

# Ambiente Moodle de auxílio ao ensino e aprendizagem em Linguagens Formais

Robson da Silva Aguiar, Janne Yukiko Yoshikawa Oeiras

Programa de Pós-Graduação em Ciências da Computação – Universidade Federal do Pará (UFPA).

Caixa Postal 479 – 66.075-110 – Belém – PA – Brasil

robson\_cbcc@yahoo.com.br, joeiras@ufpa.br

**Abstract.** *This paper presents a computational environment to support Theory of Computation learning and teaching. In the literature can be found several other environments that allow students to create and test formal models that solve computational problems proposed by the teacher. This work presents adaptations in Moodle standard version in order to allow the creation of questionnaire with specific questions of Formal Languages that can be corrected automatic or semi-automatically.*

**Resumo.** *Este artigo apresenta um ambiente computacional de suporte à aprendizagem de disciplinas de Teoria da Computação. Na literatura podem ser encontrados diversos outros ambientes que possibilitam ao aluno criar e testar modelos formais que solucionam problemas computacionais propostos pelo professor. O diferencial desta adaptação Moodle em relação à versão padrão está no fato de permitir a elaboração de questionário contendo questões especiais de Linguagens Formais que podem ser corrigidas de modo automático ou semi-automático.*

## 1. Introdução

A importância de disciplinas de Teoria da Computação (TC) é reconhecida no currículo de referência da Sociedade Brasileira de Computação para cursos de graduação em Ciência da Computação e Engenharia de Computação (SBC, 2003) devido se centrarem na matéria **Fundamentos da Computação**. Logo, o conhecimento assimilado de disciplinas de TC como: Linguagens Formais, Autômatos e Computabilidade e Análise de Algoritmos é indispensável para a boa formação do profissional de Computação.

Contudo, na literatura, existem diversos trabalhos que relatam as dificuldades que os alunos têm em compreender os conceitos estudados em TC (Gramond e Rodger, 1999; Chesñevar et al., 2004; Costa, Meneses e Uber, 2008; Goyal e Sachdeva, 2009). Por exemplo, nos trabalhos de Costa, Meneses e Uber (2008) e Goyal e Sachdeva (2009) são apontadas diversas razões para estas dificuldades dos alunos em TC como: a base matemática exigida pelas disciplinas de TC que os alunos não têm; os livros de TC em sua maioria são direcionados a conceitos teóricos e matemáticos distanciando o aluno de uma aprendizagem por meio da prática; e o baixo uso de ferramentas computacionais de visualização destinadas a superar dificuldades de aprendizagem em TC.

A respeito desse último problema, diversas ferramentas computacionais (JFLAP (Rodger, 2010), VAS (Bovet, 2006), Language Emulator (Vieira et al, 2002) e SCTMF (Costa, Meneses e Uber, 2008)) têm sido desenvolvidas para dar apoio à aprendizagem de disciplinas de TC, as quais exigem que os alunos resolvam exercícios propostos, a fim de assimilar seus conteúdos. Em Linguagens Formais e Computabilidade é assim, o aluno deve resolver exercícios e, se possível, utilizar uma ferramenta computacional para ajudá-lo e tornar sua aprendizagem mais interativa e prazerosa (Goyal e Sachdeva, 2009).

Neste contexto, na Universidade Federal do Pará, embora o incentivo para a realização de exercícios destas disciplinas seja ressaltado pelo professor com frequência, aliado ao uso da ferramenta computacional JFLAP (Rodger, 2010), têm se notado que os alunos não resolvem as listas de exercícios e, quando as resolvem alguns não utilizam a ferramenta computacional para auxiliá-los. Desta forma, a depender do tamanho da turma é ainda mais difícil para o professor saber se todos os seus alunos respondem os exercícios propostos por ele e se usam as ferramentas computacionais indicadas, logo o acompanhamento da aprendizagem pelo professor não atinge todos os seus alunos.

Em contrapartida, com a ampliação do uso de Educação a Distância (EAD) no Brasil, diversas universidades têm adotado ambientes virtuais de aprendizagem como o Moodle (Moodle, 2010), o TelEduc (Teleduc, 2010) e o Amadeus (Amadeus, 2010) buscando a facilidade na disposição e organização de conteúdos online e a extensão do ambiente físico universitário, estendendo a interação entre todos os participantes de um curso ou disciplina além dos limites de salas de aula. Assim, dado seus empregos na EAD é comum haver ferramentas para acompanhamento das ações realizadas pelos alunos nestes ambientes virtuais, que podem rastrear a frequência de entrada deles no ambiente, por quais partes eles passaram, quais atividades entregaram, com quem se comunicaram e etc.

Deste modo, visando aproveitar as facilidades de gerenciamento e acompanhamento de alunos que esses ambientes provêem, neste trabalho é apresentada uma adaptação no ambiente Moodle (Moodle, 2010) para o estudo de TC com o apoio da ferramenta JFLAP (Rodger, 2010). Esta adaptação permite ao professor incluir, além dos tipos de exercícios já proporcionados pelo ambiente, exercícios específicos para o estudo de Linguagens Formais e Computabilidade.

Ao tentar responder esses tipos de exercícios específicos, o aluno, após a submissão de sua solução a um problema proposto pelo professor, recebe *feedback* imediato informando se esta solução está correta ou não. Para que este feedback seja possível de ser informado, um mecanismo de correção automático ou semi-automático deve existir para estes tipos de exercícios. Assim, esse mecanismo de correção tem se mostrado relevante os alunos, pois eles esperam saber, o mais breve possível, se a sua resposta está certa ou errada; e também para o professor que têm a sua sobrecarga de acompanhamento da aprendizagem de seus alunos minimizada.

Na seção 2 serão apresentados alguns pontos que podem influenciar no processo de ensino-aprendizagem em TC. Na seção 3 o ambiente Moodle (Moodle, 2010) será apresentado com adaptações relativas à incorporação de funcionalidades do ambiente Laboratório de Linguagens Formais (LabLF) e, como ele pode vir a ser utilizado, a fim

de auxiliar professores e alunos de Linguagens Formais. O último tópico contém as considerações finais do trabalho.

## **2. Ensino-aprendizagem em Teoria da Computação**

Devido às várias dificuldades existentes no processo de ensino e aprendizagem de TC, há trabalhos que descrevem atividades que podem ser realizadas visando aproximar as disciplinas dessa área a outras, a fim de aumentar o fascínio de professores pela tarefa de ensinar e a participação ativa dos alunos (Chesñevar et al., 2004; Goyal e Sachdeva, 2009). Dentre essas atividades, há o destaque para a associação, durante toda uma disciplina, da teoria à prática por meio de exercícios; exemplos para elucidar detalhes; e ilustrações para tornar menos abstrata a teoria estudada. Os tópicos a seguir discursam sobre pontos importantes do ensino e aprendizagem em TC que devem ser contemplados por professores dessa área.

### **2.1. Base matemática teórica**

Para as disciplinas de TC, o conhecimento teórico matemático é necessário e primordial. Isto porque é exigido que provas sejam dadas, a fim de garantir com sustentação matemática que uma determinada solução está correta ou incorreta. Para Krishnan (2002) uma das principais dificuldades dos alunos em TC é entender e escrever (ou construir) provas. Portanto, cabe ao aluno estudar, concomitantemente, as teorias matemáticas envolvidas nos assuntos estudados nas disciplinas em que se têm dificuldades.

Mesmo sabendo que é responsabilidade dos alunos terem conhecimento dos conceitos matemáticos utilizados nas disciplinas de TC, o professor pode realizar uma avaliação inicial. Esta avaliação de conteúdo matemático teórico tem a função de descobrir se o nível do conhecimento matemático assimilado, anteriormente, por seus alunos é satisfatório para o início dos estudos destas disciplinas.

Com base nesta avaliação inicial, o professor pode incluir os assuntos iniciais dos livros de TC como os primeiros assuntos da disciplina a serem ministrados em sala. Assim, o professor deve focar neste início conceitos matemáticos como: alfabeto, linguagem e formalismos matemáticos de descrição de conjuntos. Isto é válido no momento em que houver um número considerável de alunos que possuam deficiência destes e de outros conceitos matemáticos necessários à disciplina.

Caso contrário, o professor pode sugerir que seus alunos estudem estes assuntos com o passar do tempo, a medida que o entendimento do assunto abordado em sala requerer conhecimentos matemáticos ainda não totalmente assimilados pelo aluno. Desta maneira, o professor tende a ter seu trabalho de ensinar TC minimizado e a aprendizagem dos alunos direcionada ao conteúdo de TC em si.

### **2.2. Teoria e prática**

Nas disciplinas de TC os alunos são submetidos a um grande número de conceitos teóricos e, geralmente, o professor dispõe de exercícios para que tais conceitos sejam praticados por eles. Assim, para alguns alunos, o não entendimento sobre os conceitos teóricos estudados e suas aplicações na resolução de problemas reais é um dos fatores

que os levam à desmotivação em estudar TC e a não fazer os exercícios propostos pelo professor.

Uma solução que pode ser dada a este problema que se relaciona à tarefa de ensino do professor é propor exercícios contextualizados. Ou seja, os exercícios podem ilustrar problemas diversos, porém sua solução deve estar embasada sobre os conceitos e as técnicas estudadas. Goyal e Sachdeva (2009. p. 140) afirmam que “(...) *Teaching through integration of subjects had shown how students matured in their understanding of various concepts during their learning on the course*”.

Tais exercícios contextualizados tendem a motivar os alunos no instante em que conseguem visualizar o emprego da teoria estudada em TC em aplicações reais. Por exemplo, propor um exercício em que o aluno construa uma página Web de acesso a um ambiente é uma boa maneira de aplicar os conceitos estudados em Expressão Regular (ER), pois, ao validar *logins* o aluno deveria, idealmente, construir uma ER para reconhecer todas as entradas possíveis conforme um determinado padrão (poderiam ser emails, neste caso a ER deveria obrigatoriamente possuir o símbolo “@” em sua composição).

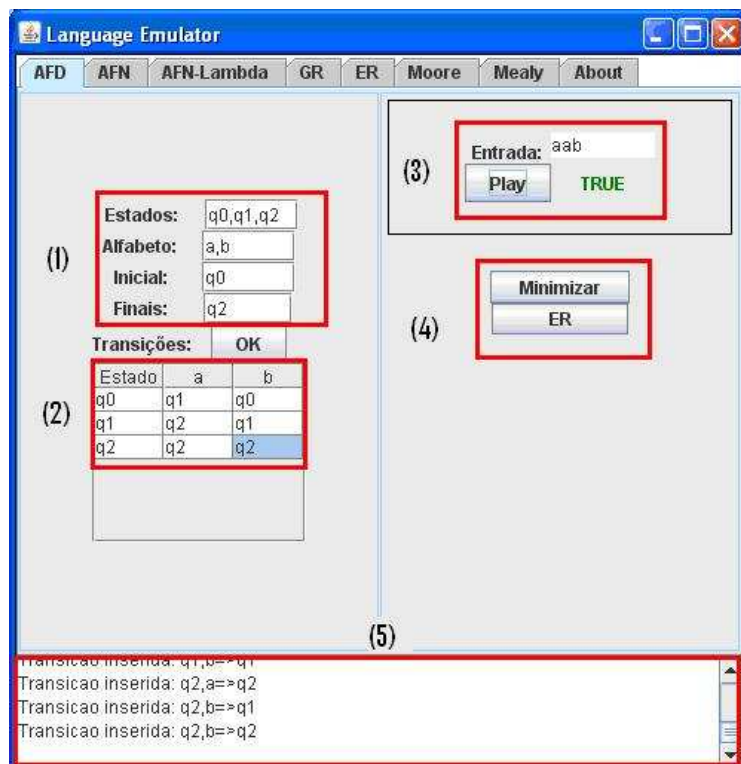
### 2.3. Exemplos e Ilustrações

Quando exemplos são bem formalizados em disciplinas teóricas, a percepção do aluno acerca do uso dos conceitos e técnicas estudadas na resolução de problemas tendem a ser aumentadas. De maneira similar, ao se utilizar ilustrações para modelar uma provável solução a um problema, a aprendizagem poderá se tornar mais fácil.

É neste ponto que ferramentas visuais, como o JFLAP (Rodger, 2010), o VAS (Bovet, 2006), o Language Emulator (Vieira et al, 2002) e o SCTMF (Costa, Meneses e Uber, 2008), devem ser usadas para efetivar a assimilação dos conceitos e técnicas estudados nas disciplinas teóricas. Goyal e Sachdeva (2009. p. 137) afirmam que a visualização ajuda a perceber aquilo que está invisível e, ajuda a introduzir conceitos e processos através de uma maneira organizada. O software JFLAP (Rodger, 2010) é citado como uma possível ferramenta de auxílio na tarefa de ensinar do professor e na aprendizagem do aluno.

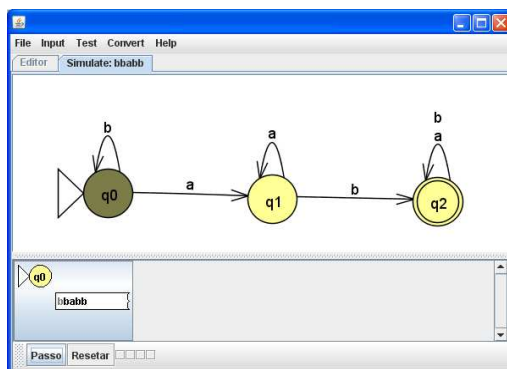
Os professores, então, podem utilizar determinadas ferramentas para ajudar os alunos a assimilarem certos pontos do assunto estudado em sala. Por exemplo, o professor poderá propor a seus alunos que utilizem a ferramenta Language Emulator (Vieira et al, 2002) (ou o SCTMF (Costa, Meneses e Uber, 2008)) para auxiliá-los a construir formalmente modelos computacionais ou a ferramenta JFLAP (Rodger, 2010) para que verifiquem todos os passos de reconhecimento de uma palavra por um Autômato Finito (AF) (ou o VAS (Bovet, 2006) que proporciona ao usuário a construção dos modelos computacionais AF e Máquina de Turing graficamente).

A Figura 1 apresenta a ferramenta Language Emulator (Vieira et al, 2002) para construção formal do modelo computacional AF. Nele, o usuário insere os estados do AF, o alfabeto, o estado inicial e conjunto de estados finais em (1), as transições em (2), testa palavras em (3) e converte para modelos equivalentes e minimiza em (4).

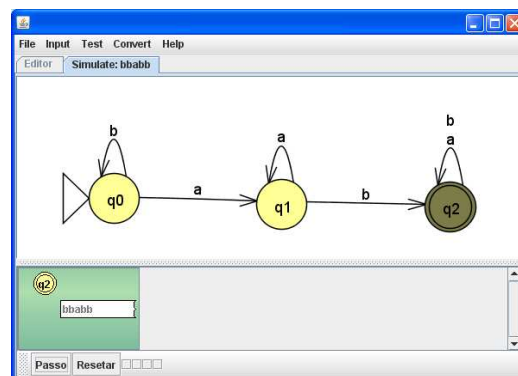


**Figura 1. Definição formal de um AF no Language Emulator (Vieira et al, 2002).**

A Figura 2 ilustra a funcionalidade de reconhecimento de autômatos no JFLAP (Rodger, 2010), que percorre toda a palavra escrita na fita de entrada (Figura 2-a), um símbolo por vez, e responde ao fim se a palavra foi aceita ou não (Figura 2-b). Este processo permite ao aluno observar e assimilar como o autômato realiza seu trabalho de reconhecimento de palavras.



**Figura 2-a: Leitura do primeiro “b”.**



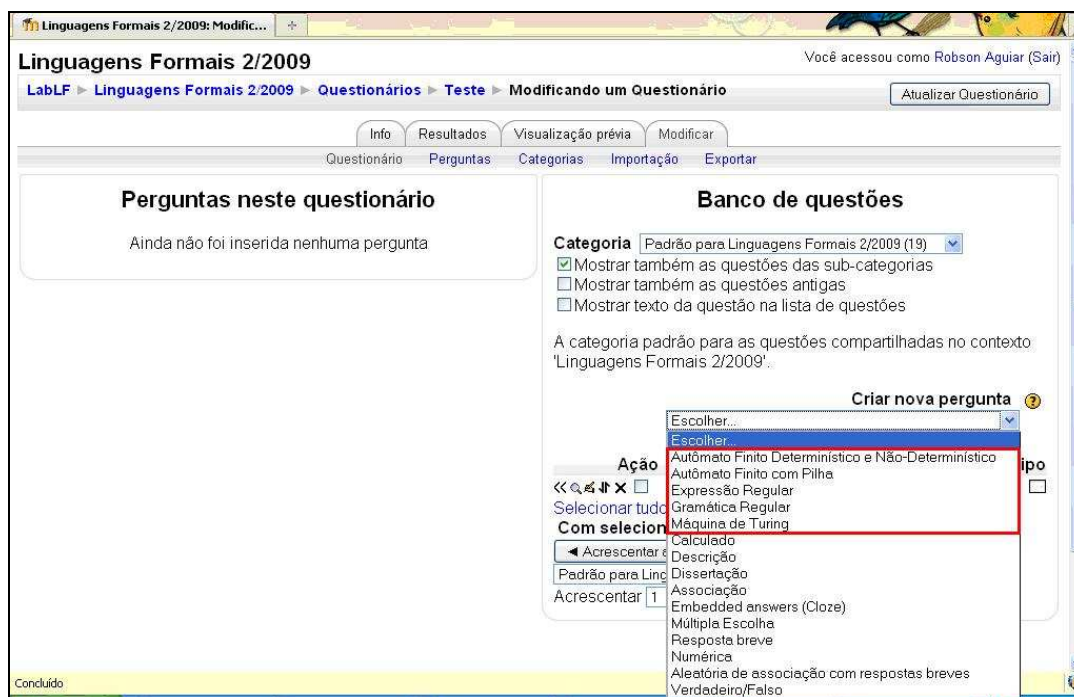
**Figura 2-b: Reconhecimento da palavra como aceita.**

Assim, levando em consideração todos estes pontos importantes que o professor de TC deve atentar e aproveitando o poder de gerenciamento de usuários e as ferramentas de acompanhamento existente no Moodle (Moodle, 2010), a seção 3 a seguir apresenta uma adaptação Moodle (MOODLE, 2010) para o estudo de Linguagens Formais e Computabilidade.

### 3. Uma adaptação do Moodle

O ambiente Moodle (Moodle, 2010) é um software desenvolvido para apoiar a realização de cursos a distância via Web (EAD) sobre qualquer assunto. Esse ambiente tem código aberto e é desenvolvido na linguagem PHP com comunicação a Banco de Dados MySQL ou PostgreSQL. Sua implementação leva em consideração padrões de projeto que permitem a inclusão de novos módulos à sua base de dados. Além disso, permite pelo menos três perfis de acesso (Tutor, Moderador e Estudante), onze tipos de atividades (uma delas é o questionário, onde poderão ser construídas questões específicas de TC) e seis tipos de recursos (um deles é o “Link a um arquivo ou site”, que pode servir como ponto de comunicação com outros ambientes).

A adaptação realizada no Moodle (Moodle, 2010) estende a infra-estrutura proporcionada por esse ambiente, permitindo a criação de questionários contendo questões específicas (Autômato Finito Determinístico e Não-Determinístico, Expressão Regular e Gramática Regular, as quais são corrigidas automaticamente, e questões de Autômato Finito com Pilha e Máquinas de Turing que são corrigidas de maneira semi-automática) para o estudo de Linguagens Formais e Computabilidade.



**Figura 2. Tipos de questões específicas no Moodle (Moodle, 2010).**

A Figura 2 exhibe, no retângulo sem preenchimento de bordas vermelhas, os cinco tipos de questões específicas de TC que a adaptação no Moodle (MOODLE, 2010) disponibiliza para o professor.

A Figura 3 apresenta a tela para cadastramento pelo professor de uma destas questões específicas, por exemplo, uma questão de Autômato Finito Determinístico e Não-Determinístico. Nela, os pontos mais importantes são: a escrita do enunciado (1), a solução da questão inserida por meio do editor (2) e as palavras de teste (3) utilizadas nos *feedbacks* dado ao aluno e base para o processo de correção de todas as questões específicas.

Linguagens Formais 2/2009 Você acessou como Robson Aguiar (Sair)

LabLF > Linguagens Formais 2/2009 > Questionários > Anterior > Modificando um Questionário > Editando uma pergunta de autômato finito determinístico

### Adicionando uma questão de Autômato Finito Determinístico

**Geral**

Categoria: Padrão para Linguagens Formais 2/2009

Nome da pergunta\*: Questao 1

Texto da pergunta

(1)

Construa um AFD que reconheça todas as palavras sobre o alfabeto  $\{0,1\}$  terminadas por 1.

Formato: Formato HTML

Avaliação predefinida\*: 1

Fator de penalidade\*: 0.1

Feedback geral

Você acertou

Resposta correta. Preencha a resposta a seguir.

**Resposta**

Resposta

(2)

```

graph LR
    start(( )) --> q0((q0))
    q0 -- 0 --> q0
    q0 -- 1 --> q1(((q1)))
    style start fill:none,stroke:none
  
```

Palavras de teste

(3)

0, 1, 00, 01, 10, 11, 100, 1001

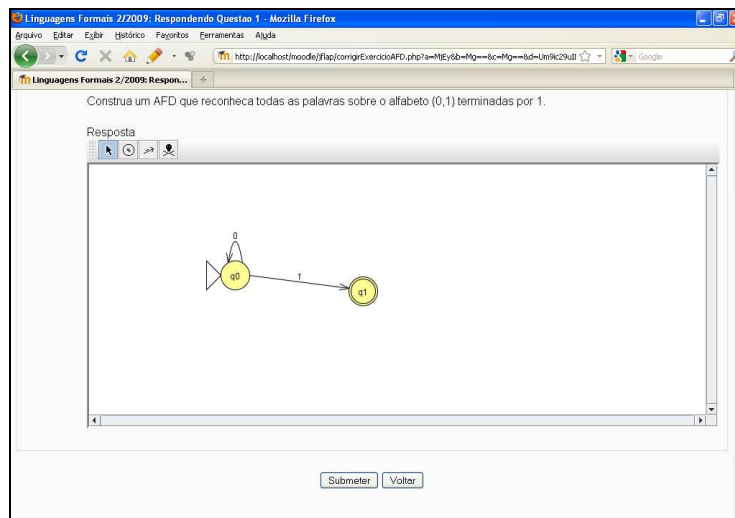
[Salvar mudanças](#) [Cancelar](#)

Este form contém campos obrigatórios

[Documentação de Moodle relativa a esta página](#)  
 Você acessou como Robson Aguiar (Sair)  
 Linguagens Formais 2/2009

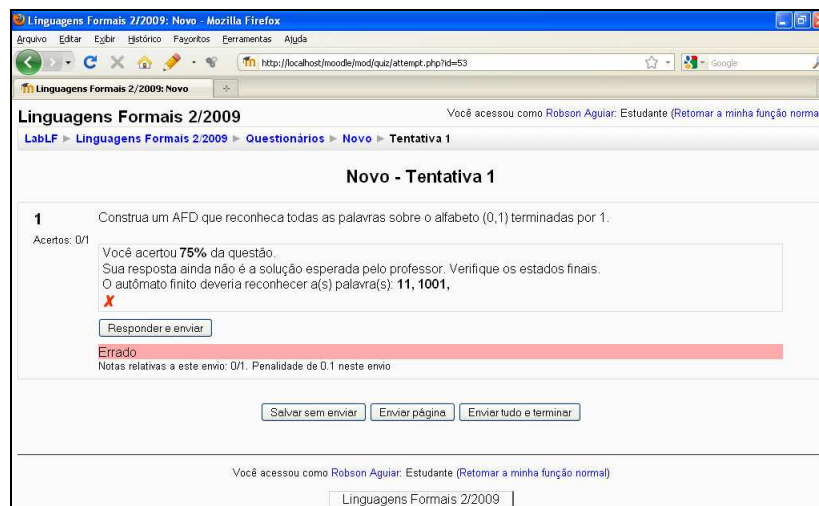
**Figura 3. Cadastramento de uma questão de AFD/AFND no Moodle (Moodle, 2010).**

Por outro lado, quando os alunos acessarem o Moodle (MOODLE, 2010) e tentarem responder o questionário que contenha esta questão específica inserida pelo professor (Figura 3), a tela vista na Figura 4 será exibida a eles.



**Figura 4. Tela para resposta dos alunos no Moodle (Moodle, 2010).**

A Figura 5 abaixo apresenta a tela de *feedback* quando as soluções propostas pelos alunos na Figura 4 são submetidas para correção ao pressionar o botão “Submeter”.



**Figura 5. Tela de feedback para os alunos.**

Analisando a Figura 5 podemos verificar que, para a solução dada ao exercício da Figura 4, o *feedback* informado foi de 75% de acerto da questão em relação à resposta dada pelo professor. Esse *feedback* também apresenta algumas palavras teste que deveriam ser reconhecidas pela solução do aluno e que não foram aceitas. Ou seja, por meio dos *feedbacks* que informam percentagens de acerto, palavra a serem aceitas e palavras a serem rejeitadas o aluno tem a possibilidade de reorganizar sua solução e tentar encontrar a solução dada pelo professor.

É no campo (3) da Figura 3 que tais palavras de teste são inseridas e separadas por vírgulas uma das outras. Assim, no momento em que o professor cria uma questão específica de TC, o modelo computacional construído como solução desta questão é testado com estas palavras e as respostas destes testes (palavras aceitas e rejeitadas) são armazenadas para os processos de correções da solução do aluno.

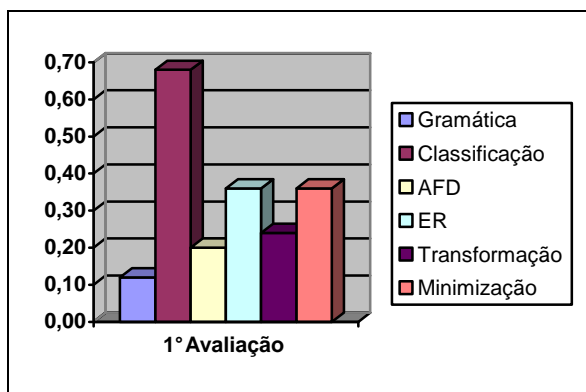


Assim, pelo fato destas palavras de testes serem cruciais para os alunos tentarem alcançar a solução do professor, elas são utilizadas nos dois processos de correção de questões específicas de TC: o automático e o semi-automático. Esta classificação da correção de questões específicas em automática e semi-automática se deve unicamente à equivalência que existe entre os modelos computacionais Autômato Finito Determinístico e Não-Determinístico, Gramática Regular e Expressão Regular, que são então corrigidos automaticamente, e os modelos formais Autômato com Pilha e Máquina de Turing, que não possui esta equivalência, corrigidos de maneira semi-automática.

#### 4. Considerações finais

O ambiente Moodle (Moodle, 2010), com a adaptação para o estudo de Linguagens Formais e Computabilidade, está em fase de testes em uma turma real de Linguagens Formais na UFPA. Porém, antes desta adaptação outro ambiente denominado Laboratório de Linguagens Formais (LabLF) já permitia a criação das questões específicas de TC e foi utilizado durante um semestre na turma de Linguagens Formais da UFPA.

Essa utilização do LabLF se mostrou útil no entendimento dos tópicos abordados em sala de aula pelos alunos, pois pôde ser observada a baixa percentagem de erros em questões de provas relativas a assuntos em que o LabLF fôra utilizado com maior frequência. A Figura 6 mostra um gráfico gerado a partir dos resultados da 1ª avaliação dos alunos na disciplina de Linguagens Formais no ano de 2009.



**Figura 6. Percentagens de alunos que erraram ou não fizeram as questões da prova.**

Este gráfico (Figura 6) representa o percentual de alunos que erraram ou não fizeram as questões da prova, as quais abordavam os assuntos: Gramática, Classificação de Linguagens, Autômatos Finitos Determinísticos, Expressão Regular, Transformação AFND-AFD e o processo de Minimização.

O que se pode notar a partir dos dados deste gráfico é que o uso do LabLF nos estudos de Autômatos Finitos Determinísticos foi determinante para o baixo percentual de alunos que erraram ou não fizeram a questão sobre AFD (20%). E o alto percentual para a questão sobre Classificação de Linguagens (68%) supõe ser o pouco contato dos alunos com materiais diferentes dos abordados em sala.

Ao fim da disciplina uma entrevista foi realizada com os alunos para levantar possíveis pontos fortes e fracos do ambiente. Por meio desta foi possível detectar que o

*feedback* no LabLF era insuficiente e que a possibilidade de saber rapidamente se um modelo computacional dado como solução a um problema proposto pelo professor é a solução esperada por este ou não. Assim, como trabalhos futuros as interfaces implementadas precisam ser revisadas quanto às sinalizações de acertos e erros que o Moodle (MOODLE, 2010) informa, além do aperfeiçoamento do processo de correção automático e semi-automático das questões específicas.

## 5. Referências

- AMADEUS. Disponível em: < <http://amadeus.cin.ufpe.br/blog/?p=37>>. Acesso em: 02 mar. 2010.
- BOVET, Jean. **Visual Automata Simulator**: A tool for simulating, visualizing and transforming finite state automata and Turing Machines. 2006. Disponível em <<http://www.cs.usfca.edu/~jbovet/vas.html>>. Acesso em: 08 mar. 2010.
- CHESÑEVAR, Carlos Iván et al. **Teaching Fundamentals of Computing Theory: A Constructivist Approach**. Journal Of Computer Science & Technology: JCS&T, Espanha, p. 91-97. ago. 2004. Acesso em: 20 mar. 2010.
- COSTA, Yandre M. e G. da; MENESES, Rafael C. de; UBER, Flavio R.. **Uma Ferramenta para Auxílio Didático no Ensino de Teoria da Computação**. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 28. 2008, Belém do Pará. Anais... Introdução, p. 209.
- GOYAL, M. e SACHDEVA, S.. Enhancing Theory of Computation Teaching Through Integration with other Courses. **International Journal Of Recent Trends In Engineering**, Oulu, p. 137-140. maio 2009. Disponível em: <<http://www.academypublisher.com/ijrte/vol01/no02/ijrte0102137140.pdf>>. Acesso em: 26 abr. 2010.
- GRAMOND, E. e RODGER, S. H. **Using JFLAP to interact with theorems in automata theory**. Thirtieth SIGCSE Technical Symposium on Computer Science Education 1999, p. 336-340.
- KRISHNAN, Padmanabhan. **Providing Assistance for Proofs in the Teaching of Theory of Computation**. Disponível em: <[http://epublications.bond.edu.au/cgi/viewcontent.cgi?article=1016&context=infotech\\_h\\_pubs](http://epublications.bond.edu.au/cgi/viewcontent.cgi?article=1016&context=infotech_h_pubs)>. Acesso em: 03 mar. 2010.
- MOODLE. Disponível em: < <http://moodle.org/>>. Acesso em: 02 mar. 2010.
- RODGER, Susan H.. **JFLAP Version 7.0: RELEASED August 28, 2009**. Disponível em: <<http://www.jflap.org/>>. Acesso em: 02 mar. 2010.
- SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (Brasil) (Ed.). **Currículo de Referência da SBC para Cursos de Graduação em Computação e Informática**. Disponível em: <<http://www.sbc.org.br/educacao>>. Acesso em: 19 mar. 2010.
- TELEDUC. Disponível em: < <http://www.teleduc.org.br/>>. Acesso em: 02 mar. 2010.
- VIEIRA, Luiz Filipe Menezes, VIEIRA, Marcos Augusto Menezes e VIEIRA, Newton José. **Fundamentos da Teoria da Computação**. 2002. Disponível em <<http://homepages.dcc.ufmg.br/~lfvieira/ftc.html>>. Acesso em 08 mar. 2010.