

Aprendizagem baseada em problema: a experiência da disciplina Sistemas de Representação do Conhecimento do doutorado em Difusão do Conhecimento

Claudia Pinto Pereira Sena¹, Elmara Pereira de Souza², Maria Teresinha Tamanini Andrade³

¹UEFS - Universidade Estadual de Feira de Santana
Av. Transnordestina, S/N Bairro Novo Horizonte
CEP 44.036-900 Feira de Santana – BA

caupinto.sena@gmail.com

²UFBA- Universidade Federal da Bahia
Av. Reitor Miguel Calmon s/n - Campus Canela
40.110 100 - Salvador – Bahia

elmarasouza@gmail.com

³IFBA-Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Campus Simões Filho Via Universitária, s/n. Pitanguinha.
CEP 43700-000 Simões Filho, BA

tamanini@ifba.edu.br

Resumo. *Este artigo descreve o processo, a análise, os resultados e o referencial teórico utilizado para a resolução do problema apresentado na disciplina Sistemas de Representação do Conhecimento do Doutorado multidisciplinar e multi-intitucional em Difusão do Conhecimento. Na disciplina, foi utilizada a estratégia educacional Aprendizagem Baseada em Problemas - PBL com o objetivo de favorecer a construção coletiva da implementação na Máquina de Turing, da Regra 90 e dos Autômatos Celulares elementares. O PBL se mostrou uma metodologia adequada, não apenas para resolução do problema, mas, também, para aquisição autônoma e auto dirigida dos conhecimentos necessários a esse fim.*

1. Introdução

No processo de ensino-aprendizagem são constantes os desafios para professores e alunos, em especial no que diz respeito à construção e difusão do conhecimento em que o aluno é sujeito ativo no processo. É crescente a necessidade, diante dos desafios postos pela sociedade da informação/conhecimento, de pessoas dispostas a conviver com o “novo”, reinventar soluções, argumentar, criar, trabalhar em grupo. Na educação,

várias estratégias e métodos têm sido experimentados com o objetivo de favorecer a formação de sujeitos autônomos, autores. Dentre esses métodos, encontra-se o *Problem Based Learning* – PBL (Aprendizagem Baseada em Problemas).

Este artigo apresenta o processo, a análise e os resultados das atividades da disciplina Sistemas de Representação do Conhecimento do Doutorado Multiinstitucional e Multidisciplinar em Difusão do Conhecimento que utilizou o PBL como proposta metodológica para favorecer a implementação na Máquina de Turing (MdT), da Regra 90 e dos Autômatos Celulares (AC) elementares.

Para cumprir a tarefa, se fez necessária a resolução do problema que desafiou a construção e compartilhamento de conhecimentos a partir de temas variados como: algoritmos, sistemas complexos, fractais, autômatos celulares e MdT. De outro lado, procedeu-se à correlação entre o problema e metodologia utilizada - PBL.

Os alunos interpretaram o trabalho como uma oportunidade de construção coletiva de conhecimento, pois o grupo era composto por pessoas com formações distintas (Pedagogia, Letras, Sistemas de Informação, Engenharia, Direito, entre outros) e com níveis de conhecimento significativamente diferentes.

O artigo está dividido em seis partes, incluindo a introdução. Na segunda seção apresenta-se a fundamentação teórica sobre a metodologia PBL; segue-se a descrição do problema; na seção quatro apresenta-se o referencial teórico utilizado para a resolução do problema: a MdT, AC e a regra 90; a resolução do problema é descrita na seção cinco e finaliza-se com as conclusões.

2. Aprendizagem baseada em problema

O Aprendizado Baseado em Problemas (*Problem-Based Learning* - PBL) surgiu no final dos anos 1960, na McMaster University Medical School, Canadá, e, em pouco tempo, difundiu-se pelo mundo, em várias universidades, principalmente, nas escolas de medicina (ALGARVE, s.d; MAMEDE e PENAFORTE, 2001). No Brasil, alguns exemplos podem ser observados como: Escola de Saúde Pública do Ceará (COELHO-FILHO; SOARES; SA, 1998), Universidade Federal de São Paulo (UNIFESP, s.d.), Universidade Estadual de Londrina, Universidade de São Paulo (SOARES, ARAUJO, s.d.) e na Universidade Estadual de Feira de Santana (LOULA, 2009; SANTOS et al., 2007; PEREIRA et al., 2005).

Esta estratégia educacional se destaca pelo uso contextualizado de uma situação problema para o aprendizado auto-dirigido centrado no estudante, promove o desenvolvimento da habilidade de trabalhar em grupo e também estimula o estudo individual, de acordo com os interesses e o ritmo de cada estudante (DELISLE, 1997). O aprendizado passa a ser centrado no aluno, que sai do papel de receptor passivo para o de agente e principal responsável pelo seu aprendizado. Os professores que atuam como tutores (ou facilitadores) nos grupos têm a oportunidade de conhecer bem os estudantes e de manter contato com eles durante todo o percurso.

Para a operacionalização do PBL, é indicado que o professor (tutor ou facilitador) reúna-se com os participantes do grupo tutorial periodicamente (pelo menos uma vez

por semana) em sessões tutoriais, durante o período de desenvolvimento da resolução do problema. O professor não "ensina" da maneira tradicional, mas facilita a discussão dos estudantes, conduzindo-a quando necessário e indicando os recursos didáticos úteis para cada situação.

Uma sessão tutorial inicial trabalha os conhecimentos prévios dos estudantes sobre o assunto apresentado; as ideias, os fatos, as hipóteses e as questões são primeiramente identificados e listados, e, em seguida, são formulados os objetivos de aprendizado, com base em tópicos considerados úteis para o esclarecimento e a resolução do problema, que é desenvolvida em sete passos (PEREIRA et al., 2005): 1. Ponto de partida: compreensão inicial do problema e esclarecimento dos termos não conhecidos por todos; 2. Tempestade de idéias: dar as possíveis explicações, de acordo com os conhecimentos prévios – "*Brain storm*"; 3. Sistematização: análise das diversas partes identificadas do problema a ser resolvido; 4. Formulação de questões: definição dos problemas a serem entendidos e explicados; 5. Metas de aprendizagem: formulação dos objetivos de aprendizado e estudo individual baseado nestes objetivos; 6. Avaliação de processos: socialização com o grupo; 7. Seguimento (SANTOS et al., 2007).

Em cada uma das sessões tutoriais, designa-se um coordenador/líder que deve orientar os trabalhos a serem desenvolvidos durante a sessão. É indicado que sempre seja escolhido um integrante do grupo para secretariar a sessão, registrando aspectos importantes do seu desenvolvimento, especialmente as ideias trazidas por cada um dos integrantes. Na etapa seguinte, os estudantes vão trabalhar independentemente, na busca de informações e na sua elaboração (estudo auto-dirigido) antes da próxima sessão tutorial, quando as informações trazidas por todos serão discutidas e integradas no contexto do caso problema.

O PBL, portanto, através de um problema apresentado ao grupo, busca motivar o aprendizado auto-dirigido e, potencialmente, focar na construção de conhecimento colaborativo e cooperativo. Ribeiro e Mizukami (2004) ressaltam que o PBL não almeja necessariamente a solução do problema inicialmente apresentado, e sim foca o processo de construção do conhecimento desenvolvido pelo grupo de forma integrada, valorizando a aprendizagem tanto autônoma quanto cooperativa, bem como a ética e o respeito ao conhecimento do outro, em torno de um problema da vida real.

3. O problema

O problema a seguir foi apresentado como motivação para a aplicação da estratégia educacional PBL na disciplina Sistemas de Representação do Conhecimento: “Num passeio vespertino por uma praia durante suas férias, um aluno de computação científica encontra uma concha com a padronagem da Figura 1. Neste instante ele associa sua concha a alguns dos autômatos celulares elementares que acabara de estudar em uma das disciplinas do curso. Imediatamente tem a idéia de implementar na MdT uma das regras (i.e. Regra 90) que, segundo suas suspeitas, é a que mais se aproxima com a padronagem observada. Passa todas as férias pensando nessa implementação. Que fazer para ajudar-lhe?”



Figura 1. Padronagem da concha apresentada no problema

Para resolver o problema, foi necessário programar e implementar, na MdT, a Regra 90 dos autômatos celulares elementares, para que pudesse ser gerado como resultado o desenho da concha, simulando o AC com a padronagem na forma de um fractal. As questões de aprendizado apresentadas foram: Como programar a MdT? Como definir uma regra de um autômato celular elementar?

O objetivo do problema era que o estudante se familiarizasse com os conceitos de algoritmo e lógica, com a prática da MdT e com as ideias iniciais dos autômatos celulares. Além disso, ao finalizar o problema, o estudante teria condições de estabelecer associações entre sistemas complexos e sua representação cognitiva.

Foram feitas três sessões tutoriais para que os alunos pudessem discutir, estudar, resolver o problema e, posteriormente, apresentar um relatório sobre: (1) a MdT (histórico, fundamentos matemáticos e aplicabilidade); (2) AC, além do algoritmo implementado e sua descrição detalhada.

4. Referencial Teórico utilizado para a resolução do problema

4.1. Máquina de Turing

A MdT é um dispositivo teórico, conhecido como máquina universal, que foi concebido pelo matemático Alan Turing em 1936, como um modelo matemático do processo de computação. Sua estrutura simples é proposital, uma vez que Turing pretendia, com sua definição, chegar a um modelo que fosse universalmente aceito (ACIÓLY e BEDREGAL, 2000).

A MdT opera através de um controle finito; de uma fita, que pode servir como dispositivo de entrada (inicialmente a cadeia de entrada está escrita na fita); de uma área de trabalho (memória, ou rascunho); de um dispositivo de saída (possíveis resultados estarão escritos na fita ao final da computação). Consiste de uma fita de comprimento infinito acoplada a uma unidade de controle, que lê/grava na fita a partir de uma cabeça de leitura/gravação. Em sua essência, toda MdT move-se ou move símbolos, de uma posição para outra em uma fita. A fita é dividida em n quadros; cada quadro pode ser preenchido com um símbolo simples de um alfabeto finito ou pode estar em branco.

Segundo Acióly e Bedregal (2000), a máquina de Turing pode ser visualizada como um vetor unidimensional de células (Figura 2), cada uma das quais pode manter um único símbolo. O vetor se estende indefinidamente para a direita e para a esquerda e pode

manter informações ilimitadas. A transição de estados na MdT é ilustrada na Figura 3 através dos estados q_0 e q_1 que mostram a leitura e escrita na fita.

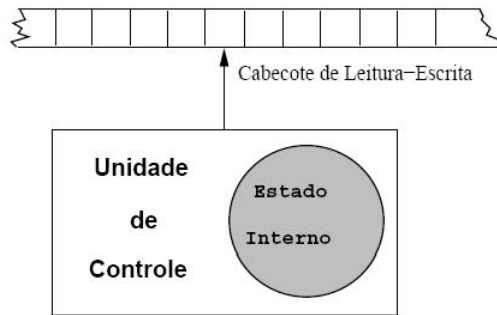


Figura 2. Representação esquemática da máquina de Turing
 Fonte: Acióly, Berdregal, 2000

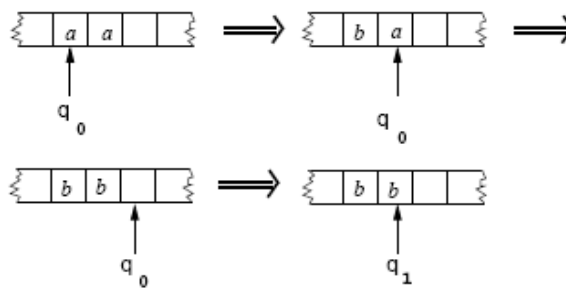


Figura 3. Movimento de transição na MdT
 Fonte: Acióly, Bendregal, 2000, p. 141.

4.1.1. Formalização da Máquina de Turing

A MdT pode ser definida com uma ou mais fitas. Na implementação com uma fita, tratada nesse artigo para resolver o problema proposto, a máquina é formalizada por uma Tupla do tipo $M = (Q, \Sigma, \Gamma, s, b, F, \delta)$, onde Q é um conjunto finito de estados; Σ é um alfabeto finito de símbolos; Γ é o alfabeto da fita (conjunto finito de símbolos); s é o estado inicial; b é o símbolo branco (o único símbolo que se permite ocorrer na fita infinitamente em qualquer passo durante a computação); F é o conjunto dos estados finais e δ é a função de transição (E é o movimento para a esquerda e D é o movimento para a direita).

4.2. Autômatos celulares

Autômatos Celulares (AC) são utilizados como modelo para a determinação e/ou formação de padrões que tentam ou explicam fenômenos da natureza, bem como comportamentos sociais, ambientais e biológicos, assim como fatores que influenciam em suas regras de inter-relação/interação. Entre esses modelos, pode-se citar o modelo de autômato celular proposto por John Von Neumann, nos anos 1950, que objetivava a simulação de auto-reprodução de organismos biológicos e o modelo de Jonh Horton Conway com o Jogo da Vida (GREMONINI, VICENTINI, 2008).

Os AC consistem em simulações discretas no tempo, espaço e no estado do sistema. A ideia básica consiste em considerar cada posição (ou região) do domínio espacial como sendo uma célula, a qual é atribuído um estado. O estado de cada célula é modificado de acordo com o seu estado e o de suas vizinhas na etapa anterior, através de uma série de regras simples que tentam imitar as leis físicas ou biológicas. A principal característica dos AC é a facilidade com que podem ser implementados em virtude da simplicidade de sua formulação e o surpreendente retorno visual capaz de sugerir equilíbrios, padrões complexos e estruturas organizadas como formações de ondas, entre outras.

O objetivo final dos modelos AC é uma descrição do comportamento macroscópico do fenômeno e não uma descrição exata e fiel do processo microscópico. Não são, em geral, instrumentos de previsão, devendo ser abordados como um meio de experimentação. Os AC são vistos, não como substitutos dos modelos matemáticos tradicionais, mas como um primeiro passo na formulação destes modelos. Os resultados obtidos através da simulação por autômatos celulares podem confirmar hipóteses para uma posterior formulação de um modelo formal.

Segundo Oliveira (2008), recentemente, vários fenômenos naturais estão usando modelos tratados em computador para descrever sua dinâmica. Um desses modelos é o AC que apresenta várias características dos sistemas complexos: (a) são dinâmicos, isto é, evoluem no tempo e são formados por unidades simples interagindo umas com as outras; (b) possuem regras locais de evolução onde cada unidade só influencia diretamente algumas poucas vizinhanças; (c) cada unidade pode se encontrar em dois estados em que qualquer par é mutuamente excludente: branco ou preto, zero ou um e (d) a atualização é feita em paralelo, isto quer dizer que, a cada fase do tempo durante a evolução, todos os componentes têm seus estados atualizados simultaneamente.

4.2.1. Regra 90

Como o problema proposto questiona a aplicação de um autômato baseado na Regra 90, convém esclarecer esta Regra, cujas células são preenchidas segundo o padrão seguinte:

- a) Se tanto a célula da esquerda, quanto a da direita estiverem preenchidas, ou ambas vazias, a célula atual **não será preenchida**;
- b) Se somente uma das células, seja a da esquerda ou a da direita, estiver preenchida, a célula atual **será preenchida**.

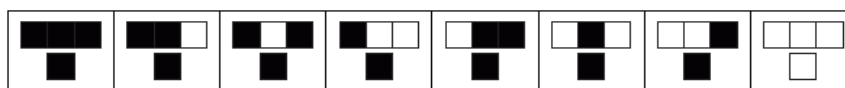


Figura 4. Regra 90
Fonte: Wolfram, 2002, p. 26.

Considera-se uma fita infinita onde, em cada instante, os estados mudarão seguindo uma determinada regra de vizinhança (Figura 4):

- se os estados das vizinhas forem diferentes, o estado posterior será “preto”;
- se os estados das vizinhas forem iguais, o estado posterior será “branco”.

Trata-se de um caso muito simples,

“[...] em que cada unidade pode se encontrar em apenas dois estados: branco ou preto, vivo ou morto, sim ou não, [...]. Em termos computacionais, cada unidade pode ser associada a um *bit* da memória do computador, cujo valor pode ser 0 ou 1. (OLIVEIRA, 2008, p. 88)

Estas regras podem ser implementadas num computador, ou de forma mais modesta num papel quadriculado. Podemos também explicitá-las em termos matemáticos: a regra 90 usa o módulo¹ do resto da divisão da soma das casas vizinhas pelo numeral dois, uma regra bem simples, apesar da surpreendente formação geométrica (Figura 5).

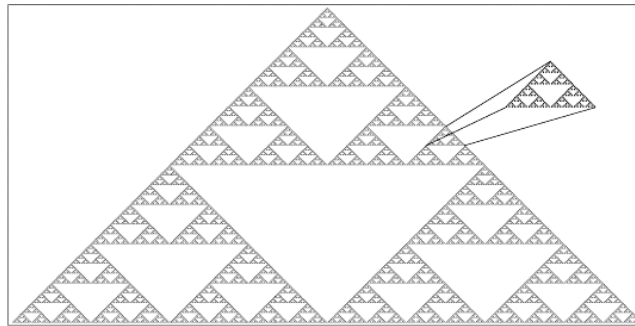


Figura 5. Regra 90
Fonte: Wolfram, 2002, p. 26.

5. Resultados

Para implementação da resolução do problema, foi utilizado o software *Visual Turing Machine* – VTM, na versão 2.0, por apresentar uma interface gráfica bem interativa. Considerando o conceito de MdT como sendo abstrato, para a compreensão da lógica, foi preciso trabalhar com um software que permitisse a visualização de todas as etapas: desde a construção da lógica à execução da fita.

A solução foi dada em pequenos módulos, chamados de máquinas, permitindo, dessa forma, analisar o problema por partes. Cada uma dessas máquinas possui uma função específica dentro do conjunto da solução, permitindo que uma faça chamada a outra. Dessa maneira, viabilizou-se a construção em módulos, descritos a seguir: (a) a máquina *Main* carrega o programa principal, chamando as máquinas Semente2 e XOR; (b) a máquina Semente2 define a semente do autômato; (c) a máquina XOR aplica o XOR nas células vizinhas, para decidir o que escrever na célula à direita da fita, identificada pelos marcadores **i** ou **j**; (d) a máquina Escreve# percorre a fita até encontrar a posição à direita adequada para escrever #, que aqui corresponde ao **zero**; (e) a máquina EscreveA percorre a fita até encontrar a posição à direita da fita adequada para escrever **a**, que aqui corresponde a **um**.

A máquina *Main* chama a máquina Semente2, estabelecendo uma seqüência inicial no autômato. Optou-se por uma configuração inicial com apenas uma “semente”, ou seja, apenas uma célula com valor **a (um)** e todas as demais com # (**zero**). Após a execução

¹ O módulo pode ser compreendido, de forma simplificada, como valor absoluto de uma grandeza, e pode ser calculado como a raiz quadrada do quadrado de um número.

dessa máquina, a fita fica com a configuração apresentada na Figura 6. Os quatro primeiros valores, representados pelos símbolos **i** e **#**, correspondem ao **zero**, o **a** corresponde ao **um**, e os quatro últimos, representados por **#** e **f**, correspondem ao **zero**.



Figura 6. Desenho da semente na fita
 Fonte: Própria, 2009

Depois de desenhada a semente na fita, reposiciona-se o cabeçote no início da semente para que a máquina *Main* chame a máquina XOR, montando as demais linhas do autômato. O XOR é um dispositivo lógico, o ou-exclusivo, que pode ser representado por $F = (A \oplus B)$. A condição para que o estado seja zero é que ambos sejam iguais. Fazendo relação com o problema da concha, percebe-se que o mesmo simula um AC, mais precisamente a regra 90 (Figura 5), de acordo com a classificação de regras de Wolfram (2002).

A máquina XOR é apenas uma máquina de decisão e não de escrita, portanto, analisa os vizinhos para decidir o que escrever. Se os vizinhos forem iguais, chama a máquina Escreve#, que imprime **# (zero/branco)** na posição adequada da outra linha; caso contrário, chama a máquina EscreveA, que escreve **a (um/preto)** na posição adequada da linha subsequente (Figura 7). Toda nova linha é construída em função da aplicação do XOR na linha anterior, até que toda a padronagem seja construída.

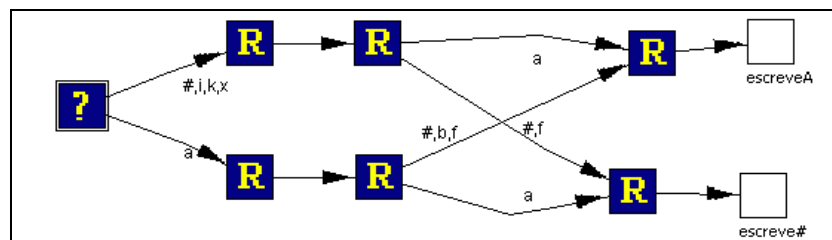


Figura 7. Máquina do XOR
 Fonte: Própria, 2009

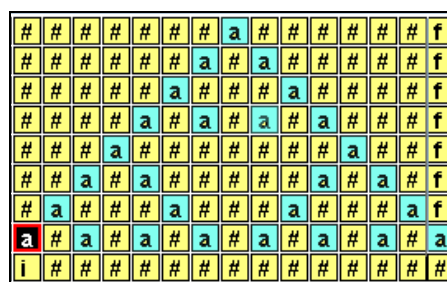


Figura 8. Exemplo de padronagem final da fita após a aplicação da lógica
 Fonte: Própria, 2009

A regra 90 tem um comportamento interessante quando aplicada a sistemas finitos, como é o caso do problema da concha. Após $2N - 1$ passos, o sistema retorna ao estado original, o que dá o efeito do triângulo de *Sierpinski* (Figura 5). A Figura 8 ilustra o

resultado final da aplicação da lógica utilizada para a solução do problema, quando aplicada a uma semente de tamanho maior. Os triângulos são redesenhados a esquerda e a direita até que ocupe o tamanho inicial definido para a suposta matriz.

6. Conclusão

A partir da experiência da disciplina Sistemas de Representação do Conhecimento, pode-se concluir que o PBL se mostrou um método adequado não apenas para resolução do problema, mas, também, para aquisição dos conhecimentos necessários a esse fim. Foi decisivo o perfil do grupo, em face da formação multidisciplinar de seus componentes. Isso possibilitou múltiplas e variadas abordagens, como geração, gestão e resolução de tensões, mas implicou em um aprendizado que foi além da resolução do problema proposto.

Quanto à lógica utilizada para a solução do problema, percebeu-se que, em grau diferente, todos puderam colaborar, de modo que a solução apresentada foi construída coletivamente, conforme intenção e proposta do PBL. Outro aspecto importante foi a conduta e condução do tutor, mediando a construção coletiva do conhecimento, orientando o grupo, questionando, estimulando novas pesquisas e descobertas. O grupo se mostrou motivado, alguns com a novidade do conteúdo; outros com a vontade em solucionar o problema.

A experimentação do PBL se configurou também como outro tipo de (auto) conhecimento: de um lado os menos experientes, tanto na metodologia utilizada quanto no conteúdo trabalhado, aprenderam com o outro que já havia percorrido trilhas anteriormente desbravadas; de outro lado os que já tinham conhecimentos prévios do conteúdo e da metodologia, romperam preconceitos, muitas vezes, abandonando a “proteção de seu próprio campo”, de seu “*locus*” habitual em que desenvolve o saber. Enfim, o PBL, nesse caso, favoreceu a aprendizagem coletiva e colaborativa.

Referências

Acióly, B. M. e Bedregal, B. R. C. Introdução a Teoria da Computação: Linguagens Formais e Computabilidade. Natal: Editora Edições UNP, 2000.

Algarve, Universidade. PBL: Problem Based Learning. Medicina. Disponível em: www.medicina.ualg.pt/site/index.php?module=Content&func=view&pid=28. Acesso em: 27 jun. 2011.

Coelho-Filho, J.; Soares, S.; SA, H. Problem-based learning: application and possibilities in Brazil. São Paulo Medicine Journal, v. 116, n. 4. São Paulo, July/Aug., 1998. Disponível em: www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-31801998000400009. Acesso em: 27 jun. 2011.

Delisle, R. How to use problem-based learning in the classroom. Alexandria: Ascd, 1997.

Gremonini, L.; Vicentini, E. Autômatos celulares: revisão bibliográfica e exemplos de implementações. Revista eletrônica lato sensu – UNICENTRO, 6 ed., 2008. Disponível em:

http://web03.unicentro.br/especializacao/Revista_Pos/P%C3%A1ginas/6%20Edi%C3%A7%C3%A3o/Exatas/PDF/1-Ed6_CE-Autom.pdf. Acesso em: 27 jun. 2011.

Loula, A. Aprendizado baseado em problemas no curso de Engenharia da Computação. Departamento de ciências exatas, UEFS, 2009. Disponível em: www.ecologia.ufba.br/PBL%20Enga%20Computacao%20UEFS.ppt. Acesso em: 27 jun. 2011.

Mamede, S.; Penaforte, J. Aprendizagem Baseada em Problemas: anatomia de uma nova abordagem educacional. Fortaleza: Hucitec, 2001.

Oliveira, P. M. A. Autômatos Celulares. In: Complexidade e Caos. Org H. M. Nussenzveig. Rio de Janeiro: Edt. UFRJ/Copea, 2008.

Pereira, H. et al. Estudo do design da interface para o ambiente PBL-VE. Revista Design em foco, julho-dezembro, 2005, v. 2, n. 2, Universidade do Estado da Bahia, Brasil, p. 95-110. Disponível em: <http://redalyc.uaemex.mx/redalyc/pdf/661/66120209.pdf>. Acesso em: 27 jun. 2011.

Ribeiro, L.; Mizukami, M. A PBL na Universidade de Newcastle: um modelo para o ensino de engenharia no Brasil? Olhar de Professor, v. 7, n. 1. Universidade Estadual de Ponta Grossa, Ponta Grossa, Brasil. p. 133-146, 2004. Disponível em: <http://redalyc.uaemex.mx/redalyc/pdf/684/68470110.pdf>. Acesso em: 27 jun. 2011.

Santos, D. M. B; Pinto, G. R. P.; Sena, C. P. P.; Bertoni, F.C.; Bittencourt, R. A. Aplicação do Método de Aprendizagem Baseada em Problemas no Curso de Engenharia de Computação da Universidade Estadual de Feira de Santana. In: XXXV Congresso Brasileiro de Educação em Engenharia, 2007, Curitiba. Anais do COBENGE 2007. Curitiba - PR: 2007. p.2A07-1 - 2A07-14.

Soares, M.; Arauno, A. Aplicação do método de ensino problem based learning (PBL) no curso de ciências contábeis: um estudo empírico. Disponível em: www.furb.br/especiais/download/879756-215326/epc%20045%20-%20index3.php.pdf. Acesso em: 27 jun. 2011.

UNIFESP. Aprendizado baseado em problemas. www.unifesp.br/centros/cedess/pbl/. Acesso em: 27 jun. 2011.

Wolfram, Stephen. A New Kind of Science. Champaign, IL: Wolfram Media, Inc., 2002. Disponível em: <http://www.wolframscience.com/nksonline/toc.html>. Acesso em: jun. 2011.