.
07	(Song, H. et al., 2009);	China	Educação personalizada voltada para formar talentos	Traz uma proposta de oferecimento de tratamento diferenciado em função do perfil dos alunos, restringindo-se a projetos e estudos de casos, sendo uma proposta de difícil operacionalização.
08	(Yadav, S. S. & Xiahou, J.,2010)	China	Metodologia orientada por projetos	Concentra-se unicamente no uso de projetos integrados ao processo educacional.
09	(Rico, D. F. & Sayani, H. H.,2009)	Eua	Utilização de métodos ágeis	Ênfase no uso de métodos ágeis no ensino de ES na implementação do projeto final do curso, sendo uma proposta até certo ponto convencional.
10	(Richardson, I. & Delaney, Y., 2009)	Irlanda	Ensino baseado em problemas	Utiliza apenas a técnica PBL (aprendizagem baseada em problemas).
11	(Qiu, M. & Chen, L.,2010)	China	Metodologia baseada em problemas e EAD	Enfatiza PBL e uso de recursos EAD, deixando de lado outras ferramentas didáticas possíveis de serem usadas..
12	(Mottok, J. & Gardeia, A.,2011)	Alemanha	Cooperação entre o ensino básico e o	Baseia-se na integração com o ensino médio e no

			universitário usando projetos de robôs LEGO	desenvolvimento de projetos na área de robótica. Boa estratégia, mas de difícil implementação universalizada por conta das questões burocráticas.
13	(Mirian-Hosseiniabadi, S.-H. et al, 2010)	Iran	Metodologia baseada em experimentos para verificar a validade das técnicas da Engenharia de Software.	A proposta foca-se nos experimentos, deixando de lado outras estratégias possíveis de serem utilizadas, como por exemplo, os estudos de caso e a implementação de projetos.
14	(Li, Z.,2009)	China	Metodologia baseada em projetos e cultura de excelência	A proposta busca ancorar-se na qualidade total e nos mais modernos conceitos da ES. Restrita e de difícil implementação generalizada.
15	(Jianguo, L.,2009)		Interação entre pesquisa e ensino usando projetos	Baseia-se na idéia de que a aprendizagem pode ser potencializada na integração dos alunos de graduação a projetos de pesquisa.
16	(Jia, Y.,2010)	China	Metodologia baseada em estudo de casos	Centra-se apenas nos estudos de casos como ferramenta básica de ensino.
17	(Gotel, O., 2009)	EUA / Camboja	Colaboração à distância à base de projetos	Enfatiza apenas duas ferramentas básicas: pedagogia de projetos e recursos EAD.
18	(Giraldo, F. D.,2009)	Colômbia	Colaboração multi-institucional e EAD	Utiliza a abordagem tradicional, focando na colaboração e no uso de recursos EAD.
19	(Garousi, V.,2010)	Canadá	Pedagogia de projetos e revisão em pares	A proposta foca apenas na pedagogia de projetos e revisão em pares.
20	(Gao, Y. & Feng, X., 2009)	China	Metodologia orientada à introdução de técnicas pedagógicas inovadoras.	A proposta visa sensibilizar os alunos para a utilização de novas propostas de ferramentas e técnicas ao longo do processo ensino-aprendizagem, sendo de difícil viabilização no universo cotidiano da grande maioria das instituições de ensino.
21	(Garg, K. & Varma, V.,2009)	Índia	Avaliação baseada em estudo de casos	Enfatiza a atividade de ensino-aprendizagem no estudo de caso como único elemento central.
22	(Dekhane, S. & Tsoi, M. Y.,2010)	EUA	Cooperação, interdisciplinaridade e projetos	A proposta enfatiza a cooperação, a integração de disciplinas e a implementação de projetos, desprezando outras ferramentas didáticas importantes.
23	(Costa-Soria et al, C.,2009)	Espanha	Engenharia reversa	Foca a estratégia de ensino na análise da engenharia reversa de produtos reais disponíveis no mercado.
24	(Coccoli, M. et al,2011)	Itália	Pedagogia de projetos e colaboração com comunidades de	Enfatiza apenas em projetos e colaboração com comunidades de desenvolvedores.

			desenvolvedores	
25	(Chen, J. et al,2009)	China	Metodologia baseada em experimentos, trabalho em grupo, estudos de casos e cooperação	Baseia-se em projetos, trabalho em grupo, estudo de casos e cooperação, deixando de focar em ferramentas importantes tais como a produção de artigos e a apresentação de ,seminários .
26	SOUZA, M. et al, 2010)	Brasil	Metodologia de ensino baseada em jogos e simulação	Ênfase na pedagogia de jogos sérios e simulação.
27	(ANISSETY, Priatham; YOUNG, Paul, 2011)	EUA	Metodologia baseada em problemas, colaboração e projetos em grupo.	Foca em 3 ferramentas didáticas apenas: PBL, colaboração e projetos.

Quadro 3: Avaliação dos trabalhos selecionados

Fonte: Criação Própria

5. CLASSIFICAÇÃO DOS TRABALHOS SELECIONADOS

A partir de uma análise criteriosa dos artigos selecionados, pode-se apresentar, no Quadro 3, uma classificação desses trabalhos com base nas suas características básicas:

	Referência	País(es) de origem	Abordagem Utilizada	Nº de Etapas	Número de instituições
1	(PRIKLADNICKI, Rafael et al., 2009)	Brasil	Metodologia baseada em dinâmicas de grupo, EAD, experimentos, projetos, atividade lúdicas, jogos	1	4
2	SILVA, Marco Aurélio Gaciotto et al., 2009)	Brasil	EAD, TV Digital e colaboração	1	4
3	ANDRADE, Rossana Maria de Castro et al, 2009)	Brasil	Metodologia de ensino	1	1
4	(Zhang, J. & Li, J, 2010)	China	Ensino com ênfase em estudo de casos	1	1
5	(Yang, C. & Liu, Y.,2009)	China	Orientação por projetos, estímulo à competição e mapas de conceitos	1	1
6	(Yadav, S. S. & Xiahou, J.,2010)	China	Metodologia orientada por projetos	1	1
7	Tiejun, P. et al., 2010)	China	Metodologia de ensino voltada para a introdução de conceitos inovadores	1	1
8	(Song, H. et al., 2009);	China	Educação personalizada voltada para formar talentos	1	1
9	(Rico, D. F. & Sayani, H. H.,2009)	Eua	Utilização de métodos ágeis	1	1
10	(Richardson, I. & Delaney, Y., 2009)	Irlanda	Ensino baseado em problemas	1	1
11	(Qiu, M. & Chen, L.,2010)	China	Metodologia baseada em problemas e EAD		
12	(Mottok, J. & Gardeia, A.,2011)	Alemanha	Cooperação entre o ensino básico e o universitário usando projetos de robôs LEGO	1	Várias
13	(Mirian-Hosseinabadi, S.-H. et al, 2010)	Iran	Metodologia baseada em experimentos	1	1
14	(Li, Z.,2009)	China	Metodologia baseada em projetos e cultura de excelência	1	1
15	(Jianguo, L,2009)	China	Interação entre pesquisa e ensino usando projetos	1	1
16	(Jia, Y.,2010)	China	Metodologia baseada em estudo de casos	1	1

	Referência	País(es) de origem	Abordagem Utilizada	Nº de Etapas	Número de instituições
17	(Gotel, O., 2009)	EUA / Camboja	Colaboração à distância à base de projetos	1	2
18	(Giraldo, F. D., 2009)	Colômbia	Colaboração multi-institucional EAD	1	4
19	(Garousi, V., 2010)	Canadá	Pedagogia de projetos e revisão em pares	1	1
20	(Garg, K. & Varma, V., 2009)	Índia	Avaliação baseada em estudo de casos	1	1
21	(Gao, Y. & Feng, X., 2009)	China	Metodologia orientada à inovação	1	1
22	(Dekhane, S. & Tsoi, M. Y., 2010)	EUA	Cooperação, interdisciplinaridade e projetos	1	1
23	(Costa-Soria et al, C., 2009)	Espanha	Engenharia reversa	1	1
24	(Coccoli, M. et al, 2011)	Itália	Pedagogia de projetos e colaboração com comunidades de desenvolvedores	1	1
25	(Chen, J. et al, 2009)	China	Metodologia baseada em experimentos, trabalho em grupo, estudos de casos e cooperação	1	1
26	SOUZA, M. et al, 2010)	Brasil	Metodologia de ensino baseada em jogos e simulação	1	1
27	(ANISETTY, Priatham; YOUNG, Paul, 2011)	EUA	Metodologia baseada na resolução de problemas, colaboração e projetos	1	1

Quadro 4: Classificação dos trabalhos selecionados

Fonte: Elaboração própria

6. CONSIDERAÇÕES FINAIS E DISCUSSÃO DOS RESULTADOS

É importante ressaltar que a grande quantidade de artigos obtida demonstra claramente que a preocupação com a melhoria do ensino de Engenharia de software é mundial. Em nossa pesquisa, encontramos trabalhos de pesquisas sendo feitos sobre o tema em dezenas de países ao redor do mundo. O número só não é maior devido às restrições impostas nos critérios de busca, como, por exemplo, apenas foram considerados os trabalhos publicados a partir de 2009, tendo sido descartada uma quantidade significativa publicada nos anos anteriores.

A maioria das pesquisas sobre o tema é predominantemente feita no âmbito de apenas uma instituição de ensino, demonstrando a dificuldade ainda existente de realizar trabalhos multi-institucionais.

No Gráfico 1, pode-se observar a distribuição das iniciativas propostas para a melhoria do ensino de engenharia de software nos trabalhos selecionados:

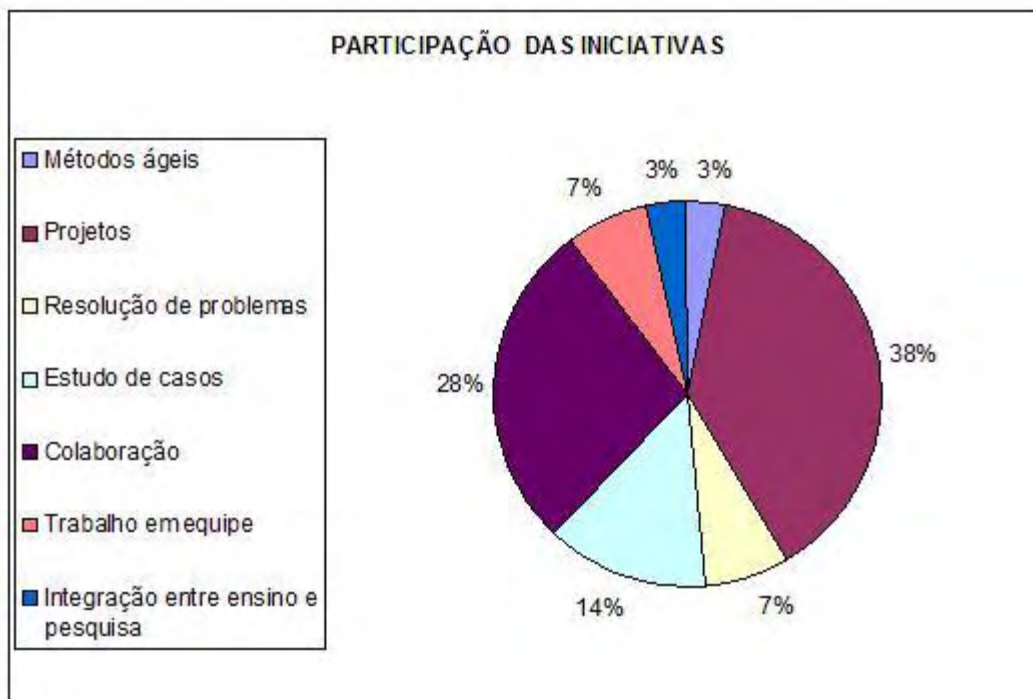


Gráfico 1: Participação percentual das iniciativas propostas

O presente trabalho reforça o sentimento dos pesquisadores do LABES quanto à pertinência e à atualidade do tema em questão, sinalizando claramente a necessidade de aprofundamento das pesquisas sobre o tema.

Uma limitação do presente trabalho é o fato de que ele foi realizado por apenas um pesquisador, o que pode gerar alguma distorção na qualidade final do mesmo, sendo importante, a título de trabalhos futuros, que os pesquisadores do LABES procedam uma revisão aprofundada sobre o seu teor no sentido de dar-lhe maior fidedignidade.

7. REFERÊNCIAS

- [1] ANISETTY, P.; YOUNG, P. **Collaboration problems in conduction a group project in a software engineering course.**Journal of Computing Sciences in Colleges, Volume 26 Issue 5, Mai. 2011.
- [2] COSTA-SORIA, C.; LLAVADOR, M. & DEL CARMEN PENADES, M. **An approach for teaching software engineering through reverse engineering.***Proc. EAEEIE Annual Conf, 2009*, 1-6
- [3] COCCOLI, M.; STANGANELLI, L. & MARESCA, P. **Computer Supported Collaborative Learning in software engineering.** *Proc. IEEE Global Engineering Education Conf. (EDUCON), 2011*, 990-995
- [4] CHEN, J.; LU, H.; AN, L. & ZHOU, Y. **Exploring teaching methods in software engineering education.***Proc. 4th Int. Conf. Computer Science & Education ICCSE '09, 2009*, 1733-1738
- [5] DEKHANE, S. & TSOI, M. Y. **Work in progress --- Inter-disciplinary collaboration for a meaningful experience in a software development course.** *Proc. IEEE Frontiers in Education Conf. (FIE), 2010*
- [6] JIANGUO, L. **Combination of Research and Teaching in Software Engineering Education.** *Proc. WASE Int. Conf. Information Engineering ICIE '09, 2009, 2, 437-440*
- [7] JIA, Y. **Improving software engineering courses with case study approach.** *Proc. 5th Int Computer Science and Education (ICCSE) Conf, 2010*, 1633-1636
- [8] GAO, Y. & FENG, X. **Incorporating innovation-oriented education into the software engineering course.** *Proc. IEEE Int. Conf. Systems, Man and Cybernetics SMC 2009, 2009.*
- [9] GARG, K. & VARMA, V. **Case Studies as Assessment Tools in Software Engineering Classrooms.***Proc. 22nd Conf. Software Engineering Education and Training CSEET '09, 2009, 8-11*
- [10] _____. **Case Studies as Assessment Tools in Software Engineering Classrooms.** *Proc. 22nd Conf. Software Engineering Education and Training CSEET '09, 2009*, 8-11
- [11] GAROUSI, V. **Applying Peer Reviews in Software Engineering Education: An Experiment and Lessons Learned.***#IEEE_J_EDU#, 2010, 53,182-193*
- [12] GIRALDO, F. D.; COLLAZOS, C. A.; OCHOA, S. F.; ZAPATA, S. & DE CLUNIE, G. T. **Teaching Software Engineering from a Collaborative Perspective: Some Latin-American Experiences.***Proc. Workshop Database and Expert Systems Applications (DEXA), 2010, 97-101*

- [13] GOTEL, O.; KULKARNI, V.; SAY, M.; SCHARFF, C. & SUNETNANTA, T. **A Global and Competition-Based Model for Fostering Technical and Soft Skills in Software Engineering Education.***Proc. 22nd Conf. Software Engineering Education and Training CSEET '09, 2009, 271-278*
- [14] LI, Z.; LI, J.-M.; GUAN, Y.-C. & YI, K. **Practice of Actual Project Driven Teaching Model on Software Engineering Major.***Proc. Int. Conf. Information Engineering and Computer Science ICIECS 2009,*
- [15] LIBÂNEO, J. C. **Didática.** Editora Cortez. São Paulo, 1990.
- [16] LU, X.; XUE, H.; OUYANG, H. & TU, M. **Practice of using suite of teaching to build the students' forming capacity of software engineering.** *Proc. Second Int Communication Systems, Networks and Applications (ICCSNA) Conf, 2010, 1, 308-311*
- [17] MEDEIROS, João B. **Redação Científica: a prática de fichamentos, resumos, resenhas.** 4. ed. São Paulo:Atlas,2000.
- [18] MINISTÉRIO DE CIÊNCIA E TECNOLOGIA. SECRETARIA DE POLÍTICA DE INFORMÁTICA. **Pesquisa de Qualidade no Setor de Software Brasileiro 2009.** Brasília, 2010. Disponível em www.mct.gov.br. Acessado em 01 de fev. de 2011.
- [19] MIRIAN-HOSSEINABADI, S.-H.; AGHAKASIRI, Z.; SADEGHI, A.; DELFANI, P. & GHANDEHARI, M. **Emphasizing experiences in teaching software engineering courses.***Proc. 2nd Int Education Technology and Computer (ICETC) Conf, 2010, 2*
- [20] MOTTOK, J. & GARDEIA, A. **The Regensburg Concept of P-Seminars --- How to organize the interface between secondary school and university education to create a didactic cooperation between teaching and learning of Software Engineering with Lego Mindstorms NXT Embedded Robot Systems.** *Proc. IEEE Global Engineering Education Conf. (EDUCON), 2011, 917-920*
- [21] PRIKLADNICKI, Rafael et al. **Ensino de Engenharia de Software: desafios, estratégias de ensino e lições aprendidas.** Disponível em: http://fees.inf.puc-rio.br/FEESArtigos/FEES09/FEES_2.pdf. Acesso em: 02 fev. de 2011.
- [22] QIU, M. & CHEN, L. **A Problem-Based Learning Approach to Teaching an Advanced Software Engineering Course.***Proc. Second Int Education Technology and Computer Science (ETCS) Workshop, 2010, 3, 252-255*
- [23] RICO, D. F. & SAYANI, H. H. **Use of Agile Methods in Software Engineering Education.** *Proc. AGILE '09. Agile Conf, 2009, 174-179*
- [24] RICHARDSON, I. & DELANEY, Y. **Problem Based Learning in the Software Engineering Classroom.** *Proc. 22nd Conf. Software Engineering Education and Training CSEET '09, 2009, 174-181*

- [25] SOUZA, Mariane et al. **SPARSE: uma abordagem para o ensino de engenharia de software baseada em jogos e simulação.** Anais do XXI Simpósio Brasileiro de Informática na Educação. João Pessoa, 2010.
- [26] SONG, H.; Li, X.; WANG, P. & ZHAO, D. **Improving Personalized Talent Education by Strengthening Specialty Direction of Information System in Software Engineering.** *Proc. Second Int. Conf. Education Technology and Training ETT '09*, **2009**, 16-18
- [27] TIEJUN, P.; YUEFENG, F.; LEINA, Z.; XIAOYAN, Y. & YUNPENG, L. **Research of High-quality innovative and pioneering undergraduate training model of Software Engineering.** *Proc. 2nd Int Education Technology and Computer (ICETC) Conf*, **2010**, 3
- [28] YADAV, S. S. & XIAHOU, J. **Integrated project based learning in software engineering education.** *Proc. Int Educational and Network Technology (ICENT) Conf*, **2010**, 34-36
- [29] YANG, C. & LIU, Y. **Teaching Reform and Practice on the Software Engineering Course.** *Proc. 1st Int Information Science and Engineering (ICISE) Conf*, **2009**, 3470-3473
- [30] ZHANG, J. & LI, J. **Teaching Software Engineering Using Case Study.** *Proc. IntBiomedicalEngineeringand Computer Science (ICBECS) Conf*, 2010

APÊNDICE B

Pesquisa de opinião sobre o ensino de Engenharia de Software

**PESQUISA DE OPINIÃO SOBRE A AVALIAÇÃO DOS ALUNOS,
PROFISSIONAIS, DOCENTES E COORDENADORES SOBRE AS
DISCIPLINAS CURSADAS/MINISTRADAS NA ÁREA DE ENGENHARIA DE
SOFTWARE**

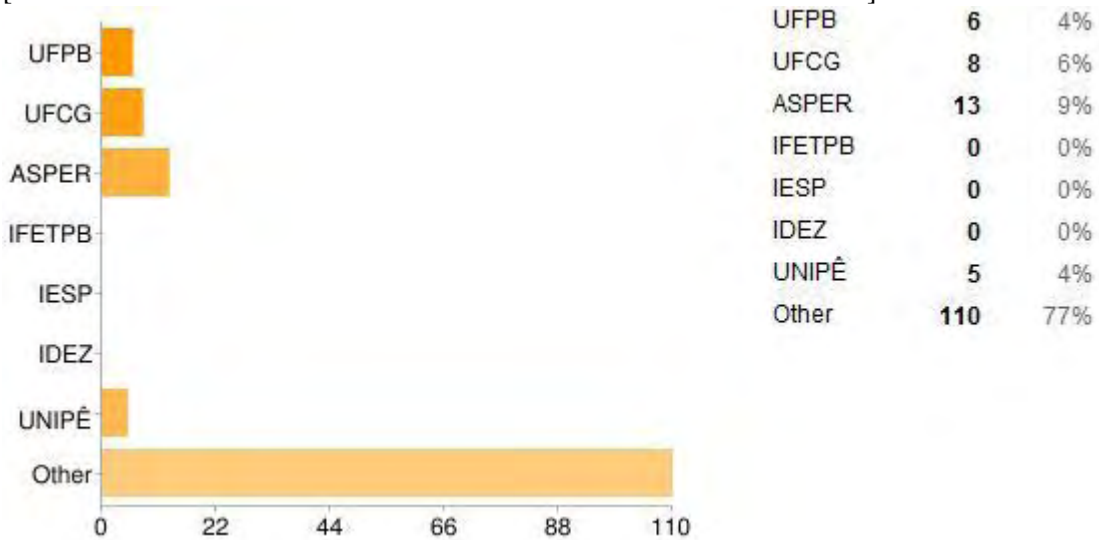
UFPB - MESTRADO EM INFORMÁTICA

Formulário de Pesquisa sobre o ensino de Engenharia de Software - Para Uso do(a) aluno(a) / Profissional

Este formulário faz parte de um projeto de pesquisa sobre o ensino de Engenharia de Software desenvolvido pelos pesquisadores do LABES - Laboratório de Engenharia de Software da UFPB.

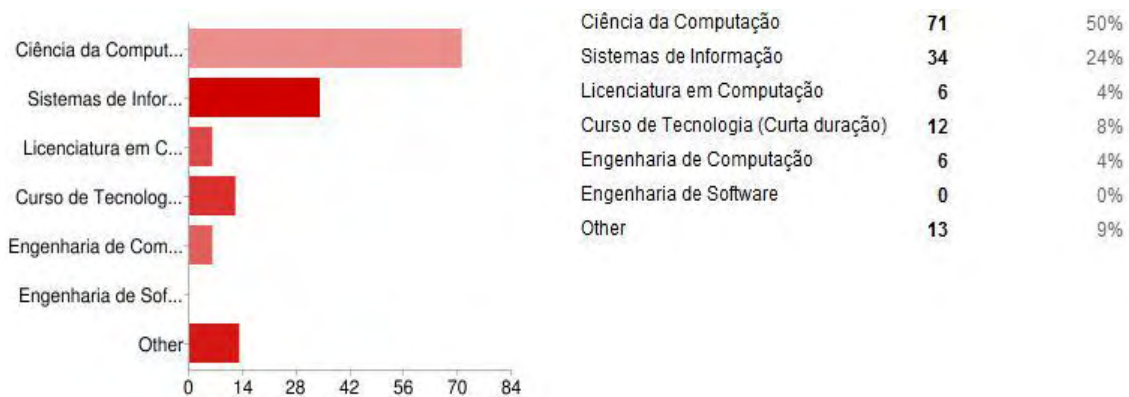
1. Identificação: Nome: [_____]
 E-mail: [_____]

2. Qual a sua instituição de ensino superior na área de Computação/Informática?
 [_____]



3. Qual é a denominação do seu curso superior na área de informática ?

- [] – Ciência Computação
 [] – Sistemas Informação
 [] – Licenciatura Computação
 [] – Curso Tecnologia _____
 [] – Engenharia Computação
 [] – Eng. Software
 Outro: [_____]



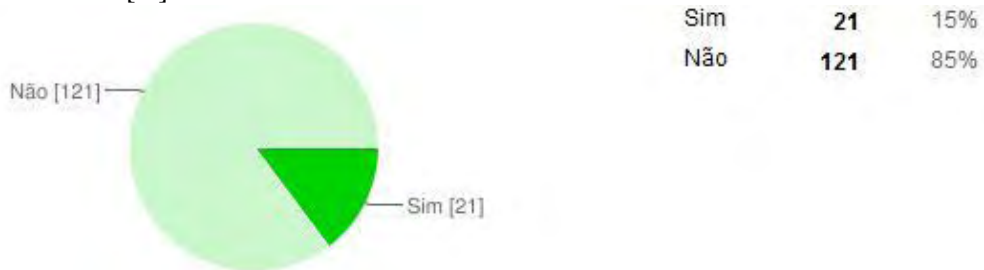
4. Você tem quanto tempo de experiência em desenvolvimento de software ?*

- Mais de 3 anos
- Mais de 2 e menos de 3 anos
- Mais de um ano e menos de 1 ano
- Menos de 1 ano
- Não tenho experiência



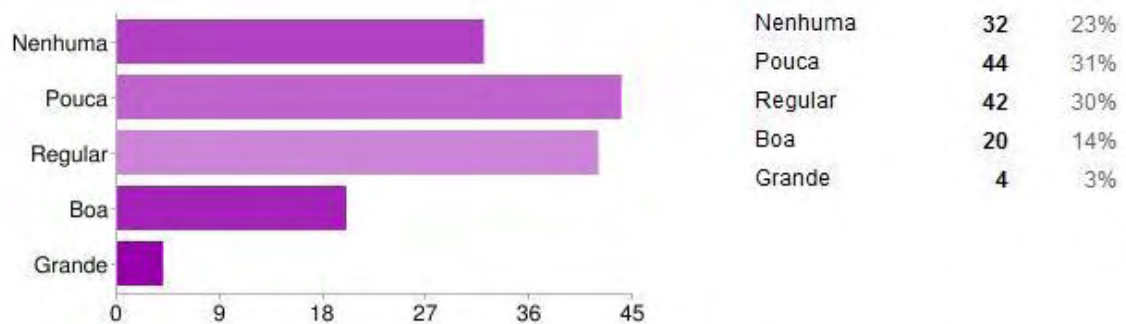
5. Você já tinha experiência no desenvolvimento de software antes de entrar na sua IES ?

- Sim
- Não



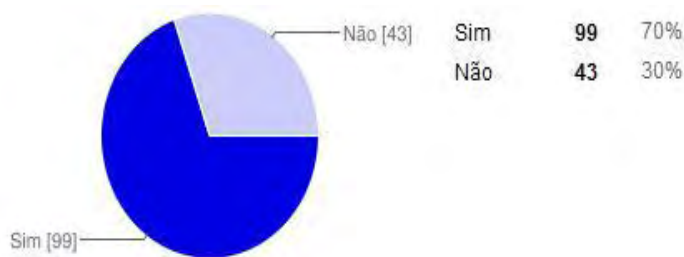
6. Como você julga sua habilidade em desenvolver software ANTES de cursar a disciplina de Engenharia de Software ?*

- Nenhuma
- Pouca
- Regular
- Boa
- Grande



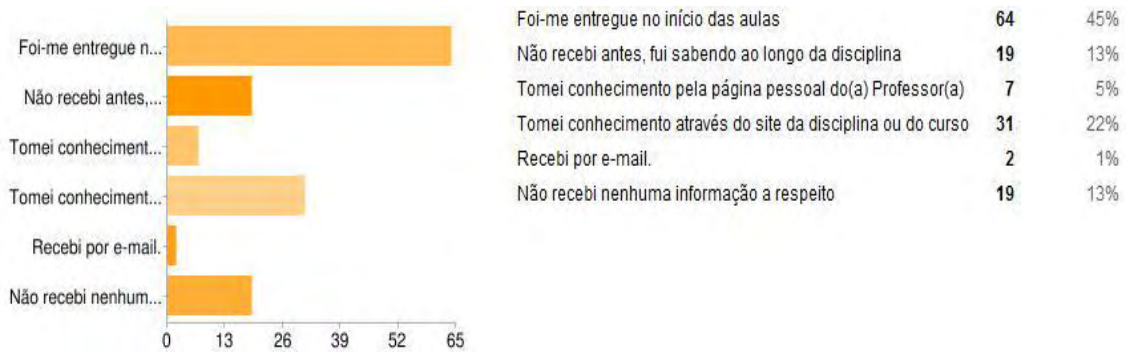
7. Você tomou conhecimento do programa e do planejamento didático da disciplina no início da mesma ?

- Sim
- Não



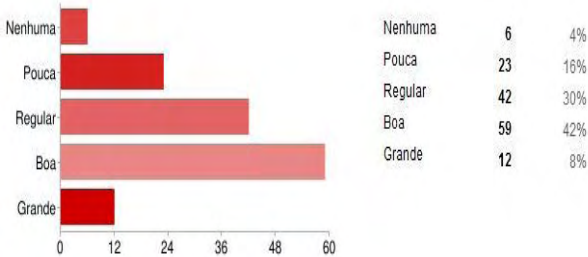
8. Se tomou conhecimento do programa/planejamento da disciplina antes das aulas, de que forma ?

- Foi-me entregue no início das aulas
- Não recebi antes, fui sabendo ao longo da disciplina
- Tomei conhecimento pela página pessoal do(a) Professor(a)
- Tomei conhecimento através do site da disciplina ou do curso
- Recebi por e-mail
- Não recebi nenhuma informação a respeito



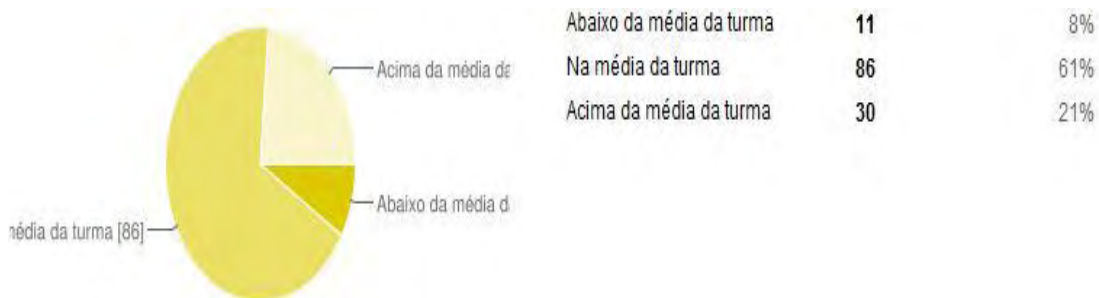
9. Como você julga suas habilidades em desenvolver software atualmente DEPOIS de cursar a(s) disciplina(s) de Engenharia de Software ? Suas habilidades são compatíveis com as exigências do mercado de trabalho ?

- Nenhuma
- Pouca
- Regular
- Boa
- Grande



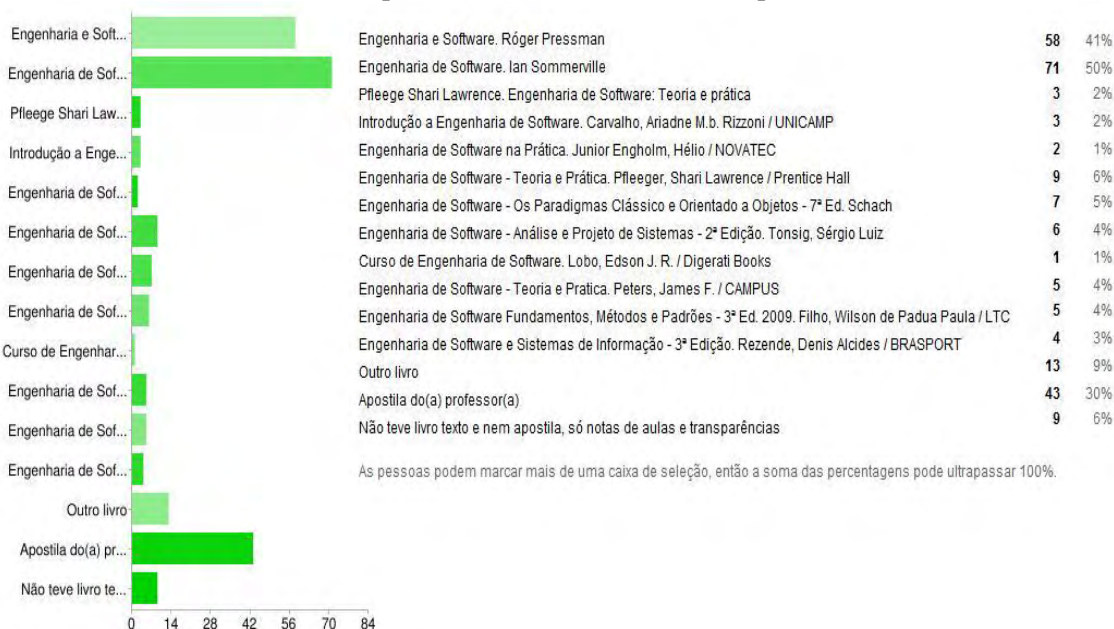
10. Em relação aos demais colegas da sua turma, você se considera ?

- Abaixo da média da turma
- Na média da turma
- Acima da média da turma



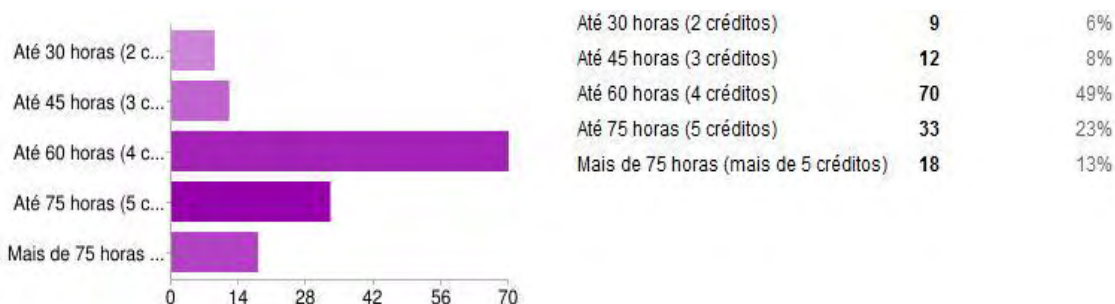
11. Qual foi o(s) livro(s) textos da(s) disciplina(s) ?

- Engenharia e Software. Róger Pressman
- Engenharia de Software. Ian Sommerville
- Pfllege Shari Lawrence.Engenharia de Software: Teoria e prática
- Introdução a Engenharia de Software Carvalho, Ariadne M.b. Rizzoni / UNICAMP
- Engenharia de Software na Prática Junior, Hélio Engholm / NOVATEC
- Engenharia de Software - Teoria e Prática Pflleeger, Shari Lawrence / Prentice Hall
- Engenharia de Software - Os Paradigmas Clássico e Orientado a Objetos - 7ª Ed.
- Engenharia de Software - Análise e Projeto de Sistemas - 2ª Ed. Tonsig, Sérgio Luiz
- Curso de Engenharia de Software Lobo, Edson J. R. / Digerati Books
- Engenharia de Software - Teoria e Pratica Peters, James F. / CAMPUS
- Engenharia de Software Fundamentos, Métodos e Padrões - 3ª Ed. 2009 Filho, Wilson de Padua Paula / LTC
- Engenharia de Software e Sistemas de Informação - 3ª Edição Rezende, Denis Alcides / BRASPORT
- Outro livro []
- Apostila do(a) professor(a)
- Não teve livro texto e nem apostila, só notas de aulas e transparências



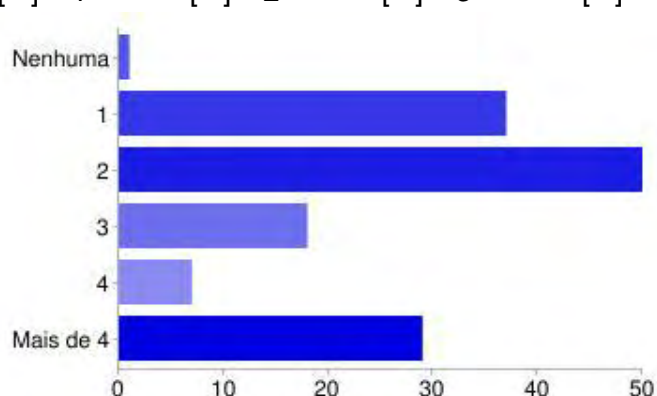
12. A(s) disciplina(s) cursada(s) na área de Engenharia de Software tinha(m), em média, quantas horas semestrais?

- Até 30 horas (2 créditos)
- Até 45 horas (3 créditos)
- Até 60 horas (4 créditos)
- Até 75 horas (5 créditos)
- Mais de 75 horas (mais de 5 créditos)



13. No seu curso, tem quantas disciplinas na área de Engenharia de Software ?

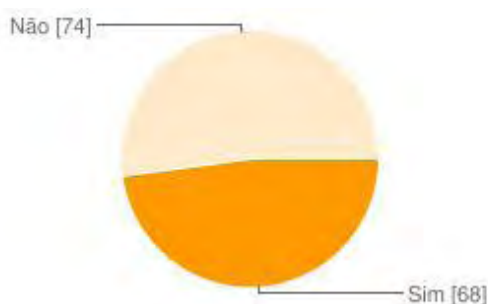
- 1 - 2 - 3 - 4 - Mais de 4



Nenhuma	1	1%
1	37	26%
2	50	35%
3	18	13%
4	7	5%
Mais de 4	29	20%

14. Você considera a carga horária voltada para a área de Engenharia de Software suficiente para um bom aprendizado ?

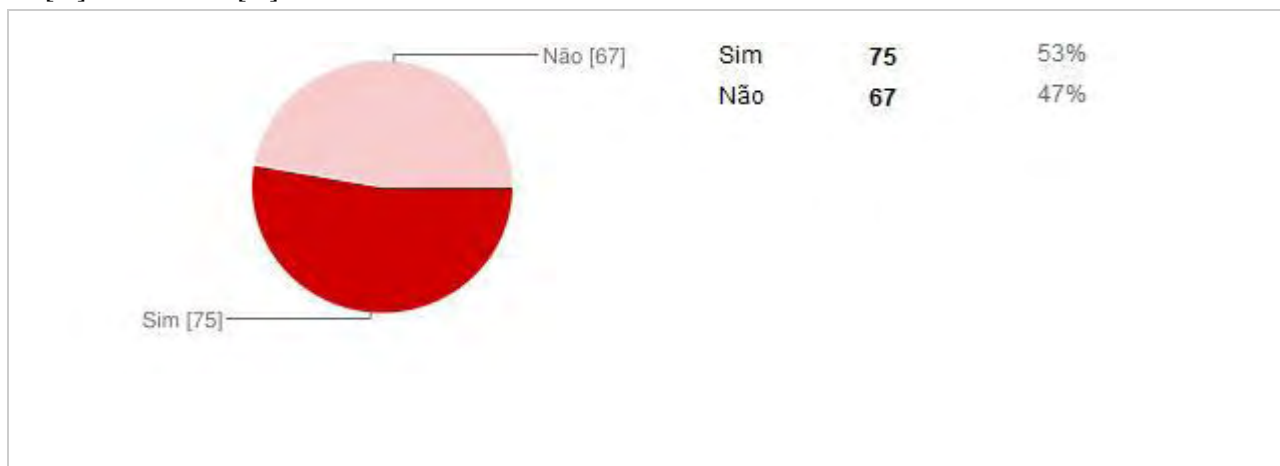
- Sim - Não



Sim	68	48%
Não	74	52%

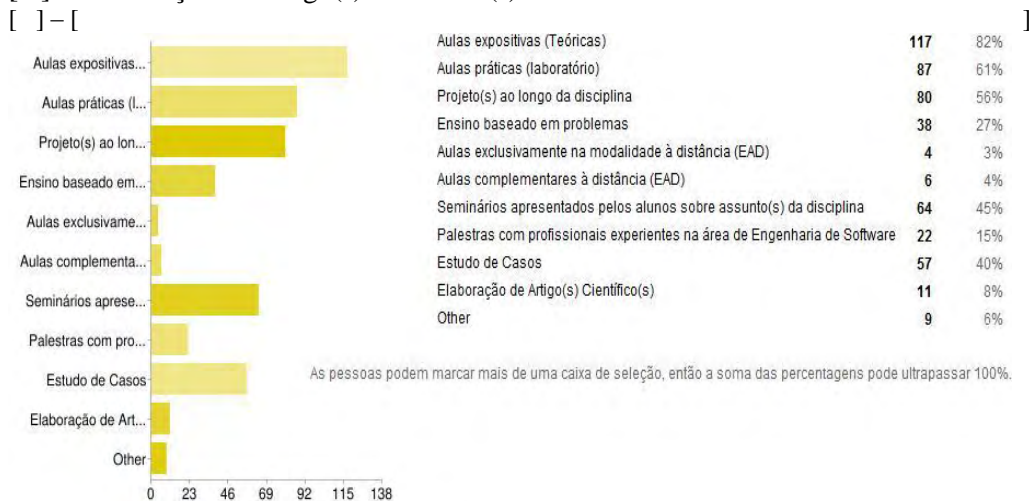
15. Nas disciplinas cursadas na área de Engenharia de Software, você participou da realização de algum projeto de desenvolvimento ao longo do curso?

- Sim - Não



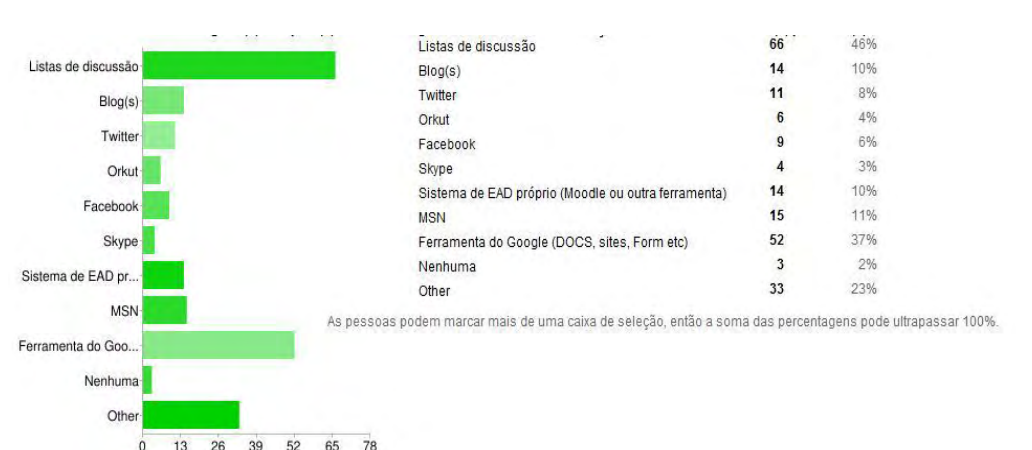
16. O(s) docente(s) ao longo da(s) disciplina(s), utilizaram que ferramentas didáticas ? Obs.: Podem ser escolhidas mais de uma alternativa.

- Aulas expositivas
- Aulas práticas (laboratório)
- Projeto(s) ao longo da disciplina
- Ensino baseado em problemas
- Aulas exclusivamente na modalidade à distância (EAD)
- Aulas complementares à distância (EAD)
- Seminários apresentados pelos alunos sobre assunto(s) da disciplina
- Palestras com profissionais experientes na área de Engenharia de Software
- Estudo de Casos
- Elaboração de Artigo(s) Científico(s)



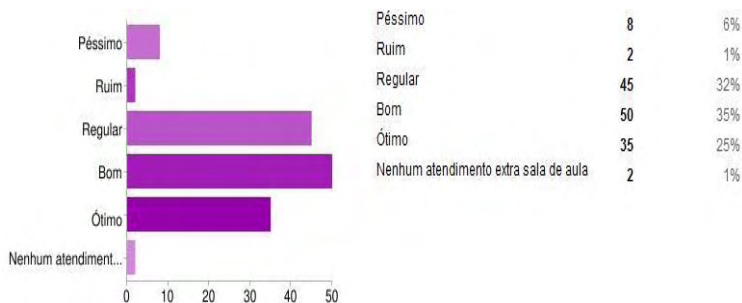
17. Você sabe informar se ao longo da(s) disciplina(s) foi utilizado algum mecanismo de comunicação extra sala de aula entre o(a) professor(a) e os alunos ?

- Listas de Discussão
- Blog(s)
- Twitter
- Orkut
- Facebook
- Skype
- Sistema de EAD próprio (Moodle ou outra ferramenta)
- MSN
- Ferramentas Google (DOCS, sites, Formetc)



18. Você considera que a qualidade do atendimento docente para dúvidas e/ou atendimento fora da sala de aula foi ?

- Péssimo - Ruim - Regular - Bom - Ótimo - Nenhum



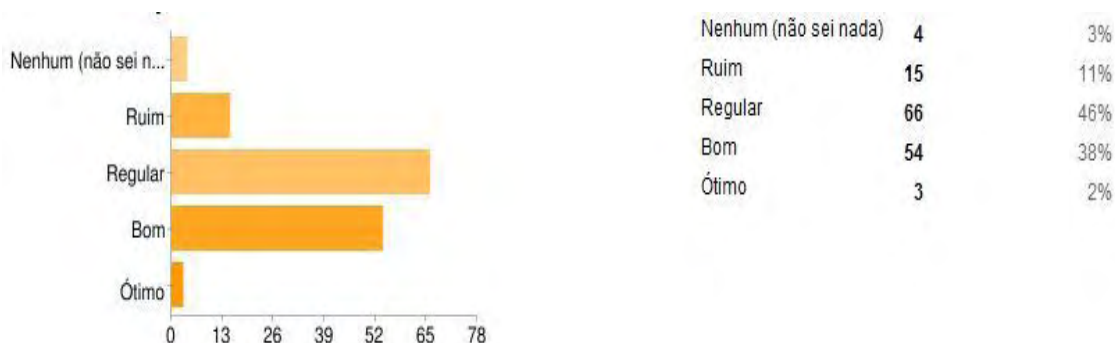
19. Com relação aos conteúdos de Engenharia de Software, você se considera apto a entrar no mercado de trabalho ?

- Sim - Não



20. Ainda com relação aos conteúdos de Engenharia de Software, você se considera em que nível de domínio do conhecimento da área ?

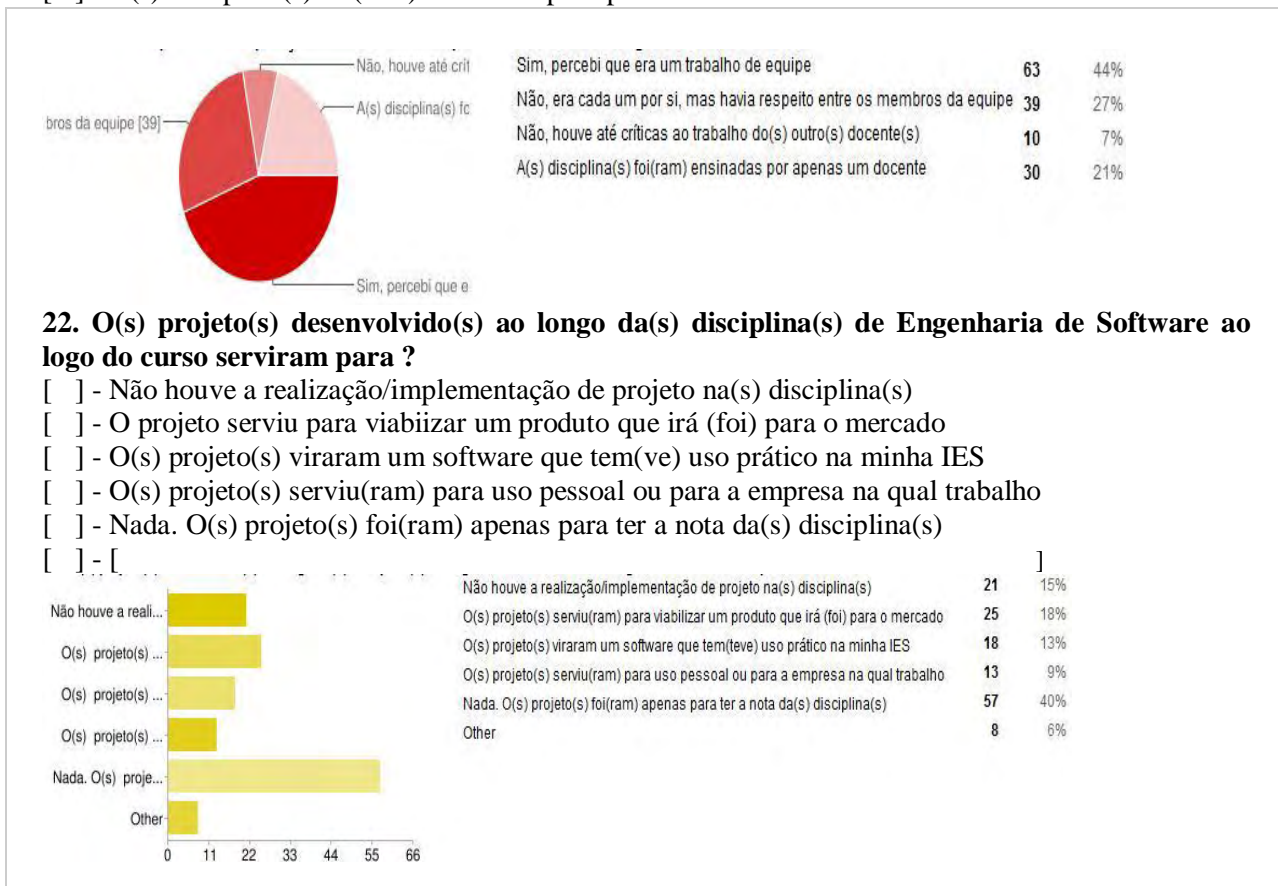
- Nenhum (não sei nada) - Ruim - Regular - Bom - Ótimo



21. Você avalia que houve cooperação entre os diferentes professores da área de Engenharia de Software durante o seu curso?

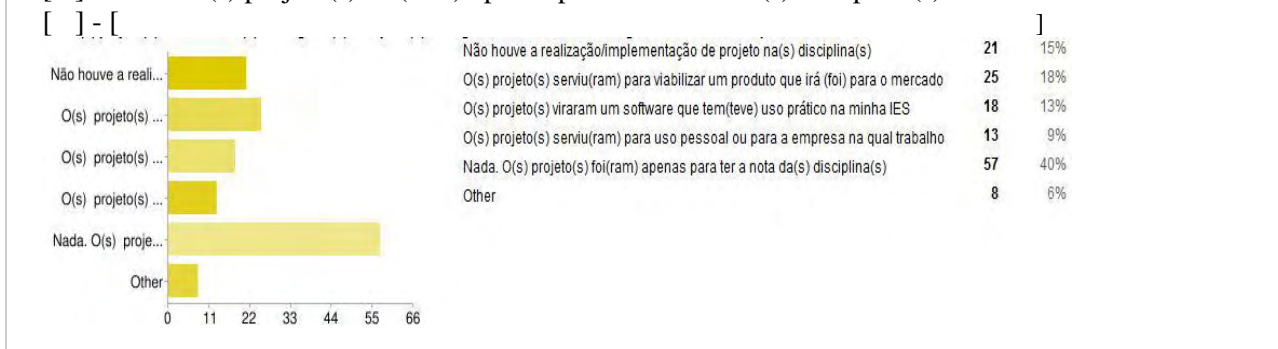
Cooperação na elaboração dos conteúdos, na continuidade dos assuntos/ projetos entre uma disciplina e outra, na utilização de recursos comuns

- Sim, percebi que era um trabalho de equipe
- Não, era cada um por si, mas havia respeito entre os membros da equipe
- Não, houve até críticas ao trabalho do(s) outro(s) docente(s)
- A(s) disciplina(s) foi(ram) ensinadas por apenas um docente



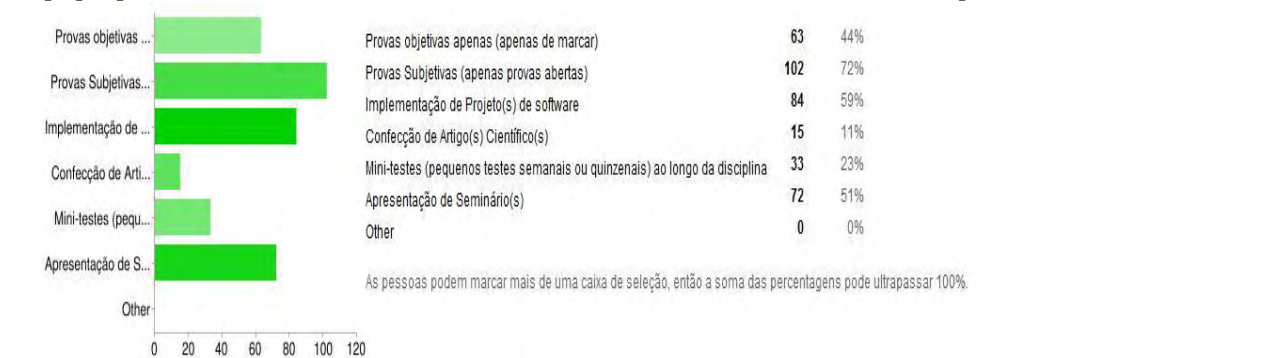
22. O(s) projeto(s) desenvolvido(s) ao longo da(s) disciplina(s) de Engenharia de Software ao longo do curso serviram para ?

- Não houve a realização/implementação de projeto na(s) disciplina(s)
- O projeto serviu para viabilizar um produto que irá (foi) para o mercado
- O(s) projeto(s) viraram um software que tem(ve) uso prático na minha IES
- O(s) projeto(s) serviu(ram) para uso pessoal ou para a empresa na qual trabalho
- Nada. O(s) projeto(s) foi(ram) apenas para ter a nota da(s) disciplina(s)



23. Qual(uais) mecanismos de avaliação foi(ram) utilizados para avaliar o(s) aluno(s) ao longo da(s) disciplina(s) ?

- Provas objetivas apenas (apenas de marcar)
- Provas Subjetivas (apenas provas abertas)
- Implementação de Projeto(s) de software
- Confecção de Artigo(s) Científico(s)
- Mini-testes (pequenos testes semanais ou quinzenais) ao longo da disciplina
- Apresentação de Seminário(s)



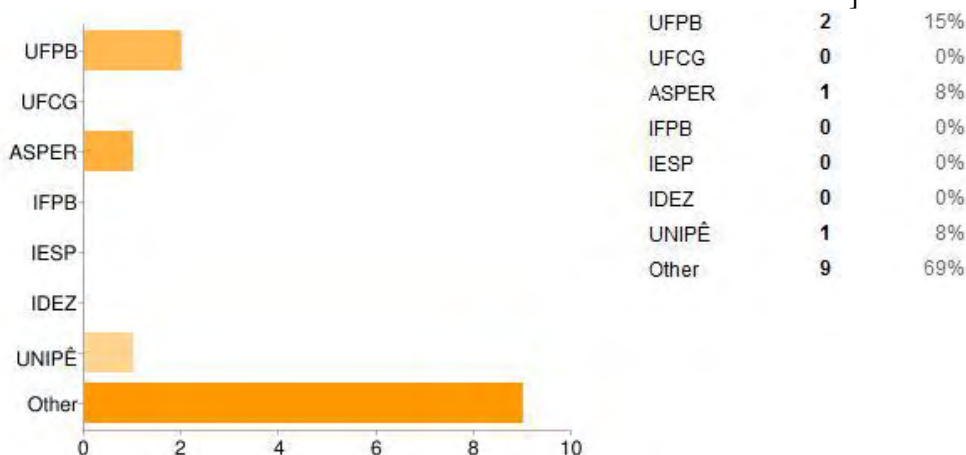
UFPB - MESTRADO EM INFORMÁTICA

Formulário de Pesquisa sobre o ensino de Engenharia de Software - Uso do(a) Professor(a) / Coordenador(a)

Este formulário faz parte de um projeto de pesquisa sobre o ensino de Engenharia de Software desenvolvido pelos pesquisadores do LABES - Laboratório de Engenharia de Software da UFPB.

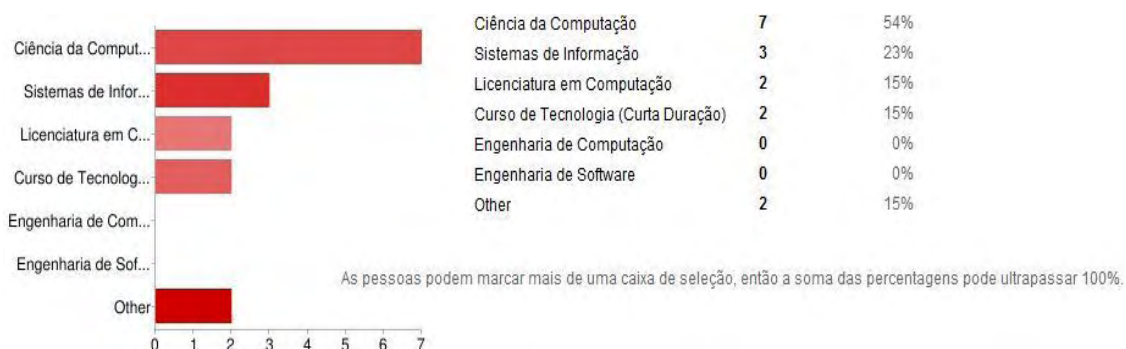
1. Identificação: Nome: []
E-mail: []

2. Qual a sua instituição de ensino superior na área de Computação / Informática ?
 []



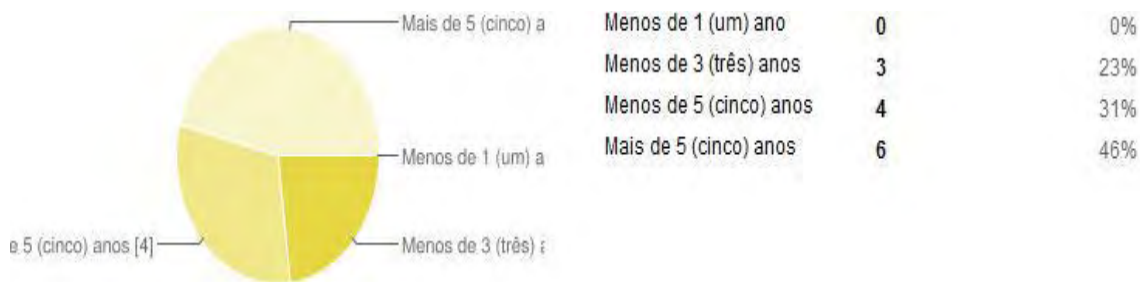
3. Qual é a denominação do seu curso superior na área de informática ?

- [] – Ciência Computação
 [] – Sistemas Informação
 [] – Licenciatura Computação
 [] – Curso Tecnologia _____
 [] – Engenharia Computação
 [] – Engenharia de Software
 Outro: []



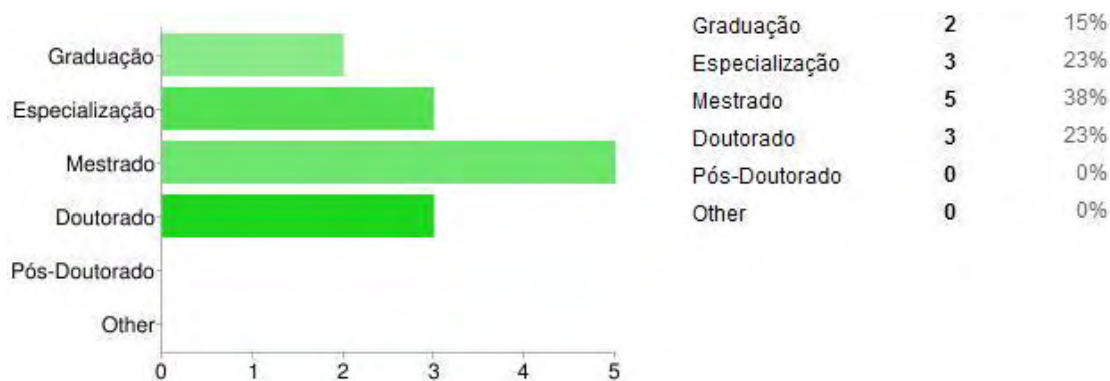
4. Você tem quanto tempo de atividade docente ?

- [] – Menos de um ano
 [] – Menos de 3 anos
 [] – Menos de 5 anos
 [] – Mais de 5 anos



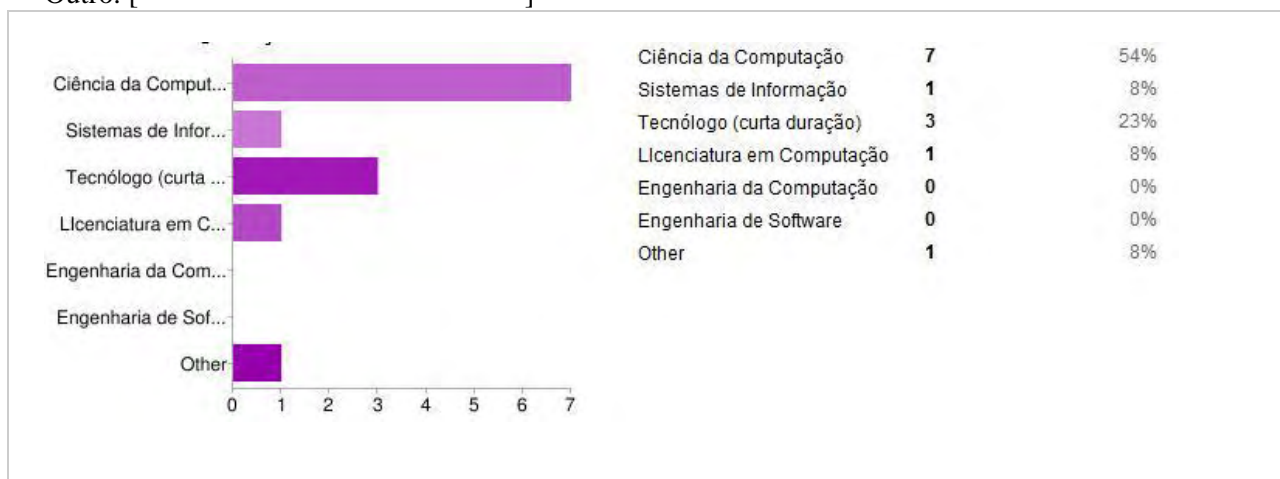
5. Atualmente a sua mais alta titulação é ?

- Graduação
- Especialização
- Mestrado
- Doutorado
- Pós-doutorado
- _____]



6. Seu curso de graduação foi ?

- Ciência Computação
- Sistemas Informação
- Licenciatura Computação
- Curso Tecnologia _____
- Engenharia Computação
- Eng. Software
- Outro: _____]



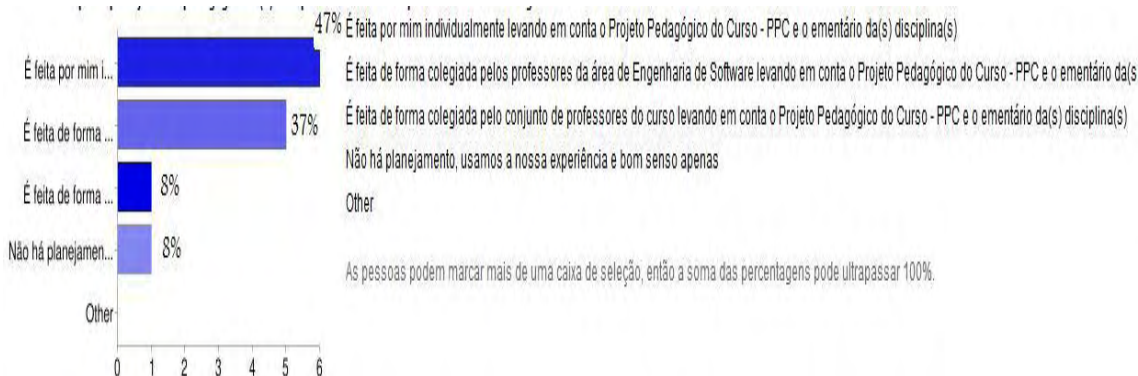
7. Sobre a etapa de planejamento pedagógico da(s) disciplina(s) ministrada(s) por você na área de Engenharia de Software:

– É feita por mim individualmente levando em conta o projeto pedagógico do curso e o ementários da(s) disciplina(s)

– É feita de forma colegiada pelos professores da área de Engenharia de Software levando em conta o projeto pedagógico do curso e o ementários da(s) disciplina(s)

– Não há planejamento, usamos a nossa experiência e o bom senso.

– []

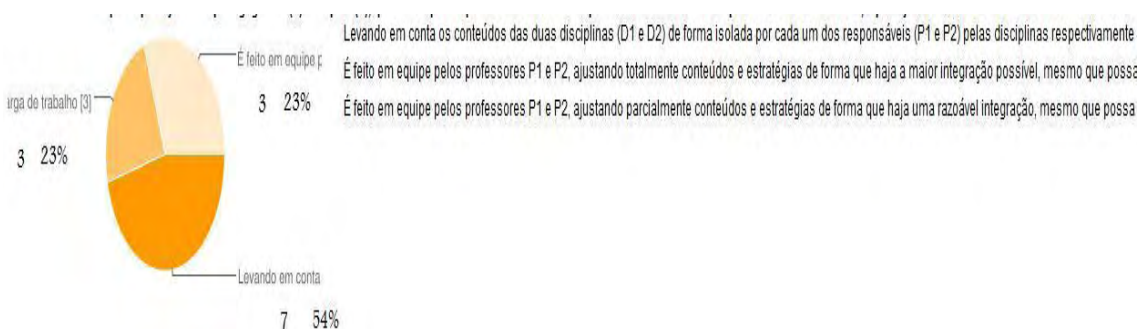


8. Ainda sobre a etapa de planejamento pedagógico da(s) disciplina(s), quando há pré-requisitos entre 2 disciplinas D1 e D2 ministradas por 2 docentes P1 e P2, o planejamento ocorre:

– Levando em conta os conteúdos das 2 disciplinas (D1 e D2) de forma isolada por cada um dos responsáveis (P1 e P2) pelas disciplinas respectivamente

– É feito em equipe pelos professores P1 e P2, ajustando totalmente conteúdos e estratégias de forma que haja a maior integração possível, mesmo que possa haver sobrecarga de trabalho

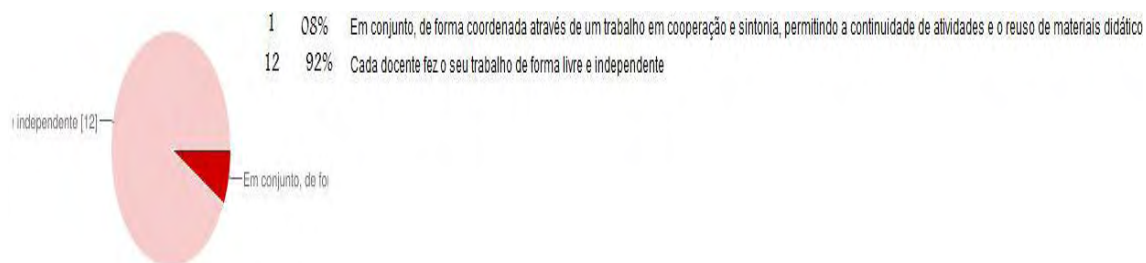
– É feito em equipe pelos professores P1 e P2, ajustando parcialmente conteúdos e estratégias de forma que haja uma razoável integração possível, mesmo que possa haver alguma sobrecarga de trabalho



9. O trabalho docente ao ensinar as disciplinas da área de Engenharia de Software é(foi) feito ?

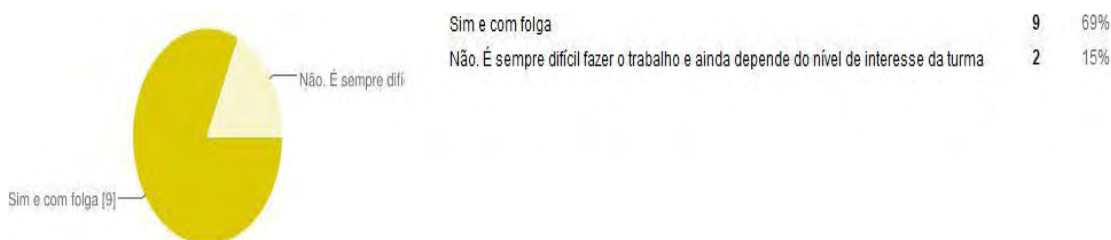
– Em conjunto, de forma coordenada através de um trabalho em cooperação e sintonia, permitindo a continuidade de atividades e o reuso de materiais didáticos

– Cada docente faz o seu trabalho de forma livre e independente



10. O conteúdo programático previsto é trabalhado com a qualidade desejada e necessária ?

- Sim – Não



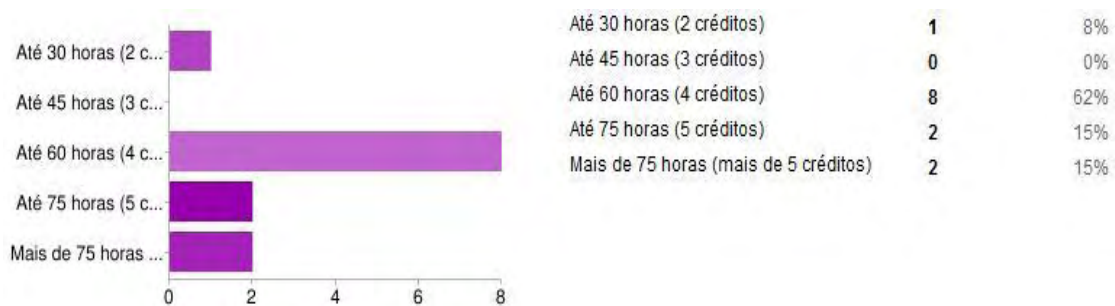
11. Você julga a carga horária da(s) disciplinas de Engenharia de Software suficiente para que se trabalhe conteúdos necessários ?

- Sim – Não



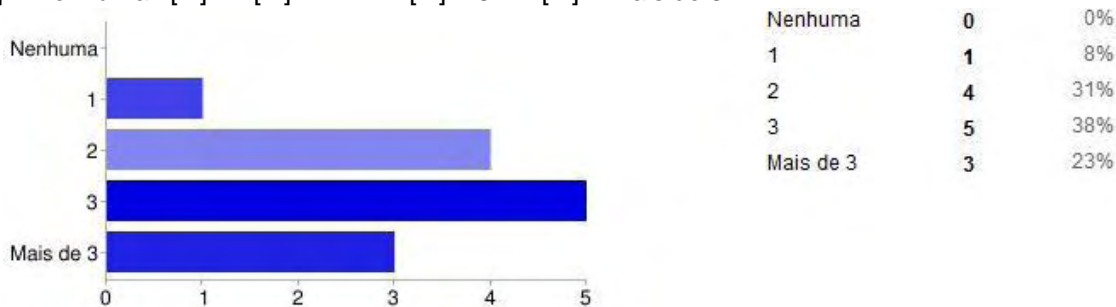
12. A(s) disciplina(s) ensinada(s) na área de Engenharia de Software tinha(m), em média, quantas horas semestrais?

- Até 30 horas (2 créditos)
- Até 45 horas (3 créditos)
- Até 60 horas (4 créditos)
- Até 75 horas (5 créditos)
- Mais de 75 horas (mais de 5 créditos)



13. Quantas disciplinas da área de Engenharia de Software tem no curso no qual você ensina ?

- Nenhuma
- 1
- 2
- 3
- Mais de 3



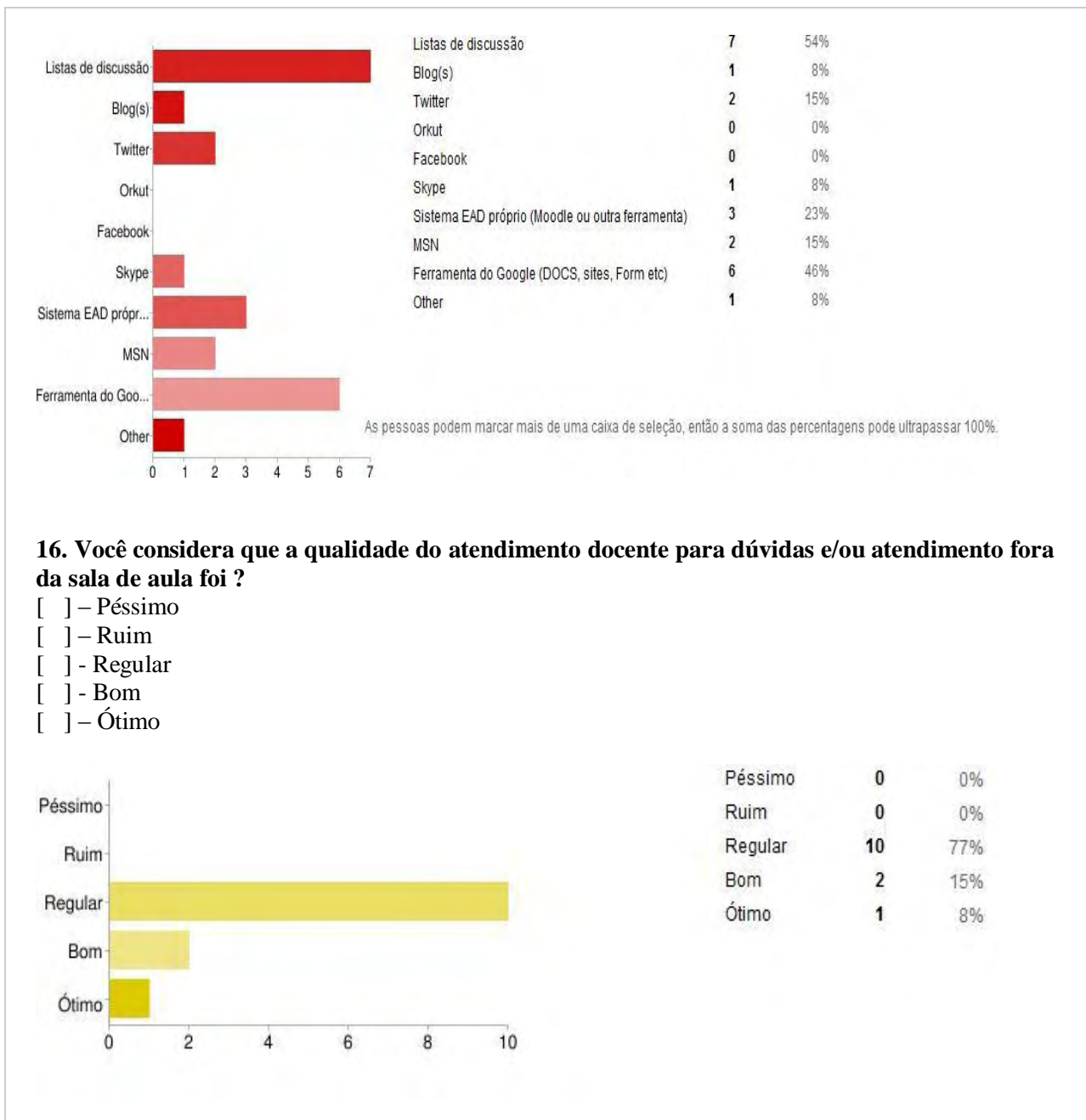
14. Você, ao longo da disciplina, usou quais ferramentas didáticas ?

- Aulas expositivas
- Aulas práticas (laboratório)
- Projeto(s) ao longo da disciplina
- Ensino baseado em problemas
- Aulas exclusivamente na modalidade à distância (EAD)
- Aulas complementares à distância (EAD)
- Seminários apresentados pelos alunos sobre assunto(s) da disciplina
- Palestras com profissionais experientes na área de Engenharia de Software
- Estudo de Casos
- Elaboração de Artigo(s) Científico(s)



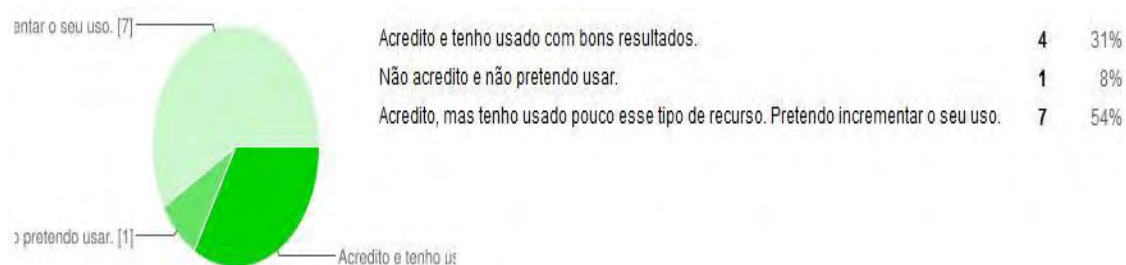
15. Você utilizou algum mecanismo de comunicação extra sala de aula com os alunos ?

- Listas de Discussão
- Blog(s)
- Twitter
- Orkut
- Facebook
- Skype
- Sistema de EAD próprio (Moodle ou outra ferramenta)
- MSN
- Ferramentas Google (DOCS, sites, Formetc)



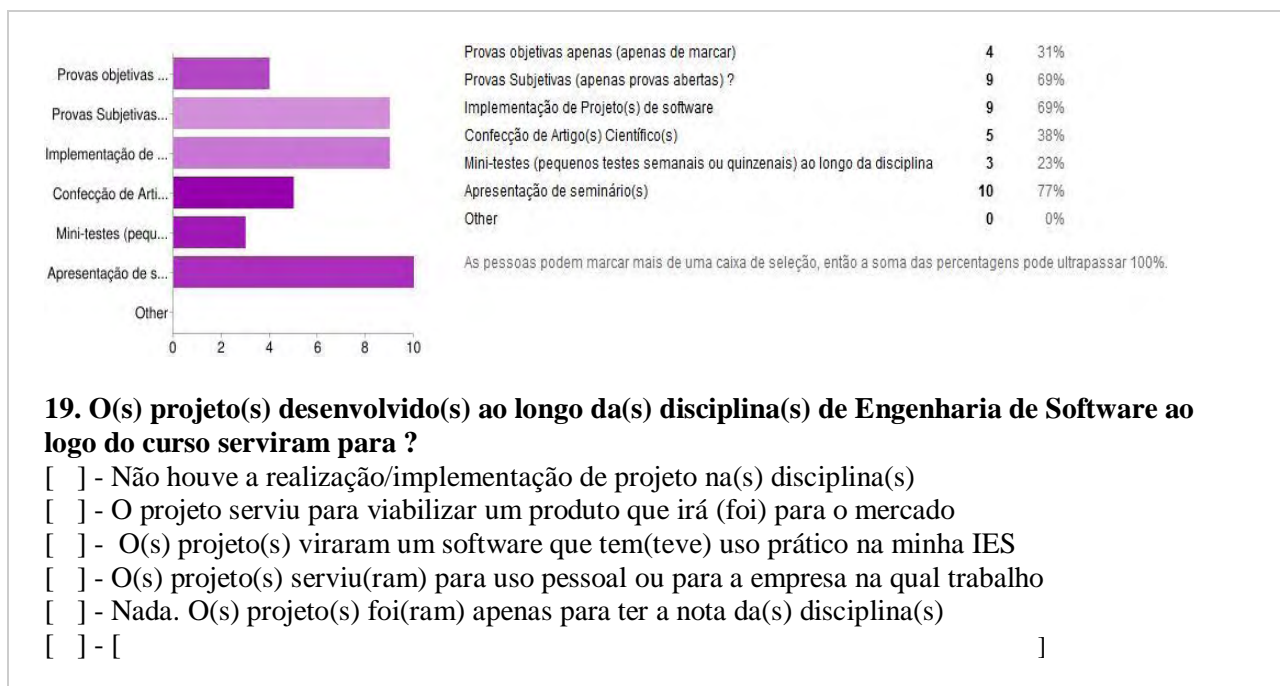
17. Você acredita que mecanismos de comunicação via Web podem ajudar na melhoria do processo educacional ?

- Acredito e tenho usado com bons resultados
- Não acredito e não pretendo usar
- Acredito, mas tenho usado pouco esse tipo de recurso. Pretendo incrementar o seu uso



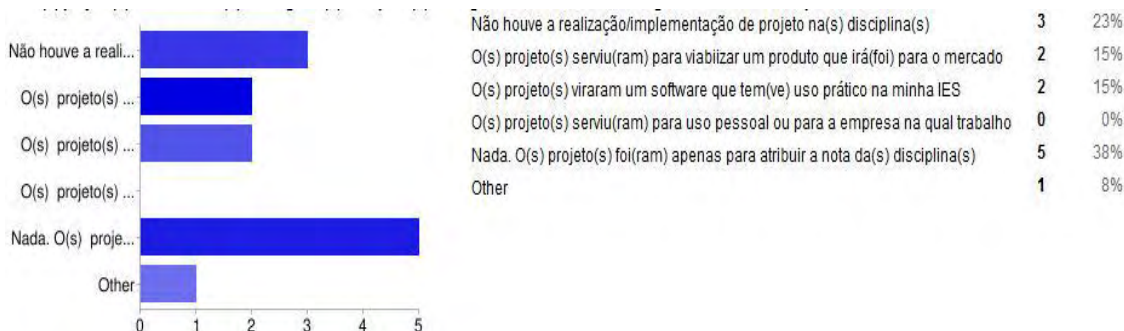
18. Qual(uais) mecanismos de avaliação foi(ram) utilizados para avaliar o(s) aluno(s) ao longo da(s) disciplina(s) ?

- Provas objetivas apenas (apenas de marcar)
- Provas Subjetivas (apenas provas abertas)
- Implementação de Projeto(s) de software
- Confecção de Artigo(s) Científico(s)
- Mini-testes (pequenos testes semanais ou quinzenais) ao longo da disciplina
- Apresentação de Seminário(s)
- []



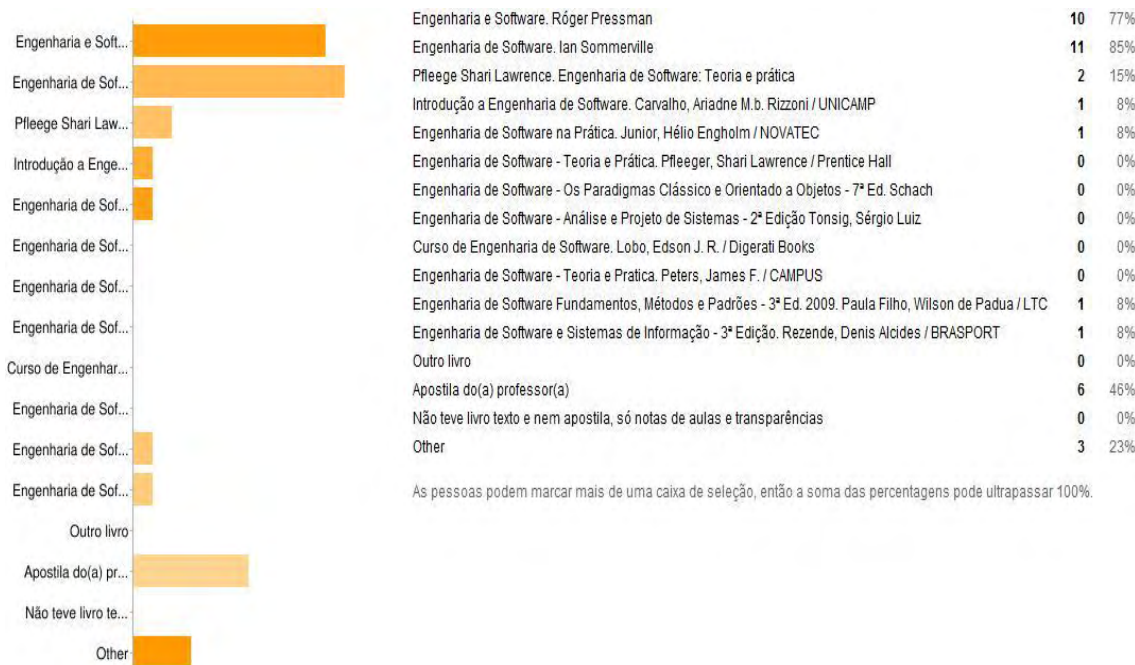
19. O(s) projeto(s) desenvolvido(s) ao longo da(s) disciplina(s) de Engenharia de Software ao longo do curso serviram para ?

- Não houve a realização/implementação de projeto na(s) disciplina(s)
- O projeto serviu para viabilizar um produto que irá (foi) para o mercado
- O(s) projeto(s) viraram um software que tem(teve) uso prático na minha IES
- O(s) projeto(s) serviu(ram) para uso pessoal ou para a empresa na qual trabalho
- Nada. O(s) projeto(s) foi(ram) apenas para ter a nota da(s) disciplina(s)
- []



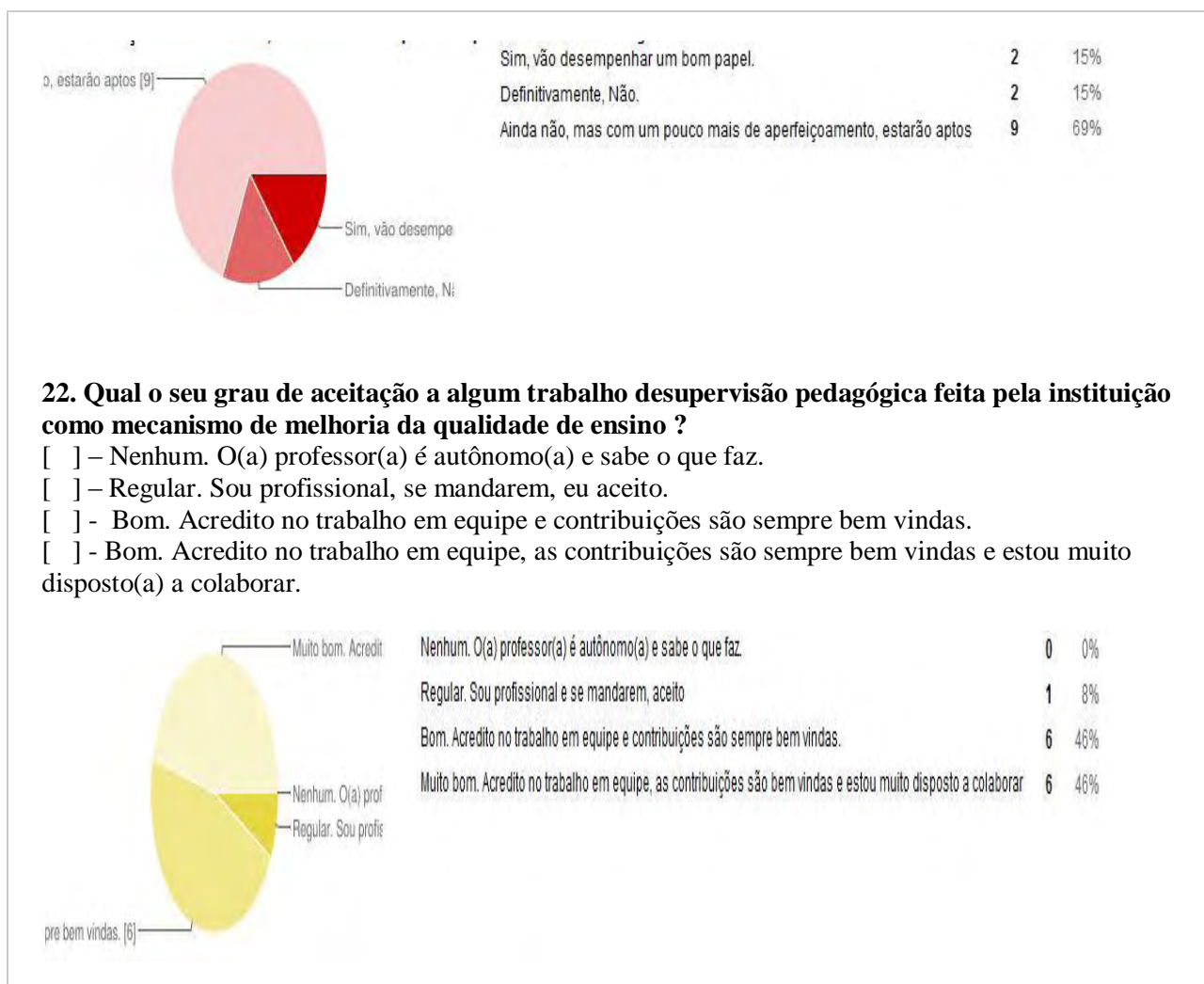
20. Qual foi o(s) livro(s) textos da(s) disciplina(s) ?

- Engenharia e Software. Róger Pressman
- Engenharia de Software. Ian Sommerville
- Pfllege Shari Lawrence. Engenharia de Software: Teoria e prática
- Introdução a Engenharia de Software Carvalho, Ariadne M.b. Rizzoni / UNICAMP
- Engenharia de Software na Prática Junior, Hélio Engholm / NOVATEC
- Engenharia de Software - Teoria e Prática Pflleeger, Shari Lawrence / Prentice Hall
- Engenharia de Software - Os Paradigmas Clássico e Orientado a Objetos - 7ª Ed.
- Engenharia de Software-Análise e Projeto de Sistemas. 2. Ed. Tonsig, Sérgio Luiz
- Curso de Engenharia de Software. Lobo, Edson J. R. / Digerati Books
- Engenharia de Software - Teoria e Pratica. Peters, James F. / CAMPUS
- Engenharia de Software Fundamentos, Métodos e Padrões - 3ª Ed. 2009 Filho, Wilson de Padua Paula / LTC
- Engenharia de Software e Sistemas de Informação - 3ª Edição Rezende, Denis Alcides / BRASPORT
- Outro livro []
- Apostila do(a) professor(a)
- Não teve livro texto e nem apostila, só notas de aulas e transparências



21. Com relação aos seus alunos, você os considera prontos / aptos a atuarem como engenheiros de software ?

- Sim, vão desempenhar um bom papel.
- Definitivamente, não.
- Ainda não, mas com um pouco mais de aperfeiçoamento, estarão aptos.



22. Qual o seu grau de aceitação a algum trabalho desupervisão pedagógica feita pela instituição como mecanismo de melhoria da qualidade de ensino ?

- Nenhum. O(a) professor(a) é autônomo(a) e sabe o que faz.
- Regular. Sou profissional, se mandarem, eu aceito.
- Bom. Acredito no trabalho em equipe e contribuições são sempre bem vindas.
- Bom. Acredito no trabalho em equipe, as contribuições são sempre bem vindas e estou muito disposto(a) a colaborar.

