

USO DE MAPAS MENTAIS NO ENSINO DE ENGENHARIA DE SOFTWARE

Klaus Fabian Côco – klausfabian@univix.edu.br / klaus@ele.ufes.br
Faculdade Brasileira – UNIVIX – Coord. de Eng. Elétrica e Eng. de Computação
Rua José Alves, 301, Goiabeiras.
29075-080 – Vitória – ES
Universidade Federal do Espírito Santo – UFES – Departamento de Eng. Elétrica
Av. Fernando Ferrari, s/n, Goiabeiras.
29060-900 – Vitória – ES
Fábio Almeida C6 – fabioco@ifes.edu.br
Instituto Federal do Espírito Santo - IFES – Departamento de Matemática do Campus Vitória
Av. Vitória, 1729 – Jucutuquara – Vitória – ES – 29040-780
Dalva Helena Lavagnoli – dhlavagnoli@gmail.com
Instituto Federal do Espírito Santo - IFES – Departamento de Matemática do Campus Vitória
Av. Vitória, 1729 – Jucutuquara – Vitória – ES – 29040-780

Resumo: *Este artigo apresenta um estudo entre diferentes abordagens de ensino de Engenharia de Software com o intuito de avaliar o aprendizado dos conteúdos ministrados no curso de Engenharia Elétrica – ênfase em computação – pelos alunos da Faculdade Brasileira (UNIVIX). O presente trabalho é resultado de um estudo realizado entre 2006 e 2010, com cinco turmas de alunos e aulas ministradas pelo mesmo professor adotando diferentes abordagens metodológicas. Assim, o presente trabalho encoraja a resultados favoráveis quando se faz uso de ferramentas cognitivas, mais especificamente os mapas mentais, para a educação de engenheiros nos conteúdos das disciplinas em que não predominam os formalismos e as deduções matemáticas que o estudante de engenharia está acostumado a lidar.*

Palavras-chave: *Metodologia, Engenharia de Software, Mapas Mentais.*

1 INTRODUÇÃO

A busca por técnicas e formas de melhorar a qualidade do mecanismo de ensino e aprendizagem em engenharia é tema recorrente nos meios acadêmicos de ensino (MATAI & MATAI, 2009), (SANTOVITO, 2009), (TOZZI, 2009). Propostas que abordam a educação nas ciências básicas, necessárias ao aprendizado em engenharia, são apresentadas há bastante tempo, todavia, as ações em busca da melhoria da qualidade do ensino desses conteúdos sempre estão em voga, mesmo que o conhecimento, nestas ciências, já tenha sido instituído a centenas de anos (SAAVEDRA & BEZERRA JR, 2009), (SARUBBI & SOARES, 2009), (VAZ *et al.*, 2009).

Se por um lado os estudos de técnicas para abordar a educação das ciências básicas vêm sendo continuamente aprimorado, o que se pode dizer dos conteúdos de engenharia originados há poucas décadas? A Engenharia de *Software*, por exemplo, têm seu marco de nascimento em meados da década de setenta, como um modelo para resolver o problema da “crise do *software*” (PRESSMAN, 2006). Todavia, as normas e procedimentos para a regulamentação das técnicas de construção de *softwares* somente começaram a surgir em meados da década de noventa, ou seja, a pouco mais de dez anos (TAIT & DELARIZZA, 2011).

Se observados os aspectos evolutivos do *software* nesses últimos dez anos, com o crescente desenvolvimento da internet e das arquiteturas multi-núcleo dos processadores, a engenharia de *software* esbarra em uma crescente demanda de técnicas e de conhecimentos ainda muito insipientes que tornam o ensino desta disciplina um desafio bastante inovador aos docentes.

Se por um lado as teorias de engenharia de *software* são relativamente recentes, por outro lado o problema é que uma aula representa a epistemologia de um professor, isso significa dizer que a concepção do professor em relação ao que é o conhecimento determinará seu processo de ensino (MORETTO, 2005), portanto, é normal que os professores tendam a gerenciar os processos de ensino-aprendizagem utilizando os seus próprios paradigmas, visto que, estes foram construídos com base em suas próprias experiências de sucesso, entretanto, infelizmente, o sucesso cognitivo de uma pessoa pode representar o fracasso de outra (FELDER & SILVERMAN, 1988).

Neste contexto, o presente trabalho busca comparar técnicas de ensino visando à melhoria da aprendizagem dos conteúdos de engenharia de *software* avaliando os resultados através do desempenho acadêmico ao longo da disciplina.

O estudo confronta diferentes formas de lecionar os conteúdos da disciplina e busca avaliar qual seria o melhor método de aprendizagem, mantendo os mesmos critérios avaliativos ao longo dos anos do experimento.

2 REFERENCIAL TEÓRICO

A Engenharia de *Software*, segundo o IEEE consiste na “*aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do software, isto é, a aplicação da engenharia ao software*” (PRESSMAN, 2006 p.17).

Esse processo de aplicar a engenharia ao *software* consiste no aprendizado de três áreas básicas: o processo de elaboração do *software*, ou seja, nos estudo dos modelos prescritivos de sistemas; a modelagem e projeto, que aborda a representação dos sistemas em linguagens de descrição de *software*, como a UML (*Unified Modeling Language*) (GUEDES, 2006); e a gestão de projetos de *software*, com a administração e a gerência de planejamento, desenvolvimento e execução do projeto de *software*.

Os conteúdos são teóricos com exposição de conceitos e representações gráficas em diagramas, sem a necessidade de cálculos matemáticos, fórmulas ou procedimentos elaborados, mas exige do aluno a interconexão desses conceitos, das relações que descrevem o fluxo das informações manipuladas pelo *software* e de raciocínio lógico construtivo baseado no entendimento da linguagem que o computador usa para escrita de seus algoritmos.

Diferente do aluno das ciências humanas, que moldam seu raciocínio baseado na leitura de textos, é comum o aluno de engenharia ter dificuldade no aprendizado da engenharia de *software*, pois não está acostumado a trabalhar essas habilidades, porque pauta seu aprendizado predominantemente no raciocínio lógico matemático.

2.1 Mapas Mentais

Normalmente, quando o indivíduo é submetido a algo novo ele faz uso de basicamente duas premissas básicas: comparação com aquilo que já se conhece (abordagem estrutural) e, então, procede de forma a armazenar (memorização) o novo conhecimento ou tentar simplificar o novo conteúdo de forma a se encaixar em outro já pré-existente (abordagem redutora) (HERMANN, 2011). Porém, se o novo conhecimento não corresponder a nenhum padrão conhecido após a abordagem estrutural ou redutora, ele geralmente será desprezado ou esquecido (HERMANN, 2011).

Assim, do ponto de vista do educador, o aprendizado é mais fácil se for resgatado no aprendiz suas referências e conhecimentos anteriores, o que torna o novo algo naturalmente adaptável.

Nesse contexto, valorizam-se as concepções prévias dos alunos, tornando-as ponto de partida para aprendizagem de novas concepções (concepções acadêmicas), ou seja, aproveita-se a estrutura cognitiva do alunado, formada pelo conjunto de suas idéias, suas experiências e seus paradigmas ligados ao senso comum para propor as concepções acadêmicas, facilitando a justaposição entre as duas concepções (MORETTO, 2003).

Então, pode-se dizer que a habilidade de ensinar outras pessoas de maneira eficaz depende do hábito de aprender de forma organizada.

A organização na aprendizagem inclui, biologicamente, a integração de diferentes partes do cérebro em prol da absorção do conhecimento apresentado. Nessa linha, diferentes técnicas de auxílio à aprendizagem passaram a compor o rol de ferramentas de estudo, como: a programação neurolingüística (SEYMOUR & O'CONNOR, 1995), a sugestologia de Lazanov (BELANGUER, 1985), o programa de enriquecimento instrumental de Feuerstein (RUBINSTEIN, 2011), a rede semântica (HARTLEY & BARNDEN, 1997) e os mapas mentais (BUZAN, 1996). Dessas ferramentas, o Mapa Mental é uma das mais simples e de fácil aprendizagem.

A técnica de construção de mapas mentais foi criada pelo inglês Tony Buzan após observar que os alunos que faziam uso de estratégias de trabalho e de anotações diferenciadas, com cores, desenhos, símbolos e ilustrações obtinham melhores resultados de aprendizagem que os alunos que não usavam tais métodos, ou seja, a exploração dos hemisférios direito e esquerdo do cérebro no processo de aprendizagem proporcionava melhor absorção do conhecimento passado pelo educador (HERMANN, 2005).

Tal observação de Buzan coincide com o Construtivismo Piagetiano que sustenta que os seres humanos são capazes de criar conhecimentos tanto mais sofisticados, quanto melhor forem as suas interações com o mundo. Portanto, sair de um estado de menor conhecimento a um conhecimento superior, depende da qualidade dessas interações, qualidade essa que por sua vez, depende de estratégias pedagógicas adequadas e da forma como são conduzidas (CÓ, 2007).

A construção de um mapa mental, como proposto por Buzan, consiste no encadeamento hierarquizado das informações de maneira não linear com formatação gráfica, colorida e contendo ilustrações que auxiliam na memorização e no aprendizado dos conteúdos abordados (BUZAN, 1996). A Figura 1 mostra um exemplo de mapa mental utilizado nas aulas do curso. O programa usado para a construção dos mapas é um *software* livre denominado *freemind* (FREEMIND, 2011).

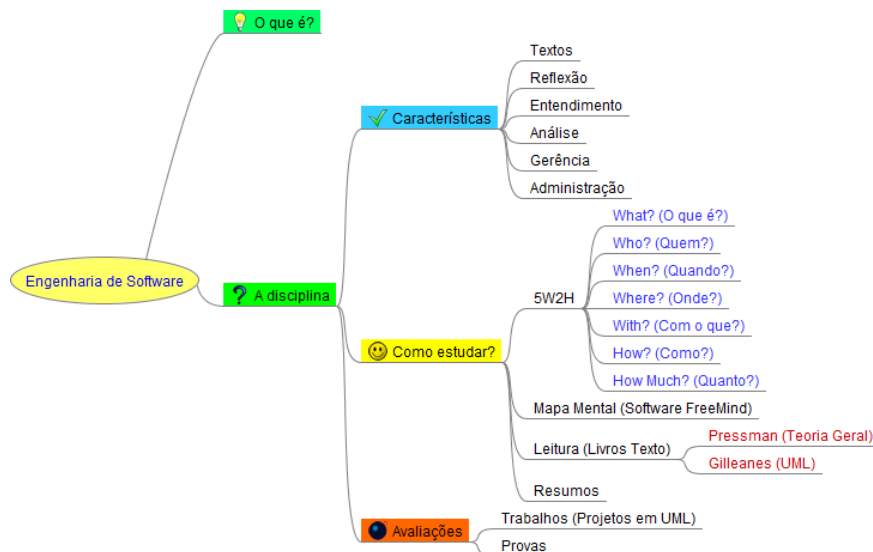


Figura 1 – Um dos mapas mentais utilizados nas aulas do curso.

Pode-se observar pela Figura 1, que a utilização de mapas mentais, faz com que a aprendizagem ganhe uma nova conotação, passando da aquisição isolada de informações para o estabelecimento de relações entre informações, ganhando significado cognitivo, lançando o conceito de aprendizagem significativa (MORETTO, 2003).

A inteligência, para Piaget, é o processo de adaptação do organismo às novas situações e, como tal, implica a construção contínua de novos esquemas mentais. Como essa adaptação diz respeito ao mundo exterior, quanto mais complexo e organizado o estímulo oferecido pelo meio e, conseqüentemente, quanto mais complexa e organizada for a sua interação com o meio, mais “inteligente” será o indivíduo (CÓ, 2007).

3 METODOLOGIA

O estudo foi realizado em cinco turmas de alunos do curso de engenharia elétrica com ênfase em computação no período de 2006 a 2010. O curso de engenharia de *software*, com duração de 40 horas semestral, foi ministrado por um mesmo professor, autor deste artigo, seguindo diferentes abordagens de ensino, mas mantendo os mesmos critérios para a verificação de aprendizagem.

Para a turma de 2006 o procedimento de aulas consistia na exposição dos conceitos usando mapas mentais com comentários, exemplos e explicações realizadas verbalmente pelo professor, com uso de projetor multimídia para apresentação dos diagramas, esquemas, figuras e outros conteúdos da disciplina.

Para a turma de 2007 o procedimento de aulas consistia no modelo de 2006, porém, os alunos compunham uma grande equipe, subdividida em grupos com tarefas específicas, coordenadores de grupos e um coordenador de toda a equipe. Os alunos tinham que efetuar um projeto de *software* único a ser entregue no final do curso, cuja pontuação apenas seria utilizada para computar 30% da nota do segundo bimestre. Eram necessários conhecimentos de gerência e de UML para a elaboração do projeto.

Para a turma de 2008 o procedimento de aulas consistia na exposição dos conceitos de forma sequencial com comentários, exemplos e explicações realizadas verbalmente pelo professor, com uso de projetor multimídia para apresentação de textos e conteúdos da disciplina. O projeto de *software* seguiu os mesmos moldes de 2007. A Figura 2 mostra um

exemplo das notas de aulas apresentadas aos alunos utilizando o *software Power Point* da Microsoft.



Figura 2 – Modelo de nota de aula com exposição dos conceitos de forma sequencial.

Para a turma de 2009 o procedimento foi análogo ao de 2008, porém não houve projeto de *software*. Para a turma de 2010 o procedimento foi idêntico ao da turma de 2007.

As avaliações consistiam de provas com questões de múltipla escolha, discursivas conceituais e de marcação verdadeira ou falsa. Foram utilizadas questões produzidas pelo professor e outras de concursos públicos, procurando manter um equilíbrio entre questões de maior dificuldade de solução e de menor dificuldade em todas as avaliações durante todos os anos.

Os trabalhos buscavam modelar um sistema utilizando a linguagem UML e envolviam o conhecimento e uso de vários diagramas dessa linguagem e conhecimentos de gestão de projetos com cumprimento de cronograma e tarefas divididas entre grupos de trabalho. Os grupos eram organizados de forma que o trabalho realizado pela equipe A era necessário ao desenvolvimento dos trabalhos da equipe B e assim sucessivamente. Como o processo de desenvolvimento de *software* não é sequencial, problemas de modelagem voltavam às equipes anteriores para correções e ajustes de projeto, comprometendo o cronograma e a qualidade dos resultados.

Para a análise das técnicas de ensino as notas dos trabalhos foram desconsideradas e a parcela e as notas bimestrais e finais apresentadas neste trabalho foram ajustadas pelas notas das provas.

4 RESULTADOS E ANÁLISES

Os resultados da média de notas semestrais da turma e da média dos coeficientes de rendimento da turma são apresentados na Tabela 1.

Tabela 1 – Média das notas na disciplina e média dos coeficientes de rendimento dos alunos no período 2006-2010.

Período	Média das Notas (<i>mn</i>)	Média do Coeficiente de Rendimento (<i>mcr</i>)	Total de Alunos
2006	7,3	7,6	20
2007	7,5	7,5	21
2008	7,0	7,5	27
2009	5,7	7,1	22
2010	6,8	7,1	21

Observa-se que em todos os casos a média de notas da turma na disciplina é no máximo igual à média do coeficiente de rendimento da turma. Isso se deve à natureza da disciplina onde há a predominância dos dados textuais, conceitos e esquemas lógicos de construção de *software* em consoante dicotomia com o predomínio lógico matemático das demais disciplinas. Assim, o aluno apresenta, naturalmente, uma dificuldade em assimilar o conhecimento inerente à disciplina de engenharia de *software*.

Por outro lado, com o intuito de comparar o aproveitamento da aprendizagem frente aos métodos usados nas aulas, as notas das turmas foram normalizadas pelos respectivos valores de coeficientes de rendimento, de forma que as médias das turmas corresponderiam ao valor 7,0 de média dos coeficientes de rendimento, como mostra a Equação (1).

$$mno = \frac{mn \times 7,0}{mcr} \quad (1)$$

A Tabela 2 apresenta o valor das médias normalizadas das notas das turmas e a variação percentual das notas em relação à média de aprovação (Nota = 7,0).

Tabela 2 – Média das notas normalizadas, resumo das ações no período 2006-2010 e percentual de variação das notas normalizadas em relação à nota 7.

Período	Média Normalizada (<i>mno</i>)	Ações	Variação (%)
2006	6,7	Mapa Mental	-4,3
2007	7,0	Mapa Mental + Projeto	0
2008	6,5	Sequencial + Projeto	-7,1
2009	5,6	Sequencial	-20,0
2010	6,7	Mapa Mental + Projeto	-4,3

Pode-se observar, pelo resultado, que nas turmas em que o professor fez uso de mapas mentais no ensino da engenharia de *software*, as notas das avaliações foram superiores às notas das turmas em que nas aulas houve a apresentação dos conteúdos de forma sequencial. A variação das notas em relação à nota 7,0 é sempre maior e negativa nas turmas em que o método de ensino foi sequencial.

De maneira geral, observa-se que o projeto de *software* contribui para o aprendizado dos conteúdos ministrados uma vez que as notas, quando confrontadas as turmas em que não houve projeto, foram no mínimo iguais ou superiores àquelas.

Outra constatação foi feita em relação ao número de aprovações, reprovações e desistências. A Tabela 3 mostra os dados referentes às turmas nos cinco anos de acompanhamento.

Tabela 3 – Percentual de alunos em relação à média semestral.

Período	Percentual de Alunos e Notas			
	Aprovados		Reprovados	Desistentes
	< 7,0	≥ 7,0		
2006	0,0	100,0	0,0	0,0
2007	14,3	76,2	0,0	9,5
2008	22,2	59,3	3,7	14,8
2009	13,6	36,4	40,9	9,1
2010	47,6	47,6	0,0	4,8

Na turma de 2006, onde foi abordada a metodologia de uso dos mapas mentais sem projeto, o índice de aprovação foi de 100% sem desistências. Ao que parece, o método manteve o alunado interessado no aprendizado e com uma boa absorção dos conhecimentos passados pelo professor.

Em 2007, apesar de ter sido a maior média normalizada das turmas, 9,5% dos alunos desistiram da disciplina, muito provavelmente devido ao projeto de *software*, agora integrado ao conteúdo avaliativo. Por outro lado, 90,5% dos alunos foram aprovados, entretanto 14,3% não obtiveram aprovação direta na disciplina, tendo então que se submeter a uma prova final, cuja nota é calculada conforme a Equação (2)

$$MF = 0,6MS + 0,4PF \quad (2)$$

onde: MF é a Média Final, cuja aprovação se dá quando esta nota é maior ou igual a 5,0; MS é a Média Semestral, obtida pelas avaliações ao longo do semestre; e PF é a nota da Prova Final.

A turma de 2008 teve o maior número de desistentes, uma vez que incorporou o método de ensino sequencial ao projeto de *software* e apresentou um índice de aprovação de 81,5% e de reprovação de 3,7%, o que não ocorreu nos anos anteriores.

A turma de 2009 teve o pior desempenho com apenas 50% de aprovações, com somente 36,6% da turma obtendo aprovação direta sem prova final. Este resultado mostra que o uso da abordagem sequencial, sem projeto de *software*, não permitiu o aprendizado suficiente à aprovação. As aulas não proporcionavam o encadeamento dos conceitos e a organização estruturada, como fazem os mapas mentais. A ausência de projeto de *software* não instigou aos alunos ao estudo e à aplicação dos conceitos repassados em sala de aula.

Na turma de 2010, foi retomado o uso dos mapas mentais e do projeto de *software* e obteve-se 95,2% de aprovação, metade com prova final e 4,8% de desistentes, porém sem reprovações. Boa parte da turma fora composta pelos alunos reprovados no ano anterior, inclusive os alunos desistentes foram também desistentes no ano anterior e se propuseram a abandonar o curso já no início das aulas, talvez por achar que a nova metodologia de aulas, com o uso de mapas mentais, fosse mais uma agravante para o aprendizado.

5 CONCLUSÕES

Existem evidências de que o uso de mapas mentais, pelo professor, nas aulas de engenharia de *software* contribui de maneira eficiente no aprendizado e na continuidade no curso dos alunos dessa disciplina, cujos conteúdos são predominantemente conceituais ou de exposição textual sem fórmulas ou gráficos matemáticos.

Existem evidências de que o uso de projeto de *software*, atrelado aos ensinamentos dos conteúdos da disciplina de engenharia de *software*, aprimoram o fator de aprendizagem dos alunos, resultando em avaliações com notas maiores que nas turmas em que os projetos não foram solicitados. Todavia, a abordagem de projeto pode tornar-se um agravante e estimular a desistência do curso, caso os alunos não tenham um bom acompanhamento dos conteúdos ministrados pelo professor.

Portanto, o presente trabalho encoraja a resultados favoráveis quando se faz uso de mapas mentais, para a educação de engenheiros nos conteúdos da disciplina de engenharia de *software* em que não predominam os formalismos e as deduções matemáticas que o estudante de engenharia está acostumado a lidar.

REFERÊNCIAS BIBLIOGRÁFICAS

BUZAN, Tony . Saber pensar. 1. ed. Lisboa: Presença II, 1996. 160 p.

BELANGUER, Bagrina. La sugestologia. 1. ed. Espanã: Mensajero, 1985. 256 p.

CÓ, F. A. A aplicação de uma estratégia lúdica de ensino-aprendizagem para garantir o desenvolvimento simultâneo dos pensamentos enxuto e sustentável na construção civil. 2007. Tese (Doutorado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil, Universidade Federal Fluminense, Niterói – RJ.

FREEMIND. *Free mind mapping software*. Disponível em: <http://freemind.sourceforge.net/wiki/index.php/Main_Page> Acesso em 02 maio 2011.

FELDER, R. M., & SILVERMAN, L.K. *Learning and teaching styles in engineering education*. Engineering Education, 78(7), 674-681.

GUEDES, Gilleanes T. A. UML: uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2006. 319 p.

HARTLEY, Roger T.; BARNDEN, John A. *Semantic networks: visualizations of knowledge*. Trends in Cognitive Sciences, Volume 1, Issue 5, August 1997, Pages 169-175. (doi:10.1016/j.physletb.2003.10.071)

HERMANN, Walther; BOVO, Viviani. Mapas mentais – enriquecendo inteligências. 1. ed. São Paulo: Walther Hermann, 2005. 372 p.

HERMANN, Walther. **Saber aprender – uma visão panorâmica**. Disponível em: <<http://www.idph.net/artigos/idph/saberaprender.shtml>> Acesso em: 21 abr. 2011.

MATAI, Patricia H. L. dos Santos; MATAI, Shigueharu. A reengenharia para se tornar um professor. **Anais: XXXVII – Congresso Brasileiro de Ensino de Engenharia**. Recife: UFPE, 2009.

MORETTO, V. P. **Construtivismo: a produção do conhecimento em aula**. Rio de Janeiro: DP&A editora, 2003.

MORETTO, V. P. Prova: um momento privilegiado de estudo não um acerto de contas. Rio de Janeiro: DP&A editora, 2005.

PRESSMAN, Roger S. Engenharia de software. 6. ed. São Paulo: McGraw-Hill, 2006. 720 p.

RUBINSTEIN, Edith. **PEI – programa de enriquecimento instrumental**. Disponível em: <<http://www.psicopedagogia.com.br/entrevistas/entrevista.asp?entrID=5>> Acesso em: 21 abr. 2011.

SAAVEDRA, Nestor; BEZERRA JR, Arandi G. Novas experiências e inovações no ensino de leis de conservação e simetrias em disciplinas de física para cursos de engenharia. **Anais: XXXVII – Congresso Brasileiro de Ensino de Engenharia**. Recife: UFPE, 2009.

SANTOVITO, Rogério F. A promoção da aprendizagem significativa na educação continuada em engenharia: o uso de mapas conceituais e a formação de subsunçores. **Anais:** XXXVII – Congresso Brasileiro de Ensino de Engenharia. Recife: UFPE, 2009.

SARUBBI, Pedro A.; SOARES, Flávia. Investigando dificuldades de alunos de cálculo em problemas de taxas relacionadas. **Anais:** XXXVII – Congresso Brasileiro de Ensino de Engenharia. Recife: UFPE, 2009.

SEYMOUR, John; O'CONNOR, Joseph. Introdução a programação neurolingüística. 6. ed. São Paulo: Summus, 1995. 232 p.

TAIT, Tania F. C.; DELARIZZA, João Carlos. **Aplicação das normas ISO no desenvolvimento de software: o caso de uma empresa de componentes eletrônicos.** Disponível em: < http://www.abepro.org.br/biblioteca/ENEGEP1998_ART418.pdf> Acesso em: 21 abr. 2011.

TOZZI, Marcos J. Perfil do professor de engenharia. **Anais:** XXXVII – Congresso Brasileiro de Ensino de Engenharia. Recife: UFPE, 2009.

VAZ, Ieda do C.; COURA SOBRINHO, Jerônimo; LAUDARES, João B. A prática educativa de professores de matemática e de disciplinas específicas em cursos de engenharia. **Anais:** XXXVII – Congresso Brasileiro de Ensino de Engenharia. Recife: UFPE, 2009.

USE OF MIND MAPS IN TEACHING SOFTWARE ENGINEERING

Abstract: *This article presents a study of different approaches to teaching software engineering in order to assess the learning of content taught in the Electrical Engineering course - emphasis in computing - by the students of the Faculdade Brasileira (UNIVIX). This work is the result of a study conducted between 2006 and 2010, with five classes of students and classes taught by the same teacher adopting different methodological approaches. Thus, this work encourages the favorable results when it makes use of cognitive tools, specifically the mind maps for the education of engineers in the disciplines that are not prevalent in the formalisms and mathematical deductions that the engineering student is accustomed deal.*

Key-words: Methodology, Software Engineering, Mind Maps.