

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Robson da Silva Aguiar

ENSINO E APRENDIZAGEM DE TEORIA DA COMPUTAÇÃO COM O
SUPORTE DE AMBIENTES COMPUTACIONAIS: UTILIZAÇÃO DE
AVALIAÇÃO AUTOMÁTICA E SEMI-AUTOMÁTICA PARA PROVER
FEEDBACK RÁPIDO PARA O ALUNO E FACILITAR O
ACOMPANHAMENTO PELO PROFESSOR

Belém
2010

Robson da Silva Aguiar

ENSINO E APRENDIZAGEM DE TEORIA DA COMPUTAÇÃO COM O SUPORTE DE AMBIENTES COMPUTACIONAIS: UTILIZAÇÃO DE AVALIAÇÃO AUTOMÁTICA E SEMI-AUTOMÁTICA PARA PROVER FEEDBACK RÁPIDO PARA O ALUNO E FACILITAR O ACOMPANHAMENTO PELO PROFESSOR.

Dissertação de Mestrado apresentada para obtenção do grau de Mestre em Ciência da Computação. Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará.

Orientadora Prof^ª Dr^ª Janne Yukiko Yoshikawa Oeiras.

Belém
2010

Robson da Silva Aguiar

ENSINO E APRENDIZAGEM DE TEORIA DA COMPUTAÇÃO COM O
SUPORTE DE AMBIENTES COMPUTACIONAIS: UTILIZAÇÃO DE
AVALIAÇÃO AUTOMÁTICA E SEMI-AUTOMÁTICA PARA PROVER
FEEDBACK RÁPIDO PARA O ALUNO E FACILITAR O
ACOMPANHAMENTO PELO PROFESSOR.

Dissertação apresentada para obtenção
do grau de Mestre em Ciência da
Computação. Programa de Pós-
Graduação em Ciência da Computação.
Instituto de Ciências Exatas e naturais.
Universidade Federal do Pará.

Data da aprovação: Belém-PA. 25-06-2010.

Banca Examinadora

Prof.^a Dr.^a Janne Yukiko Yoshikawa Oeiras.

Programa de Pós-graduação em Ciência da Computação – UFPA – Orientadora.

Prof.^a Dr.^a Alessandra Dahmer.

Universidade Federal de Ciências da Saúde de Porto Alegre – UFCSPA – Membro.

Prof. Dr. Eloi Luiz Favero.

Programa de Pós-graduação em Ciência da Computação – UFPA – Membro.

À minha Família:
Mamãe, Papai e minha Irmã e
à minha orientadora Professora Janne.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela saúde e tranqüilidade a mim dadas para que desenvolvesse da melhor maneira possível este trabalho.

Em segundo lugar agradeço à minha mãe e a meu pai pela compreensão em não poderem contar com 100% da minha ajuda no período de escrita desta dissertação. Também deixo aqui registrado um agradecimento à minha irmã, que em pleno estudo para o vestibular não a ajudei como deveria em suas dificuldades.

Em terceiro lugar agradeço à professora Janne por nunca ter desistido de me assistir e por realizar seu papel de orientadora brilhantemente. Deixo também agradecimentos aos professores Eloi e Benedito por suas valiosas observações e dicas para a escrita desta no seminário de andamento.

À Fundação de Amparo à Pesquisa do Estado do Pará, pelo apoio financeiro concedido por meio do Edital 003/2007.

À Universidade Federal do Pará e seu Programa de Pós-Graduação em Ciência da Computação pelos custeios financeiros durante todo o período de meu Mestrado.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior meus agradecimentos pelos auxílios financeiros a mim concedidos.

SUMÁRIO

1. INTRODUÇÃO	1
1.1 PROBLEMÁTICA E JUSTIFICATIVA	2
1.2 OBJETIVO GERAL.....	4
1.3 OBJETIVOS ESPECÍFICOS	4
1.4 RESULTADOS ESPERADOS E LIMITAÇÕES.....	5
1.5 TRABALHOS PUBLICADOS	6
1.6 ORGANIZAÇÃO DA DISSERTAÇÃO	6
2. REVISÃO BIBLIOGRÁFICA	7
2.1 O ENSINO E APRENDIZAGEM EM TC.....	7
2.1.1 INCENTIVAR A PARTICIPAÇÃO DOS ALUNOS	7
2.1.2 RESERVAR AULAS INICIAIS PARA ABORDAR FUNDAMENTOS MATEMÁTICOS	8
2.1.3 UTILIZAR PROVADOR DE TEOREMAS.....	8
2.1.4 CONTEXTUALIZAR E ASSOCIAR TEORIA E PRÁTICA	9
2.1.5 USAR FERRAMENTAS COMPUTACIONAIS	10
2.1.5.1 APRENDIZAGEM DA DEFINIÇÃO FORMAL DE MODELOS COMPUTACIONAIS	10
2.1.5.2 COMPREENSÃO SOBRE O PROCESSO DE CONSTRUÇÃO DE AUTÔMATOS E RECONHECIMENTO DE LINGUAGENS	12
2.2 AMBIENTES COMPUTACIONAIS PARA EDUCAÇÃO A DISTÂNCIA (EAD) 16	
2.3 CONSIDERAÇÕES DO CAPÍTULO.....	17
3. ADAPTAÇÃO DO AMBIENTE MOODLE PARA SUPORTE AO ENSINO E À APRENDIZAGEM DE TC.....	20
3.1 FACILIDADES IMPLEMENTADAS PARA O PROFESSOR.....	21
3.2 FACILIDADES IMPLEMENTADAS PARA O ALUNO	29
3.3 INTEGRAÇÃO DO JFLAP NO MOODLE	34
3.4 MATERIAIS DIDÁTICOS PARA O ENSINO E A APRENDIZAGEM EM TC..	35
3.4.1 SISTEMA DE ARQUIVOS	36
3.4.2 DOCUMENTOS AUXILIARES	37
3.4.3 MATERIAL A SER USADO EM SALA	39
3.4.4 LISTAS DE EXERCÍCIOS.....	41
3.4.5 APOSTILAS.....	42
3.4.6 ARTIGOS	44
3.4.6.1 ARTIGOS PARA O PROFESSOR.....	44
3.4.6.2 ARTIGOS PARA O ALUNO	46
3.4.7 FERRAMENTAS COMPUTACIONAIS	47
3.5 CONSIDERAÇÕES DO CAPÍTULO.....	49
4. A UTILIZAÇÃO DO AMBIENTE MOODLE EM UMA DISCIPLINA DE LINGUAGENS FORMAIS	50
4.1 O USO DAS FUNCIONALIDADES DE CORREÇÃO	50
4.1.1 NO ENSINO.....	51
4.1.1.1 OS DADOS	52

4.1.1.2	AVALIAÇÃO DA REORGANIZAÇÃO DA DISCIPLINA	53
4.1.2	USO DA CORREÇÃO AUTOMÁTICA EM AULAS EM LABORATÓRIO	54
4.2	AS ENTREVISTAS COM OS ALUNOS	58
4.3	CONSIDERAÇÕES DO CAPÍTULO	59
5.	CONCLUSÕES	61
5.1	AS CONTRIBUIÇÕES DESTE TRABALHO	63
5.2	OS TRABALHOS FUTUROS	64
	REFERÊNCIAS BIBLIOGRÁFICAS	65
	APÊNDICE A INTERFACES PARA CRIAÇÃO DE QUESTÕES ESPECÍFICAS DE TC	69
	APÊNDICE B INTERFACES PARA SUBMISSÃO DE SOLUÇÕES	74
	APÊNDICE C DETALHAMENTO SOBRE A INTEGRAÇÃO DO JFLAP NO MOODLE	77
	APÊNDICE D CRONOGRAMA	81
	APÊNDICE E INTERFACE PRINCIPAL DO CURSO BÁSICO	83
	APÊNDICE F QUESTIONÁRIO PARA PROFESSORES DE TC	88
	APÊNDICE G PERGUNTAS DA ENTREVISTA SEMIESTRUTURADA	90

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Editor para construção de AFDs.	10
Figura 2.2 – Reconhecimento de palavras no SCTMF.....	12
Figura 2.3 – Tela principal de GAM (JUKEMURA, NASCIMENTO e UCHÔA, 2005).....	13
Figura 2.4 – Interface de MT no VAS.....	14
Figura 2.5 – Editor para AFNDs no JFLAP.	15
Figura 2.6 – Ferramentas de Cursos a Distância. (ROCHA, 2000).....	16
Figura 2.7 – Reprovação em LF na UFPA.	18
Figura 3.1 – Questões específicas de TC implementadas.	22
Figura 3.2 – Tela para criação de uma questão do tipo AFD/AFND.	23
Figura 3.3 – Tela de acompanhamento de atividades realizadas.....	25
Figura 3.4 – A solução submetida ainda não é a esperada pelo professor.	26
Figura 3.5 – A solução submetida é a esperada pelo professor.	28
Figura 3.6 – A solução submetida completamente errada.....	29
Figura 3.7 – Tela para tentativa de solução de questão pelo aluno.	30
Figura 3.8 – Tela de feedback de solução completamente incorreta.....	31
Figura 3.9 – Tela de feedback de solução correta.	32
Figura 3.10 – Tela de feedback de solução correta.	32
Figura 3.11 – Tela para tentativa de solução de questão de AFP pelo aluno.	33
Figura 3.12 – Tela principal do Moodle.	36
Figura 3.13 – Sistema de arquivos do Moodle.	37
Figura 3.14 – Conteúdo de “Documentos_auxiliares”.	37
Figura 3.15 – <i>Link</i> do arquivo contendo o cronograma na disciplina na tela principal.....	38
Figura 3.16 – Conteúdo de “Material_de_sala”.	40

Figura 3.17 – <i>Link</i> do arquivo “LF_aula01_conceitos_basicos.pdf” na tela principal.....	41
Figura 3.18 – Conteúdo de “Listas_de_exercicios”.....	42
Figura 3.19 – Conteúdo de “Apostilas”.....	43
Figura 3.20 – Apostilas disponibilizadas aos alunos.....	44
Figura 3.21 – Conteúdo de “Artigos_do_professor”.....	45
Figura 3.22 – Conteúdo de “Artigos_do_aluno”.....	46
Figura 3.23 – Artigos disponibilizados aos alunos.....	47
Figura 3.23 – Conteúdo de “Ferramentas_computacionais”.....	48
Figura 3.24 – Ferramentas computacionais disponibilizadas aos alunos.....	49
Figura 4.1 – Questões da avaliação e frequência.....	56
Figura 6.1 – Tela da criação de questões do tipo Gramática Regular.....	70
Figura 6.2 – Tela da criação de questões do tipo Expressão Regular.....	71
Figura 6.3 – Tela da criação de questões do tipo Autômato Finito com Pilha.....	72
Figura 6.4 – Tela da criação de questões do tipo Máquina de Turing.....	73
Figura 7.1 – Tela para submissão de solução de questão do tipo Gramática Regular.....	74
Figura 7.2 – Tela para submissão de solução de questão do tipo Expressão Regular.....	75
Figura 7.3 – Tela para submissão de solução de questão do tipo Máquina de Turing.....	76
Figura 10.1 – Tela principal do curso básico.....	84
Figura 10.2 – Seção inicial do curso.....	85
Figura 10.3 – Material para a Miniavaliação 1.....	85
Figura 10.4 – Material para a Miniavaliação 2.....	86
Figura 10.5 – Os seminários.....	86
Figura 10.6 – Links a materiais didáticos.....	87

LISTA DE TABELAS

Tabela 4.1 – As dez melhores universidades do Brasil segundo a revista Info Exame.	52
Tabela 4.2 – Assuntos das questões e frequência das categorias.	55
Tabela 9.1 – Cronograma.	81

LISTA DE ABREVIATURAS E SIGLAS

AFD	—	Autômato Finito Determinístico.
AFND	—	Autômato Finito Não-Determinístico.
AFP	—	Autômato Finito com Pilha.
BD	—	Banco de Dados.
CC	—	Ciência da Computação.
CTD-IE	—	Concurso de Dissertações e Teses em Informática na Educação.
EAD	—	Educação a Distância.
ER	—	Expressão Regular.
GAM	—	GINUX Abstract Machine.
GR	—	Gramática Regular.
JFLAP	—	Java Formal Languages and Automata Package.
LabLF	—	Laboratório de Linguagens Formais.
LF	—	Linguagens Formais.
MOODLE	—	Modular Object-Oriented Dynamic Learning Environment.
MT	—	Máquinas de Turing.
PVS	—	Specification and Verification System.
RBIE	—	Revista Brasileira de Informática na Educação.
SCTMF	—	Sistema de Criação e Teste de Modelos Formais.
SBIE	—	Simpósio Brasileiro de Informática na Educação.
TC	—	Teoria da Computação.
VAS	—	Visual Automata Simulator.
WebCT	—	Web Course Tools.

RESUMO

Esta dissertação aponta várias dificuldades encontradas por professores e alunos em ensinar e em aprender os fundamentos de Teoria da Computação (TC), respectivamente. Dentre estas, **a dificuldade do professor em acompanhar a aprendizagem de seus alunos**, quando há sobrecarga de trabalho e um grande número de alunos por turma, e **a dificuldade dos alunos em saber rapidamente se sua solução está correta ou não** a um problema proposto pelo professor, abrem caminhos para a criação de estratégias para auxiliar professores e alunos a superarem tais dificuldades. Assim, este trabalho apresenta suas contribuições para a linha de Educação em Computação por meio do desenvolvimento de tecnologia, no caso uma adaptação do ambiente Moodle (MOODLE, 2010) para auxiliar o ensino e a aprendizagem em disciplinas de TC. **Para o professor**, um lugar onde se pode encontrar material a ser apresentado em sala, listas de exercícios, apostilas e a possibilidade de construir questões específicas de TC e acompanhar a realização delas pelos alunos por meio das funcionalidades de verificação de tarefas realizadas do ambiente. **Para o aluno**, um ambiente com apostilas, listas de exercícios, artigos, ferramentas de auxílio a sua aprendizagem e um mecanismo de correção automática e semi-automática com *feedback* rápido de questões específicas de TC em questionários Moodle propostos pelo professor.

PALAVRAS-CHAVE: Teoria da Computação, Linguagens Formais, Computabilidade, Correção Automática, Acompanhamento da aprendizagem, Moodle, Educação em Computação.

ABSTRACT

This dissertation points out several difficulties encountered by teachers and students in teaching and learning the fundamentals of the Theory of Computation (TC), respectively. Among these, **the teacher's difficulty in following the learning of their students**, when there is extra work and a large number of students per class, and **the difficulty of students to know quickly if your solution is correct or not** to a problem proposed by the teacher, open paths to create strategies to help teachers and students to overcome such difficulties. Thus, this paper presents their contributions to the line of Computer Education through the development of technology, where an adaptation of the environment Moodle (Moodle, 2010) to support teaching and learning in the disciplines of TC. **For the teacher**, a place where you can find material to be presented in class, lists of exercises, handouts and the opportunity to build specific issues of TC and follow up the implementation of them by the students through the features of verification of tasks accomplished the environment. **For students**, an environment with handouts, lists of exercises, articles, tools to aid their learning and a mechanism for automatic correction and semi-automatic with quick feedback with specific issues of TC in Moodle questionnaires proposed by the teacher.

KEYWORDS: Theory of Computation, Formal Languages, Computability, Automatic correction, Monitoring learning, Computing Environment, Moodle, Education Computing.

CAPÍTULO 1

INTRODUÇÃO

O computador é capaz de auxiliar o homem a solucionar um grande número de problemas reais. No entanto, existem problemas que não podem ser resolvidos via computador e outros que podem ser resolvidos teoricamente, porém em um tempo inviável para o seu contexto. Deste modo faz-se necessário conhecer quais tipos de problemas não podem ser resolvidos via computador e quais os que podem ser resolvidos, porém em um tempo inviável na prática. Em Ciência da Computação (CC), a linha que se dedica ao estudo destes problemas é Teoria da Computação (TC).

Teoria da Computação é dividida em três grandes eixos: a Teoria da Computabilidade, que fornece meios para identificar os problemas que não podem ser resolvidos via computador; a Teoria da Complexidade, que está relacionada à dificuldade de um problema e, intimamente ligada ao fator tempo para encontrar uma solução; e a Teoria dos Autômatos, que se destina ao estudo de modelos computacionais e constitui a base para os dois eixos anteriormente citados.

Neste contexto, TC para o profissional de CC passa a ser vista como ponto crucial para a sua boa formação. No entanto, alguns problemas que envolvem o ensino e a aprendizagem de TC podem interferir na formação deste profissional. A seguir são apresentados alguns destes problemas.

1.1 Problemática e Justificativa

A literatura aponta para a existência de dificuldades tanto de ensino de TC (CHESÑEVAR et al., 2004; GOYAL e SACHDEVA, 2009) quanto de aprendizagem (COSTA, MENESES e UBER, 2008; CHESÑEVAR et al., 2004; GOYAL e SACHDEVA, 2009; e GRAMOND e RODGER, 1999).

Tais dificuldades de ensino de TC partem do problema que é ensinar seus fundamentos que são de caráter abstratos e que envolvem base matemática (CHESÑEVAR et al., 2004; GOYAL e SACHDEVA, 2009). Outro ponto é o acompanhamento da aprendizagem de alunos, quando o professor tem uma sobrecarga de trabalho e um grande número de alunos por turma. Além disso, quando o professor é “principiante” (aquele que nunca ministrou quaisquer disciplinas ou nunca ministrou uma disciplina de TC), são somadas às dificuldades anteriores suas dificuldades em: ministrar aulas (dada sua pouca experiência), elaborar material didático a ser apresentado em sala e encontrar rapidamente estratégias (dispersas na literatura) de ensino e aprendizagem de TC que possa aplicar.

Por outro lado, na literatura há também relatos de dificuldades de aprendizagem de fundamentos de TC. Os trabalhos de Chesñevar et al. (2004) e Goyal e Sachdeva (2009) são bons exemplos disto, pois apontam a pouca base matemática dos alunos como um fator que os levam às dificuldades em disciplinas de TC. Para outro autor, a dificuldade é intrínseca ao conteúdo, pois “(...) a typical undergraduate student in Computer Science finds the classical course on automata theory very hard”¹ (KRISHNAN, 2002, p. 1083).

Estas dificuldades de aprendizagem, quando somadas ao pouco tempo extraclasse (relatado por alunos) para resolver listas de exercícios, revisar o conteúdo apresentado em sala e realizar exercícios de implementação, freqüentemente se refletem nos conceitos das avaliações individuais. Na Universidade Federal do Pará, cerca de 30% dos alunos não conseguem alcançar a média mínima de aprovação (nota cinco).

¹ Tradução do autor: “(...) um típico estudante de graduação em Ciência da Computação acha um curso clássico de teoria dos autômatos muito difícil”.

Portanto, diante destes problemas enfrentados por professores e alunos no ensino e aprendizagem de TC, vários pesquisadores vêm a utilização de ferramentas computacionais, voltadas a um estudo amplo de fundamentos de TC ou específico de modelos computacionais, como uma maneira de se alcançar uma aprendizagem ativa (CHESÑEVAR et al., 2004; GOYAL e SACHDEVA, 2009) e ainda segundo Chesñevár et al. (2004) como um excelente e motivador *link* entre teoria e prática.

Assim, é possível encontrar na literatura ferramentas computacionais voltadas a um estudo amplo de TC como: Minerva (ESTREBOU et al., 2002) apud (CHESÑEVAR et al., 2004), DeusExMachina (TAYLOR, 1998) e Tags (ESMORIS e CHESÑEVAR, 2003). Há ainda outras direcionadas especificamente ao estudo de modelos computacionais, que por sua vez podem enfatizar a aprendizagem por meio da definição formal dos modelos como o Language Emulator (VIEIRA, J. F. M.; VIEIRA, M. A. M. e VIEIRA, N. J., 2002) e o SCTMF (COSTA, MENESES e UBER, 2008); e da visualização e da manipulação de seus estados e transições como o VAS (BOVET, 2006), o GAM (JUKEMURA, NASCIMENTO e UCHÔA, 2005) e o JFLAP (RODGER, 2010).

No estudo de TC, o aluno quando utiliza ou não estas ferramentas, além de se preocupar em construir uma solução a um problema proposto, ele deve aprender a testar sua solução e verificar se a mesma está correta ou não. No caso de recair em dúvidas quanto à solução, ele (a) pode procurar o monitor ou o próprio professor para elucidá-las.

Neste contexto o tema do trabalho está direcionado à aplicação de correção automática e semi-automática de questões específicas² de Teoria da Computação no ambiente computacional Moodle (Moodle, 2010). E a pergunta de pesquisa a ser respondida é: A utilização dos mecanismos de correção automática e semi-automática de questões específicas de TC atreladas a um ambiente computacional podem ajudar o professor a acompanhar a aprendizagem de seus alunos e, ao mesmo tempo, permitir a estes, em primeira instância, saber rapidamente se seu modelo computacional proposto como solução a um problema está correto?

² As questões específicas deste trabalho são questões dos modelos computacionais: Autômato Finito Determinístico e Não-Determinístico, Autômato com Pilha, Expressão Regular, Gramática Regular e Máquina de Turing.

1.2 Objetivo Geral

O objetivo deste trabalho é auxiliar o ensino e a aprendizagem em TC provendo recursos integrados ao ambiente computacional Moodle que permita ao professor acompanhar a aprendizagem de seus alunos com uma ferramenta de correção automática e semi-automática de questões específicas de TC e, a partir destas questões específicas permitir ao aluno saber rapidamente, por meio de *feedback* do sistema, se sua solução está correta ou não, um diferencial em relação às outras ferramentas relatadas anteriormente.

1.3 Objetivos Específicos

Os objetivos específicos deste trabalho são divididos entre o auxílio dado ao professor e o dado ao aluno. Portanto, o trabalho visa auxiliar o professor com a:

- disponibilização online via Web de questões específicas de TC;
- correção de questões específicas automática ou semi-automaticamente, modificando o ambiente Moodle;
- acompanhamento da aprendizagem dos alunos observando-se a realização das atividades propostas; e
- organização de materiais diversos como: slides, apostilas, listas de exercícios, artigos, software e outros para ministrar uma disciplina de TC.

Em relação ao aluno, os objetivos específicos do trabalho são:

- permitir o acesso às questões específicas de TC online via Web;
- permitir a correção de questão específica com o uso das funcionalidades de correção automática e semi-automática;
- gerar um *feedback* a respeito da solução dada a uma questão específica; e
- auxiliar a aprendizagem por meio de acesso a materiais didáticos como: os slides apresentados em sala de aula, apostilas, listas de exercícios, artigos da área de TC, software etc.

1.4 Resultados esperados e limitações

Espera-se que alcançados os objetivos deste trabalho o professor possa criar questionários a serem disponibilizados a seus alunos, contendo questões específicas de TC que podem ser corrigidas automática ou semi-automaticamente e gerar um *feedback* rápido. Acredita-se que isto pode reduzir os problemas deste professor advindos de uma realidade de sobrecarga de trabalho e de grande número de alunos em cada uma de suas turmas. Além disso, o professor, ao usar os recursos do ambiente computacional de verificação de atividades realizadas, pode ter facilitada a tarefa de rastreamento dos alunos com possíveis dificuldades em construir modelos computacionais. Isto pode ser feito por meio da verificação das tentativas destes em solucionar questões específicas de TC propostas pelo professor em questionários.

Um resultado esperado deste trabalho é que o ambiente computacional Moodle modificado possa ser utilizado por outras instituições de ensino e por professores experientes na área ou não. Também se espera que os alunos nas disciplinas de TC tenham melhores índices de aprovação e, posteriormente, mantenham-se estimulados a usar os conceitos inerentes à TC para resolver problemas de outras áreas.

Entretanto, é necessário deixar claras as limitações deste trabalho que podem influenciar os processos de ensino e aprendizagem de TC. A primeira consiste no conjunto de questões específicas escolhidas, a saber: Autômato Finito Determinístico e Não-Determinístico, Autômato com Pilha, Expressão Regular, Gramática Regular e Máquina de Turing, para comporem novos tipos de questões no ambiente computacional. Portanto, modelos computacionais que não estiverem neste conjunto, como Máquinas de Moore e Mealy, Máquinas de Turing com Múltiplas Fitas e outras, não contam com as funcionalidades de correção automática e semi-automática.

Uma segunda limitação é referente ao processo de correção utilizado nas questões específicas de TC: a correção automática e a correção semi-automática. Os modelos computacionais que possuem algoritmos para testar a equivalência entre máquinas têm o seu processo de correção feito automaticamente. Isto se aplica à Classe das Linguagens Regulares ou tipo 3. Portanto, para os modelos relacionados às Classes de Linguagens dos tipos 0, 1 e 2, o processo de correção é o semi-automático.

Outra limitação é referente ao próprio ambiente computacional Moodle que embora possa ser utilizado sob os Bancos de Dados (BD) MySQL e PostGresSQL e uma grande parcela de navegadores, sua adaptação apresentada neste trabalho foi realizada apenas sob BD MySQL e navegador Mozilla-Firefox.

1.5 Trabalhos Publicados

Durante o desenvolvimento desta pesquisa foram publicados três trabalhos completos: o artigo “Laboratório de Linguagens Formais – LabLF” (AGUIAR e OEIRAS, 2009) no *Concurso de Dissertações e Teses em Informática na Educação* (CTD-IE) no *XX Simpósio Brasileiro de Informática na Educação* (SBIE 2009) em 2009, que foi premiado com o 3º lugar na categoria Trabalhos de Conclusão de Curso (nível graduação). Os autores foram convidados a submeter uma versão estendida desse trabalho para uma edição especial da *Revista Brasileira de Informática na Educação* (RBIE). Recentemente, o artigo completo “Ambiente Moodle de auxílio ao ensino e aprendizagem em Linguagens Formais” (AGUIAR e OEIRAS, 2010) foi aceito no *XVIII Workshop Sobre Educação em Computação* (WEI 2010), evento integrante do Congresso da Sociedade Brasileira de Computação.

1.6 Organização da Dissertação

Esta dissertação está organizada como segue. O Capítulo 2 apresenta uma revisão bibliográfica a respeito dos principais trabalhos encontrados na literatura que descrevem estratégias para auxiliar o ensino e aprendizagem em TC. O Capítulo 3 apresenta os novos tipos de questões específicas adaptadas ao ambiente computacional Moodle (MOODLE, 2010). O Capítulo 4 relata o observado a partir do uso do ambiente computacional com novos tipos de questões específicas em uma turma real. Em seguida são apresentadas no Capítulo 5 as Conclusões e ao final as Referências Bibliográficas consultadas para a escrita e desenvolvimento deste trabalho.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta uma revisão bibliográfica dos principais trabalhos encontrados na literatura que descrevem estratégias para auxiliar o ensino e aprendizagem em TC (Seção 2.1). A Seção 2.2 apresenta sucintamente as características principais dos Ambientes Computacionais voltados a cursos de Educação à Distância (EAD) na Web, os quais podem ser utilizados por professores como uma estratégia de ensino que estende o ambiente físico universitário para além dos limites de salas de aula.

2.1 O ensino e aprendizagem em TC

Na Introdução deste trabalho, algumas dificuldades do professor em ensinar fundamentos de Teoria da Computação e dos alunos em aprendê-los foram citadas. As subseções a seguir (Subseções 2.1.1, 2.1.2, 2.1.3, 2.1.4 e 2.1.5) apresentam estratégias encontradas em trabalhos, disponíveis na literatura, que propõem maneiras de reduzir ou superar algumas das dificuldades do professor e dos alunos de TC.

2.1.1 Incentivar a participação dos alunos

Os trabalhos de Berque, Johnson e Jovanovic (2001), Chesñevar et al. (2004), Chua e Winton (1987) e Goyal e Sachdeva (2009) apontam para as dificuldades do professor em ensinar fundamentos de TC quando seus alunos estão desmotivados a aprendê-los. Assim, o engajamento ativo dos alunos nas disciplinas de TC é uma maneira para o professor alcançar sua meta de ensino.

Neste contexto, algumas atividades sugeridas e que podem ser realizadas pelo professor são: a introdução de ferramentas computacionais para auxiliar os alunos a resolver os exercícios propostos em listas pelo professor (CHUA e WINTON, 1987); e discussões em sala de aula sobre aplicações reais dos conceitos estudados, incentivando o contato com pesquisas na área por meio da leitura de artigos e dissertações de mestrado (CHESÑEVAR et al., 2004; GOYAL e SACHDEVA, 2009).

2.1.2 Reservar aulas iniciais para abordar fundamentos matemáticos

Algumas disciplinas em Ciência da Computação, dentre elas as de TC, requerem que os alunos tenham base matemática. No entanto, a depender das ofertas das disciplinas, um aluno com dificuldades matemáticas pode ser matriculado em uma disciplina de TC e ter dificuldades em compreender seus fundamentos.

É assim que Chua e Winton (1987) propõem a estratégia de ofertar uma disciplina preliminar no currículo do curso para preparar os alunos com dificuldades em fundamentos matemáticos às disciplinas de TC. Basicamente, esta disciplina envolveria a apresentação de conceitos matemáticos, como alfabetos, tipos e representação de conjunto e suas operações, funções, além de tópicos sobre Teoria dos Autômatos, Linguagens Formais, Computabilidade e Complexidade Computacional, que serão vistos aprofundadamente em disciplinas de TC e ferramentas computacionais para auxiliá-los em sua aprendizagem nesses tópicos.

No entanto, é possível que o professor inicie suas disciplinas de TC com aulas de fundamentos matemáticos exigidos para os estudos de TC, caso não seja possível ofertar tal disciplina preliminar.

2.1.3 Utilizar provador de teoremas

O trabalho de Krishnan (2002) aponta para a dificuldade do professor de TC em ministrar aulas que tratam do processo de construção, escrita e entendimento de provas devido à desmotivação dos alunos em aprenderem este processo. Além dessa desmotivação, o trabalho aponta para uma preocupação adicional do professor que é tornar compreensível a seus alunos a inexistência de um algoritmo, uma “receita”, um mecanismo para automatizar este processo de prova.

Assim, Krishnan (2002) propõe o uso de provadores de teoremas como uma maneira de se conseguir a motivação e o entendimento dos alunos de que não existe um número fixo de passos para realizar provas. Em linhas gerais, ele propôs a seus alunos (em laboratório) que provassem alguns teoremas disponíveis em listas por meio de um provador de teoremas chamado PVS (OWRE, 2010).

Como resultados destas atividades em laboratório, Krishnan (2002) observou (em termos de questões perguntadas em sala e em tentativas de resolver problemas propostos como “Dever de casa”) que o entendimento dos alunos sobre tópicos estudados em TC melhorou. Portanto, é de suma importância que o professor tenha habilidades sob o provador de teoremas escolhido e no conjunto de teoremas a serem provados e disponibilizados em listas, que permitam aos alunos observarem maneiras diferentes de se provar o mesmo teorema.

2.1.4 Contextualizar e associar teoria e prática

Segundo Chesñevar et al. (2004) a contextualização histórica tem-se mostrado ser muito proveitosa e motivadora para os alunos, pois os ajudam a identificar como diferentes conceitos teóricos estão envolvidos através do tempo e como os trabalhos de pesquisa contribuem para o crescimento de um novo campo de conhecimento (neste caso TC).

Esta estratégia, segundo Chesñevar et al. (2004), pode ser implementada por meio de uma página *Web* em que os alunos possam ter acesso a diferentes leituras e *links* para outros materiais relacionados. Além disso, os materiais para o curso devem estar dispostos segundo a linha do tempo para que os alunos entendam em que cenário os fundamentos de TC a serem estudados foram necessários e, então, concebidos.

Muito embora no artigo de Chesñevar et al. (2004) haja um direcionamento à estratégia de ensinar por meio da associação entre teoria e prática, é em Goyal e Sachdeva (2009) que a integração de assuntos, focando a resolução de problemas de outras áreas com os conceitos estudados em TC, mostra significativas melhoras na percepção dos alunos em aprender autômatos, especialmente aqueles que aplicavam os conceitos teóricos estudados em TC mecanicamente.

Por exemplo, a integração de assuntos pode ser vista ao se propor a construção de um Autômato Finito (AF) que remova marcadores de comentários em uma linguagem de programação, ao se escrever uma Expressão Regular para validar um código em HTML, ao se desenhar um AFD com os passos de criação de uma tabela em SQL, etc.

2.1.5 Usar ferramentas computacionais

Ao longo desta seção 2.1, o leitor vem acompanhando que, oportunamente, são indicadas ferramentas computacionais para auxiliar professor e alunos em TC. A sugestão nos trabalhos de Berque, Johnson e Jovanovic (2001), Chesñevar et al. (2004), Chua e Winton (1987), Goyal e Sachdeva (2009) e Krishnan (2002) é que o professor indique algumas ferramentas computacionais para auxiliar seus alunos em suas dificuldades. As subseções a seguir abordam dois conjuntos destas ferramentas para dificuldades específicas dos alunos.

2.1.5.1 Aprendizagem da definição formal de modelos computacionais

Na literatura, várias ferramentas computacionais foram desenvolvidas para auxiliar os alunos com dificuldades em definir formalmente um modelo computacional estudado em disciplinas de TC. Uma delas é o Language Emulator (VIEIRA, J. F. M.; VIEIRA, M. A. M. e VIEIRA, N. J., 2002) (Figura 2.1).

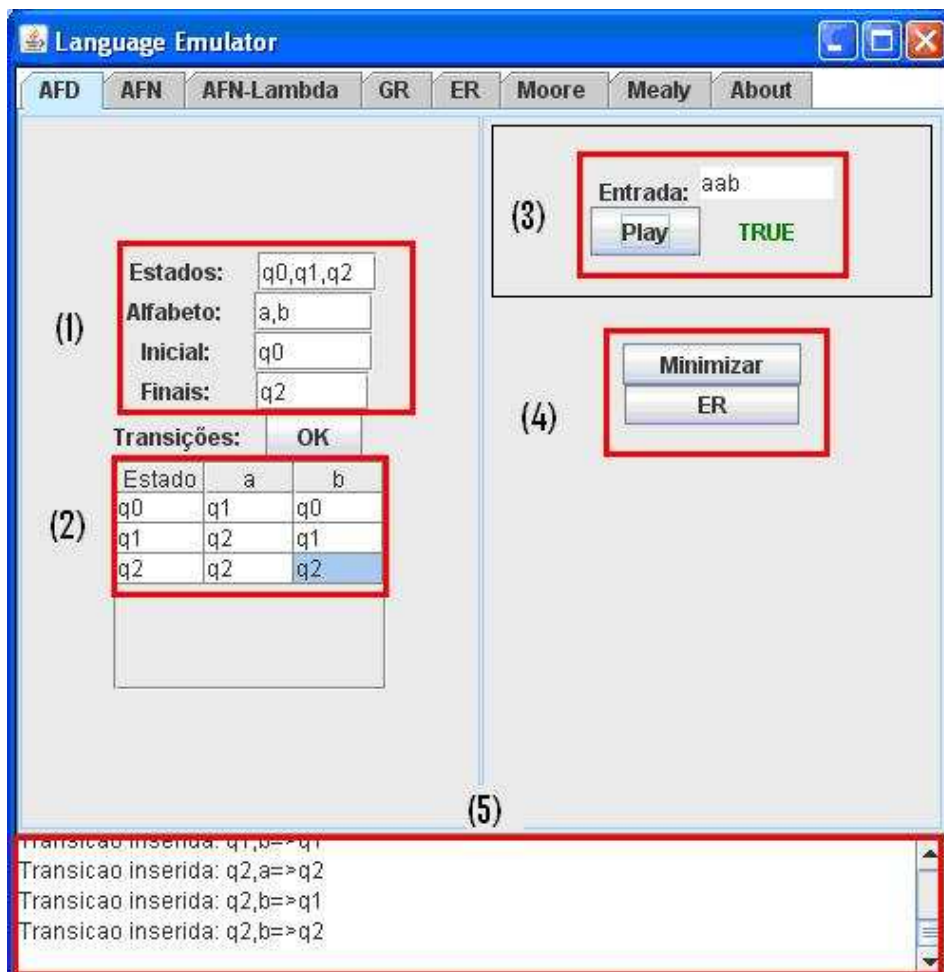


Figura 2.1 – Editor para construção de AFDs.

A Figura 2.1 apresenta a tela principal desta ferramenta dividida em setores numerados de 1 a 5 que significam:

- (1) permite que o usuário, possivelmente o aluno, insira os elementos correspondentes da definição formal do modelo computacional AFD, a saber: o conjunto de estados (“Estados”), o alfabeto (“Alfabeto”), o estado inicial (“Inicial”) e o conjunto de estados finais (“Finais”). Após inserir estas informações o usuário clica no botão “OK” para começar a inserir as transições do modelo.
- (2) solicita ao usuário o preenchimento da tabela com as transições do AFD. Por exemplo, estando o AFD no estado “q0” e ao ler o símbolo “a” na fita de entrada o estado corrente do modelo passa a ser “q1”.
- (3) refere-se aos testes que o usuário pode fazer sob o AFD construído por ele. Inserindo uma palavra qualquer, por exemplo, “aab” e clicando-se no botão “Play” o software processa o teste e informa “TRUE” ou “FALSE” se o modelo computacional reconheceu a palavra ou não respectivamente.
- (4) proporciona ao usuário a conversão do modelo computacional AFD para seus respectivos modelos equivalentes, neste caso para o AFD Mínimo (“Minimizar”) e Expressão Regular (“ER”).
- (5) exhibe as ações realizadas pelo usuário sob a ferramenta.

Além do Language Emulator (VIEIRA, J. F. M.; VIEIRA, M. A. M. e VIEIRA, N. J., 2002) há outra ferramenta chamada SCTMF (COSTA, MENESES e UBER, 2008) que pode auxiliar também os alunos em sua dificuldade de definir formalmente um modelo computacional. O diferencial desta em relação àquela está no processo de interação do usuário com o sistema em que são usadas interfaces separadas para definir: o alfabeto, o conjunto de estados, as transições e uma última interface para testes com palavras fornecidas pelo usuário (palavras de teste). A Figura 2.2 apresenta a “última” interface para realizar testes com palavra (“1111”) no SCTMF.

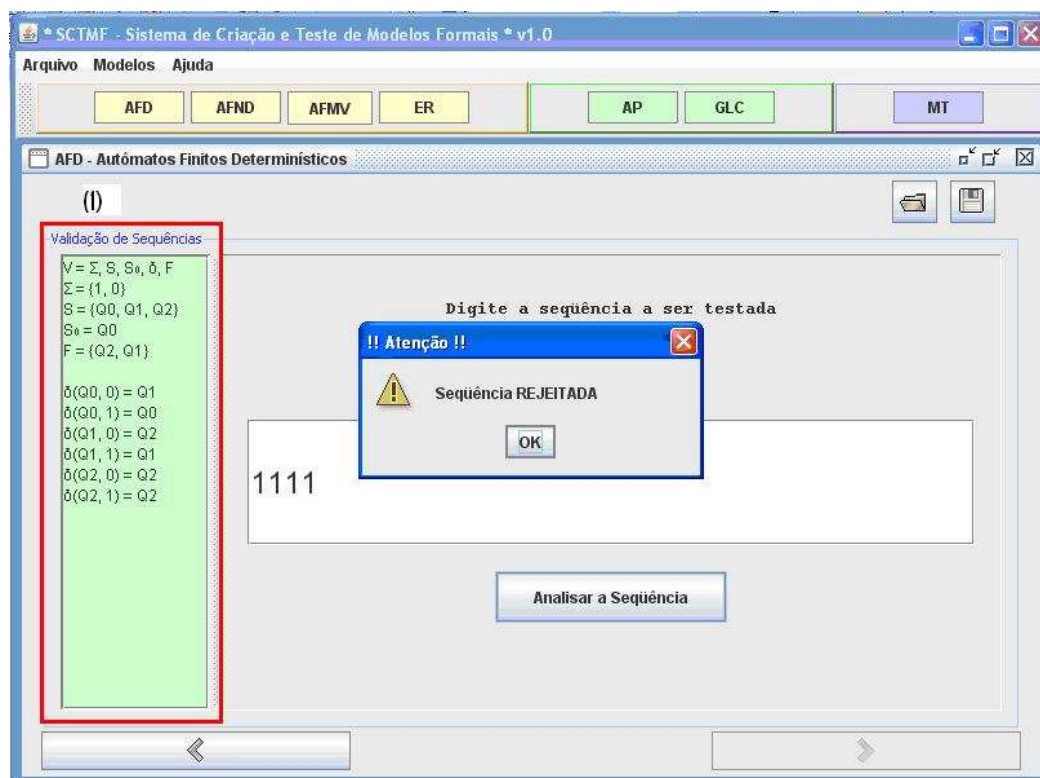


Figura 2.2 – Reconhecimento de palavras no SCTMF.

A partir de definições inseridas em etapas prévias, o modelo formal vai sendo construído, podendo ser visto na parte marcada em vermelho e rotulada com (1) à esquerda (Figura 2.2).

2.1.5.2 Compreensão sobre o processo de construção de autômatos e reconhecimento de linguagens

Os trabalhos de Chesñear et al. (2004), Chua e Winton (1987), Goyal e Sachdeva (2009) e Krishnan (2002) apontam a existência de dificuldades inerentes aos próprios fundamentos de TC. CHUA e WINTON (1987, p. 71) afirmam que “(...) Students generally have difficulty constructing automata and verifying their functionality”³. Algumas ferramentas computacionais disponíveis na literatura para ajudar a superar essas dificuldades são: GAM (JUKEMURA, NASCIMENTO e UCHÔA, 2005), VAS (BOVET, 2006) e JFLAP (RODGER, 2010).

³ Tradução do autor: “(...) Alunos geralmente têm dificuldades em construir autômatos e verificar suas funcionalidades”.

A ferramenta computacional GAM (JUKEMURA, NASCIMENTO e UCHÔA, 2005), por exemplo, tem seu domínio restrito ao estudo dos modelos computacionais AFD e AFND e sua principal vantagem é a utilização de interface gráfica que permite ao usuário visualizar o diagrama do modelo computacional e manipulá-lo. A tela principal desta ferramenta computacional pode ser vista na Figura 2.3.

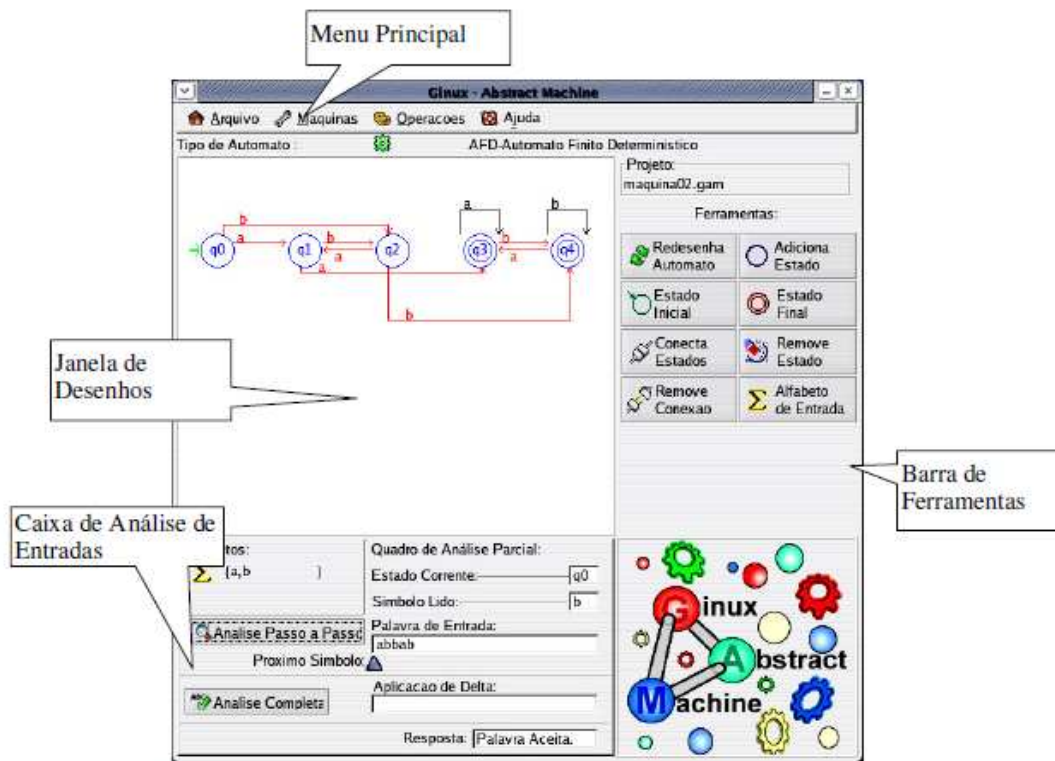


Figura 2.3 – Tela principal de GAM (JUKEMURA, NASCIMENTO e UCHÔA, 2005).

Além disso, GAM permite o estudo de conversão de AFND para AFD e ainda conta com o processo de minimização deste último. No entanto, não permite ao aluno saber rapidamente se o AFD ou AFND construído como solução a um exercício é realmente a solução correta. O aluno precisa concluir que sua resposta está correta ou não por meio de palavras de teste inseridas por ele na ferramenta e testadas sobre o AFD ou AFND construído.

Outra ferramenta que pode ser indicada por professores de TC a seus alunos é a VAS (BOVET, 2006). Ela permite o estudo de um modelo computacional a mais do que a GAM: Máquinas de Turing (MT). Suas interfaces gráficas permitem ao usuário visualizar e manipular estados e transições diretamente, enquanto se criam AFD, AFND ou MT como solução a um exercício. A VAS, embora permita a conversão de AFND em AFD, não executa o processo de minimização de AFDs.

O ponto forte de VAS (BOVET, 2006) é sua manipulação de MTs: a interface permite uma manipulação diferente das demais ferramentas, pois possui botões que facilitam o processo de ida e volta realizada pela cabeça de leitura sobre a fita no processamento da máquina e uma visualização da fita após o processo de reconhecimento ter sido completado. Na Figura 2.4 a marcação em vermelho rotulada como (1) representa o uso do botão do menu que retorna a cabeça de leitura da MT até o símbolo “#”.

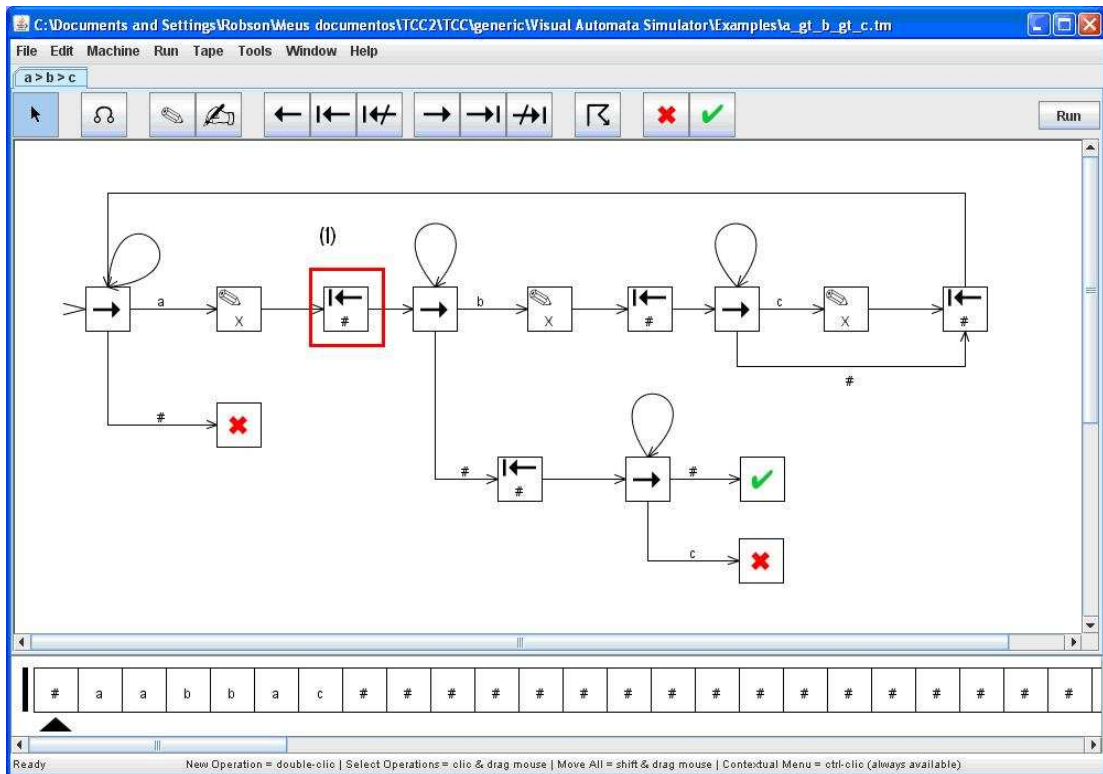


Figura 2.4 – Interface de MT no VAS.

Uma terceira ferramenta que pode ser indicada a alunos com dificuldades em construir autômatos e verificar seu processo de reconhecimento de palavras é JFLAP (RODGER, 2010). Dentre todas as ferramentas computacionais apresentadas até então neste trabalho, ela é a mais completa. Além de implementar AFDs e AFNDs como todas as outras ferramentas, também permite o estudo de MT com múltiplas fitas, Gramáticas e ERs e mais recentemente Máquinas de Moore e Máquinas de Mealy.

Os pontos fortes de JFLAP (RODGER, 2010) são: o processo de reconhecimento de palavras passo a passo (o aluno acompanha todos os passos envolvidos no reconhecimento de uma palavra pelo modelo computacional desenhado, em vez de simplesmente ser informado que a palavra foi aceita ou rejeitada), a variedade implementada de conversões de modelos computacionais em seus equivalentes e a possibilidade de visualizar e manipular diretamente os estados e as transições dos modelos computacionais criados. A Figura 2.5 exibe a tela para a construção do modelo computacional AFND no JFLAP.

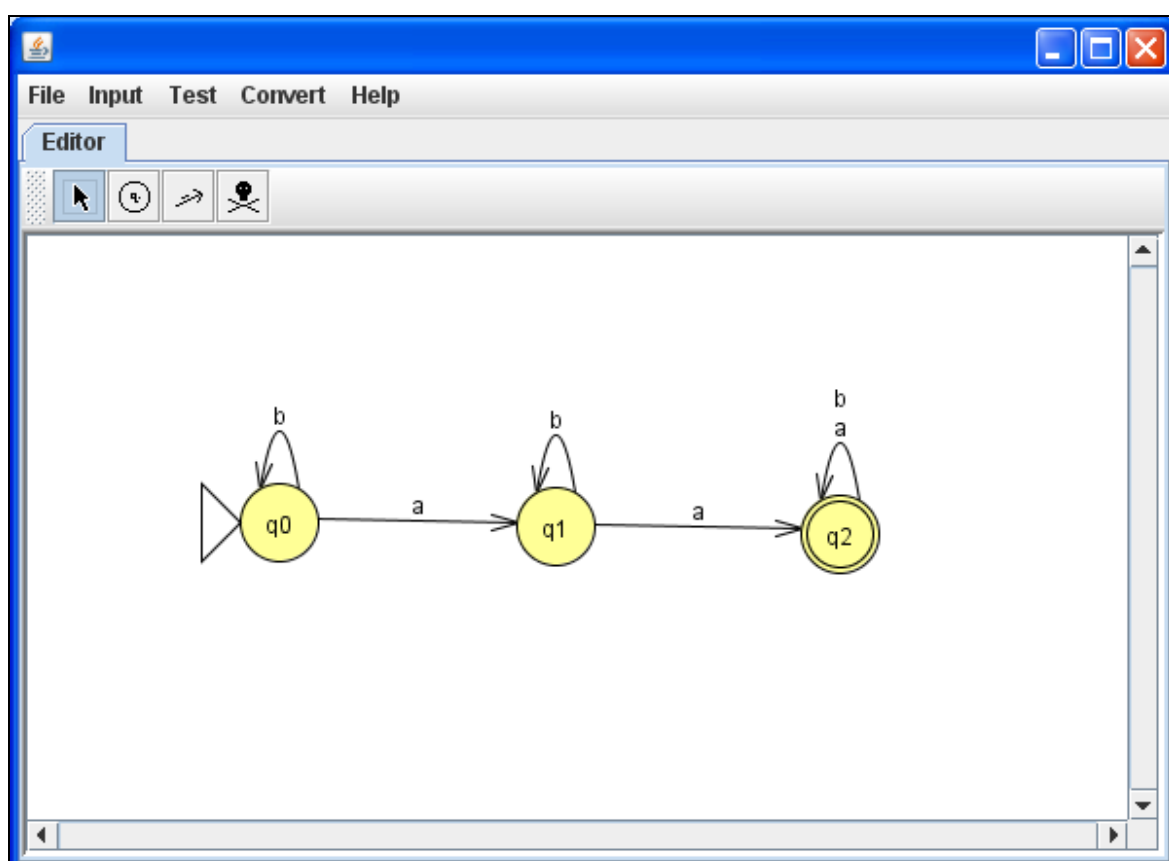






Figura 2.5 – Editor para AFNDs no JFLAP.

Os ícones disponíveis na barra de ferramentas de JFLAP permitem:

-  : movimentar os estados criados ou suas transições; e definir, com o botão direito do mouse, se um estado é final e/ou inicial.
-  : criar estados bastando selecionar esta opção e clicar sob a área com o botão esquerdo do mouse.
-  : criar transições de um estado para outro.

-  : apagar estados e transições também selecionando e clicando com o botão esquerdo sob o estado que se deseja excluir.

2.2 Ambientes Computacionais para Educação a Distância (EAD)

Segundo Rocha (2000) a “(...) característica principal de **Educação a Distância** é o estabelecimento de uma comunicação de dupla-via, na qual o professor e o aluno não se encontram juntos no mesmo espaço físico, necessitando de meios que possibilitem a **comunicação** entre ambos.”.

Neste contexto, a rede mundial de computadores, *Internet*, pode ser vista como um meio que permite a comunicação em dupla-via entre dois ou mais pontos dela. É nisto que se baseia tecnicamente a construção de cursos a distância na Web em que professores e alunos estão geograficamente separados, mas necessitam se comunicarem, bem como compartilharem materiais de aula, apostilas, listas de exercícios, recursos de áudio e vídeo etc.

Assim, surgiram vários ambientes computacionais para EAD possibilitando ao professor:

- inserir e disponibilizar materiais de aula a seus alunos;
- elaborar métodos de ensino conforme suas turmas;
- usufruir de modelos básicos para condução de uma disciplina; e
- utilizar diversas ferramentas de interação e gerenciamento (ROCHA, 2000).

A respeito destas ferramentas, Rocha (2000) as divide em três tipos: *comunicação*, *edição do conteúdo* e *administração* como podem ser vistas na Figura 2.6.

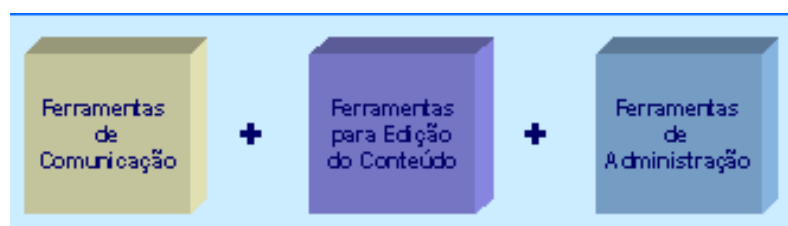


Figura 2.6 – Ferramentas de Cursos a Distância. (ROCHA, 2000).

As *ferramentas de comunicação* compreendem todos os mecanismos implementados no ambiente EAD que possibilitam a comunicação entre seus usuários. Assim, são exemplos destas ferramentas: o correio, o fórum, o *blog*, o *chat* etc.

As *ferramentas para edição do conteúdo* se caracterizam por permitir a inserção, a retirada, o cancelamento, a disponibilização de vários materiais do professor a seus alunos, bem como a disposição visual destes materiais. Ferramentas de manipulação de conteúdo por meio de *tags* em HTML, existentes no Moodle são bons exemplos.

As *ferramentas de administração* tratam do gerenciamento de usuários e de cursos. Um exemplo deste tipo de ferramenta existente em vários ambientes é a que permite a criação de cursos, no qual geralmente se indica o nome do curso e de seu responsável, contato por *email*, qual o número máximo de alunos, se é público ou privado etc.

Atualmente, vários ambientes computacionais EAD estão disponíveis para uso, como: Amadeus (AMADEUS, 2010), AulaNet (AULANET, 2010), e-ProInfo (EPROINFO, 2010), LearningSpace (LEARNINGSPACE, 2010), Moodle (MOODLE, 2010), TelEduc (TELEDUC, 2010), WebCT (WEBCT, 2010) e outros. Dentre estes, o Moodle é um ambiente computacional de código livre, que pode ser encontrado para *download* na *Internet*, utiliza BD MySQL (Software livre) e vem sendo utilizado por muitas universidades nacionais e internacionais como suporte à educação presencial para disponibilizar arquivos, enviar *emails* etc.

Ao mesmo tempo em que se tem disponível esta variedade de ambientes EAD, o professor que ministra uma disciplina em específico sente a falta de funcionalidades adicionais em tais ambientes para auxiliá-lo em sua tarefa de ensino. Um exemplo disso seria a integração destes ambientes EAD com ferramentas computacionais isoladas como as vistas nas Subseções 2.1.5.1 e 2.1.5.2. Assim, neste trabalho buscou-se integrar no ambiente EAD Moodle a ferramenta computacional JFLAP para ajudar professores e alunos nos estudos de TC.

2.3 Considerações do capítulo

Mesmo se aplicando algumas estratégias disponíveis na literatura para auxiliar o professor a ensinar e os alunos a aprender fundamentos de TC, na Universidade Federal do Pará em seu curso de Bacharelado em Ciência da Computação há um percentual de reprovação em torno de 30% na disciplina de Linguagens Formais.

No entanto, diferentemente do que se poderia pensar, não é um problema de um professor em específico. Isto pode ser observado na Figura 2.7 em que se apresentam os índices de reprovação na disciplina de Linguagens Formais durante seis semestres (do 2º Semestre de 2005 ao 2º Semestre de 2009) ministrada por três professores diferentes. As quatro primeiras barras (2005/2, 2006/1, 2006/2 e 2007/2) são relativas ao índice de reprovação na disciplina de Linguagens Formais ministrada pelo professor A. Esta mesma disciplina no segundo semestre de 2008 (2008/2) fôra ministrada pelo professor B e a última barra (2009/2) corresponde a percentagem de reprovação na disciplina de Linguagens Formais ministrada pelo professor C.

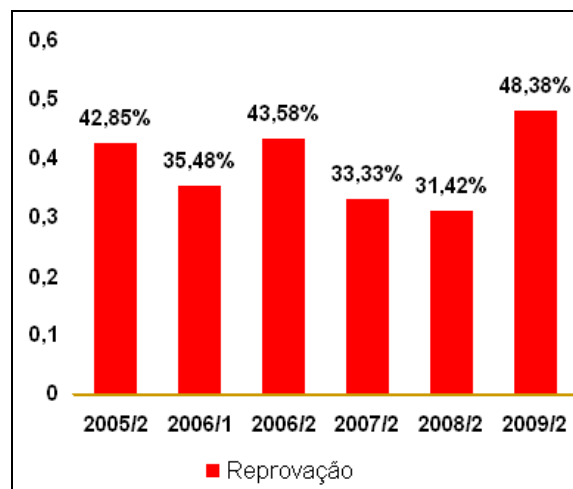


Figura 2.7 – Reprovação em LF na UFPA.

Portanto, este problema de alto índice de reprovação em Linguagens Formais na UFPA pode ter várias origens como:

- o conjunto de estratégias utilizadas pelos professores,
- as dificuldades dos alunos em assimilar os fundamentos de TC,
- a dificuldade do professor em acompanhar a aprendizagem de todos os seus alunos, e
- a dificuldade que os alunos têm em saber rapidamente se sua solução a um problema proposto em listas pelo professor é a solução esperada por este ou não. Geralmente se tem que responder a todas as questões de uma lista de exercícios e depois levar todas as suas soluções para o professor ou o monitor corrigi-las.

A respeito da última dificuldade, os trabalhos da literatura consultados nesta dissertação não ajudam no problema enfrentado pelos alunos da UFPA em saber se sua solução está correta ou não imediatamente. As ferramentas pesquisadas não contemplam esta necessidade. É o aluno quem tem que concluir, a partir de vários testes sobre a solução construída por ele, que sua solução está correta ou não.

Acerca dos problemas citados, buscou-se neste trabalho fazer a integração do ambiente Moodle com a ferramenta JFLAP. O Moodle tem diversas facilidades que são utilizadas por várias disciplinas na UFPA, como suas ferramentas *de comunicação, para edição do conteúdo e de administração*, e tem a vantagem de ser possível o seu download e a modificação de seu código-fonte. Já o JFLAP (RODGER, 2010) tem um grande número de modelos e conversões entre modelos equivalentes implementados.

O ambiente Moodle foi adaptado para suportar novos tipos de questões específicos de TC em que suas soluções são modelos computacionais desenhados nos editores do JFLAP, permitindo o acompanhamento da aprendizagem dos alunos pelo professor e, auxiliando os alunos a saberem rapidamente se suas soluções são as esperadas pelo professor ou não. O Capítulo 3 trata desta adaptação.

CAPÍTULO 3

ADAPTAÇÃO DO AMBIENTE MOODLE PARA SUPORTE AO ENSINO E À APRENDIZAGEM DE TC

Nenhum dos trabalhos apresentados no Capítulo 2 propõe estratégia(s) que permita(m) ao professor acompanhar a aprendizagem de seus alunos e também não propõem mecanismos de correção automático de questões específicas de TC com um *feedback* rápido, que possibilitem ao aluno saber o quanto antes se sua solução é a solução esperada pelo professor ou não.

As Seções 3.1 e 3.2 a seguir se destinam a apresentar as adaptações realizadas no Moodle (MOODLE, 2010) na busca de facilitar o ensino e a aprendizagem de TC. Na Seção 3.1 serão apresentadas as facilidades implementadas para o professor e, logo após, na Seção 3.2 as facilidades implementadas para o aluno. A Seção 3.3 apresenta uma visão geral da integração entre Moodle e JFLAP. Por último, a Seção 3.4 apresenta os materiais inseridos no Moodle que estão disponíveis ao professor, em especial para o professor principiante, e ao aluno.

3.1 Facilidades implementadas para o professor

Os questionários online do ambiente computacional Moodle possibilitam ao professor a inserção de questões de tipos variados em que algumas delas podem ser corrigidas automaticamente (questões do tipo Verdadeiro/Falso) e outras não (questões do tipo Dissertação). Assim, a vantagem desses questionários em relação às tradicionais listas de exercícios é a redução do tempo empreendido pelo professor para corrigir a lista de cada um de seus alunos, desde que os tipos das questões contidas neles permitam correção automática de suas respostas.

A respeito destas correções para questões específicas de TC cabe o esclarecimento da necessidade em se ter estas duas possibilidades de correção:

1. A **correção automática** existe para os modelos computacionais para os quais há um algoritmo que verifique a equivalência entre eles. Assim, os modelos computacionais Autômato Finito Determinístico e Não-Determinístico, Gramática Regular e Expressão Regular que possuem equivalência entre máquinas são corrigidos automaticamente quando implementados no Moodle;
2. Os modelos computacionais Autômato Finito com Pilha e Máquinas de Turing, por não possuírem equivalência entre máquinas são corrigidos de maneira **semi-automática**.

A partir desses esclarecimentos, a seguir serão apresentados os recursos implementados no ambiente Moodle para o usuário professor. As interfaces de criação de questões específicas nesse ambiente foram implementadas de maneiras muito semelhantes, elas se diferenciam apenas no editor do JFLAP (RODGER, 2010) usado para inserção da solução da questão. Logo, a título de ilustração, serão apresentadas somente as telas para criação de uma questão do tipo Autômato Finito Determinístico e Não-Determinístico. As telas para criação dos outros tipos de questões específicas podem ser encontradas no Apêndice A deste trabalho. A Figura 3.1 apresenta a tela do Moodle para criação de uma nova pergunta em seu banco de questões. Nela é chamada a atenção para as questões, envoltas pelo retângulo com bordas em vermelho, que foram implementadas.

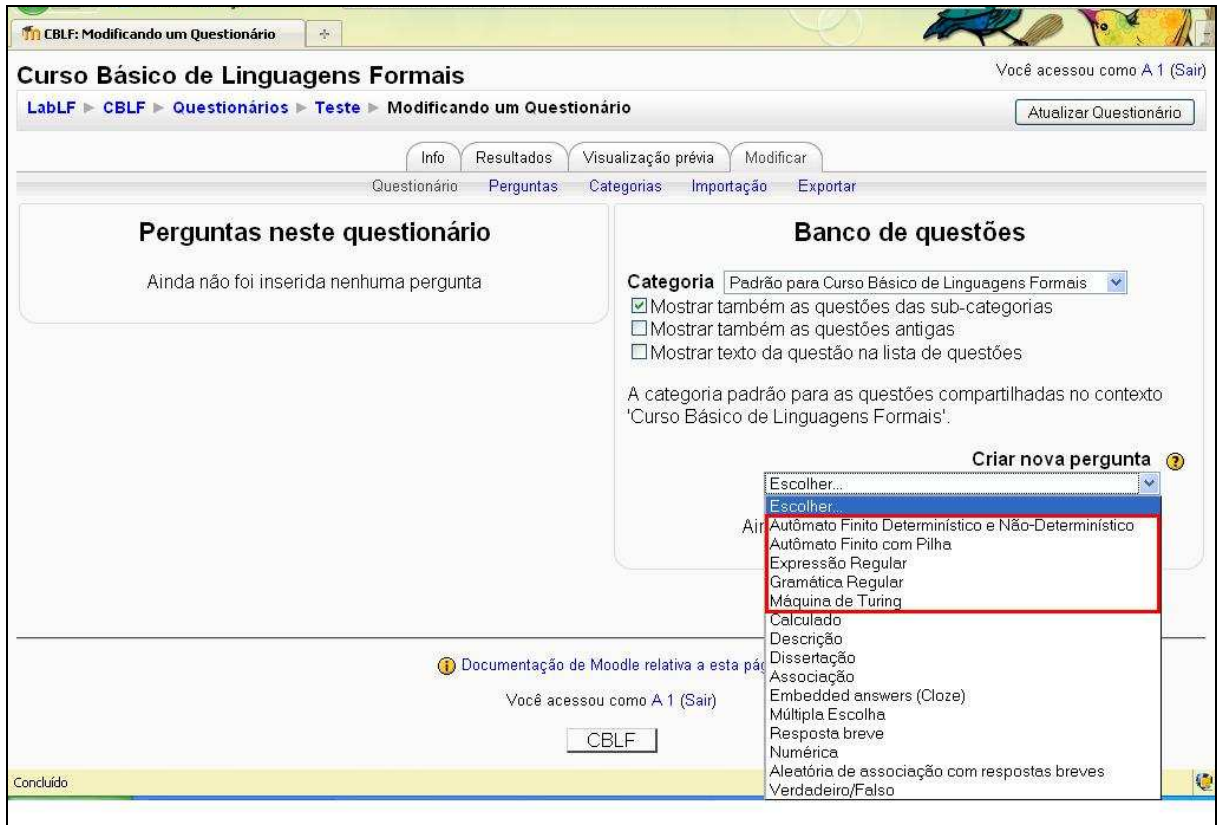


Figura 3.1 – Questões específicas de TC implementadas.

Ao se escolher criar uma nova pergunta de Autômato Finito Determinístico e Não-Determinístico selecionando a primeira opção do *select* expandido na Figura 3.1, a tela da Figura 3.2 é exibida ao professor para que possa preencher as informações necessárias. As regiões envoltas pelos retângulos com bordas vermelhas devem necessariamente ser preenchidas com:

- o enunciado da questão em (1);
- o diagrama do modelo computacional (resposta da questão) em (2); e
- palavras de teste necessárias para o processo de correção em (3).

Curso Básico de Linguagens Formais Você acessou como A 1 (Sair)

LabLF > Curso Básico de Linguagens Formais > Questionários > Anterior > Modificando um Questionário > Editando uma pergunta de autômato finito determinístico

Adicionando uma questão de Autômato Finito Determinístico

Geral

Categoria: Padrão para Curso Básico de Linguagens Formais

Nome da pergunta*: Pergunta 1

Texto da pergunta

(1) Construa um AFND que aceite todas as palavras sob o alfabeto $\{0,1\}$ terminadas em 1.

Formato: Formato HTML


Avaliação predefinida*: 1

Fator de penalidade*: 0.1

Feedback geral:

Resposta correta. Preencha a resposta a seguir.

Resposta

(2) 

(3) Palavras de teste*: 0, 1, 10, 11, 100, 101, 1000, 1001, 0001, 10100, 1010, 01010101

Este form contém campos obrigatórios

Documentação de Moodle relativa a esta página
Você acessou como A 1 (Sair)

Figura 3.2 – Tela para criação de uma questão do tipo AFD/AFND.

Estas palavras de teste devem ser inseridas para permitir que o processo de correção seja efetivo, pois tanto a solução do professor como a do aluno são testadas com elas. Assim, as que forem aceitas pela solução do professor devem também ser aceitas pela resposta do aluno, bem como as palavras rejeitadas pela solução do professor devem ser rejeitadas pela solução do aluno. Portanto é a partir das palavras aceitas e rejeitadas que se atribui um percentual de acerto da questão. Uma sugestão é inserir pelo menos cinco palavras a serem aceitas pela solução e outras cinco que sejam rejeitadas, pois o percentual de acerto é dependente do número de palavras de teste inseridas pelo professor.

Desta forma, com as questões específicas de TC implementadas o professor pode usufruir das funcionalidades de acompanhamento de atividades disponíveis no Moodle (MOODLE, 2010) e saber, prontamente, se seus alunos responderam as questões específicas propostas por ele em um questionário.

A Figura 3.3 exibe a tela de acompanhamento customizada das atividades realizadas pelos participantes da disciplina, disponível na seção “Administração - Relatórios” do Moodle. Nessa parte do sistema, o usuário pode selecionar as informações que deseja visualizar por meio de filtros que são aplicados pela escolha de:

- uma disciplina (1),
- participantes da disciplina (2),
- um determinado dia escolhido (3),
- atividades propostas (4),
- ações realizadas pelos alunos (5) e
- downloads realizados pelos alunos (6).

Curso Básico de Linguagens Formais

Você acessou como A 1 (Sair)

LabLF > CBLF > Relatórios > Logs > B 1, quinta-feira, 26 agosto 2010

Curso Básico de Linguagens Formais: B 1, quinta-feira, 26 agosto 2010 (UTC-3)

(1) Curso Básico de Linguagens Formais (2) B 1 (3) Hoje, 26 agosto 2010

(4) Teste (5) Mostrando 10 registros (6)

Hora	Nome completo	Ação
qui 26 agosto 2010, 16:31	B 1	visualização da questão
qui 26 agosto 2010, 16:31	B 1	revisão de questão
qui 26 agosto 2010, 16:31	B 1	revisão de questão
qui 26 agosto 2010, 16:31	B 1	continuar tentativa na questão
qui 26 agosto 2010, 16:31	B 1	encerramento de tentativa na questão
qui 26 agosto 2010, 16:31	B 1	continuar tentativa na questão
qui 26 agosto 2010, 16:30	B 1	continuar tentativa na questão
qui 26 agosto 2010, 16:30	B 1	continuar tentativa na questão
qui 26 agosto 2010, 16:29	B 1	tentativa na questão
qui 26 agosto 2010, 16:29	B 1	visualização da questão

Documentação de Moodle relativa a esta página

Você acessou como A 1 (Sair)

CBLF

Figura 3.3 – Tela de acompanhamento de atividades realizadas.

Ainda nessa tela da Figura 3.3, a linha contornada pelo retângulo com bordas vermelhas exhibe a informação que o usuário “B 1” realizou tentativa de responder uma questão do questionário “Teste” no dia 21 de Abril de 2010. A Figura 3.4 apresenta a tela que se segue ao se clicar em “tentativa na questão” nesta linha.

Na tela ilustrada na Figura 3.4 é possível verificar a trajetória do aluno até a construção da solução esperada pelo professor. Por esta interface é permitido ao professor saber que o aluno realizou duas tentativas⁴, como pode ser visto em (B), em que a primeira tentativa iniciada por “1” resultou na geração de um *feedback* (visto em (A)) ao aluno informando que ele ainda não alcançara a solução esperada pelo professor.

⁴ Duas tentativas e uma confirmação.

Pergunta 1

B 1
Questionário: Teste
Completado em: quinta-feira, 26 agosto 2010, 17:52

1 Construa um AFND que aceite todas as palavras sob o alfabeto {0,1} terminadas em 1.

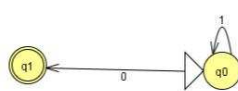
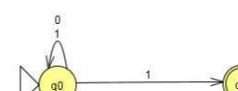

Acertos: 1

Você acertou **33.33333333333333%** da questão.
 Sua resposta ainda não é a solução esperada pelo professor.
 O autômato finito deveria reconhecer a(s) palavra(s): **1, 11, 101, 1001, 0001, 01010101,**
 O autômato finito **não** deveria reconhecer a(s) palavra(s): **0, 10,**

Faça um comentário ou modifique a avaliação

Parcialmente correta

Notas relativas a este envio: 0.33/1. Considerando as penalidades: 0/1. Você não foi penalizado em relação a este envio

Histórico das respostas		Hora	Classificação	Avaliar	
#	Ação	Resposta			
1	Nota		17:52:33 em 26/08/10	0.33	0
2	Nota		17:52:35 em 26/08/10	1	1
3	Fechar Avaliação		17:53:55 em 26/08/10	1	1

(B)

Documentação de Moodle relativa a esta página
 Você acessou como A.1 (Sair)
[Home Page](#)

Figura 3.4 – A solução submetida ainda não é a esperada pelo professor.

Ao clicar no *link* “2”, visto na segunda linha em (B), o professor é direcionado à segunda tentativa do aluno. A Figura 3.5 apresenta a tela exibida para esta tentativa e mostra que o aluno solucionou o problema como esperado pelo professor. A terceira linha indicada pelo *link* “3” é a confirmação da última resposta do aluno.

Pergunta 1

B 1

Questionário: Teste

Completado em: quinta-feira, 26 agosto 2010, 17:52

1 Construa um AFND que aceite todas as palavras sob o alfabeto $\{0,1\}$ terminadas em 1.

Acertos: 1

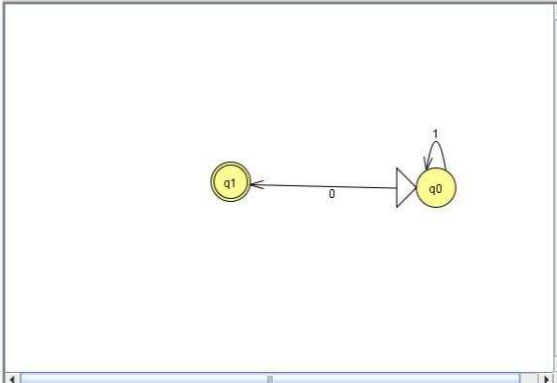
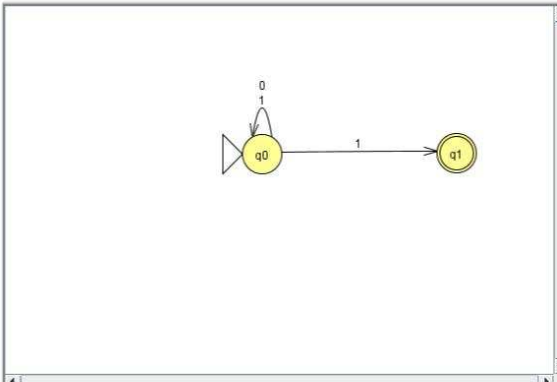
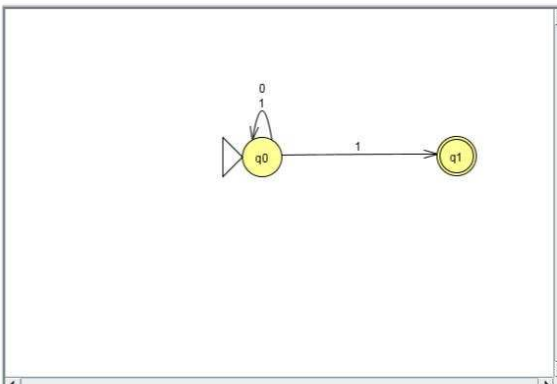
Resposta Correta. Você acertou **100%** da questão. ✓

Faça um comentário ou modifique a avaliação

Correto

Notas relativas a este envio: 1/1.

Histórico das respostas

#	Ação	Resposta	Hora	Classificação	Avaliar
1	Nota		17:52:33 em 26/08/10	0,33	0
2	Nota		17:52:35 em 26/08/10	1	1
3	Fechar Avaliação		17:53:55 em 26/08/10	1	1

[Documentação de Moodle relativa a esta página](#)
 Você acessou como A 1 (Sair)
[Home Page](#)

Figura 3.5 – A solução submetida é a esperada pelo professor.

Por último, para exemplificar a possibilidade de erro da solução em sua totalidade a Figura 3.6 apresenta a tela que é exibida ao aluno. Nela se pode verificar que o *feedback* informa ao aluno que sua resposta não é a solução esperada pelo professor e apresenta palavras que sua solução deveria reconhecer, como “1, 11, 10001, 00001, 0000101”, e algumas que não deveria reconhecer, como “0, 10, 1000, 10100, 1010”.

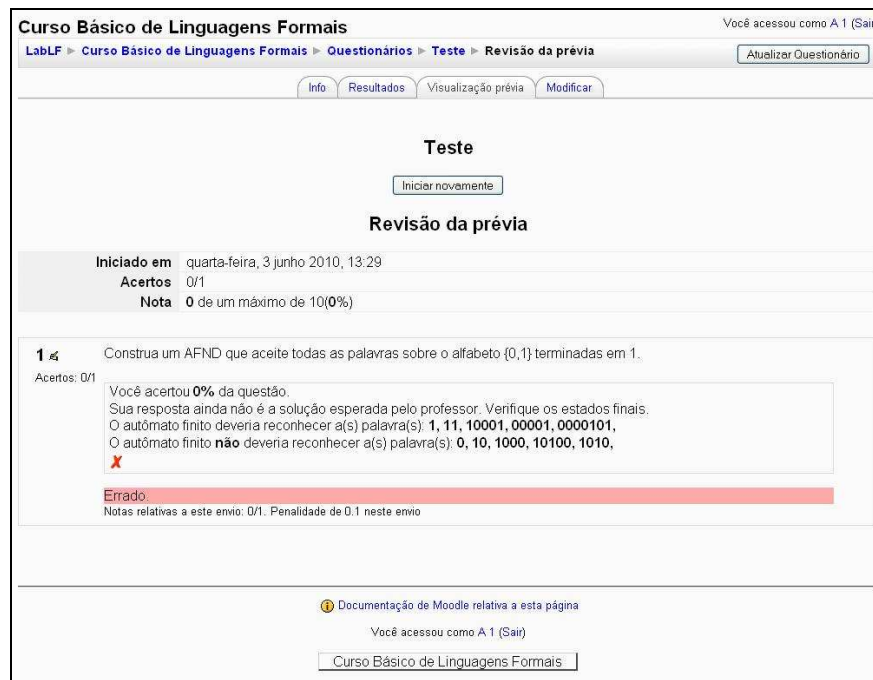


Figura 3.6 – A solução submetida completamente errada.

3.2 Facilidades implementadas para o aluno

Dando seqüência às implementações realizadas no Moodle, agora para o aluno, a Figura 3.7 apresenta a interface para construção do modelo computacional Autômato Finito Determinístico e Não-Determinístico como resposta a uma questão proposta pelo professor. Por meio desta interface o aluno ao tentar responder uma pergunta específica do questionário tem em:

- (1) o enunciado do problema proposto pelo professor,
- (2) um editor correspondente do JFLAP ao tipo de questão específica (neste exemplo aparece o editor para construir o modelo computacional AFD/AFND) e
- (3) um botão para submissão de solução.

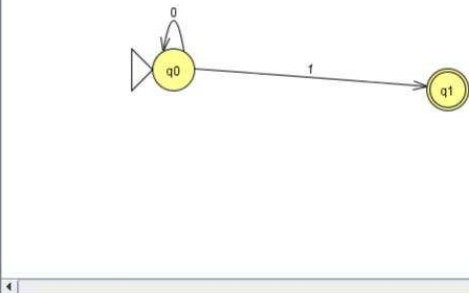
Curso Básico de Linguagens Formais Você acessou como B 1 (Sair)

LabLF > Curso Básico de Linguagens Formais > Questionários

Respondendo Pergunta 1

(1) Construa um AFND que aceite todas as palavras sob o alfabeto $\{0,1\}$ terminadas em 1.

Resposta

(2) 

(3)

[Documentação de Moodle relativa a esta página](#)

Você acessou como B 1 (Sair)

Figura 3.7 – Tela para tentativa de solução de questão pelo aluno.

Supondo-se agora que o aluno tenha inserido sua resposta para a questão da Figura 3.7 “Construa um AFND que aceite todas as palavras sobre o alfabeto $\{0,1\}$ terminadas em 1.”, ao submetê-la à correção⁵ clicando em “Submeter”, um dentre três *feedbacks* é exibido ao aluno informando que sua solução:

- ainda não é a solução esperada pelo professor estando:
 - completamente incorreta ou
 - parcialmente correta.
- é a solução esperada pelo professor.

⁵ Neste caso a correção é automática, ou seja, é encontrado um autômato determinístico mínimo internamente e comparado com o respectivo autômato mínimo da solução pré-cadastrada pelo professor.

A Figura 3.8 apresenta o *feedback* exibido ao aluno quando ele não alcançara a resposta esperada pelo professor, pois sua resposta está completamente incorreta. Além desta informação, o *feedback* apresenta possíveis palavras que a solução do aluno deveria aceitar ou não aceitar. Neste exemplo da Figura 3.8 fôra exibido ao aluno que sua solução deveria reconhecer as palavras “1, 11, 10001, 00001, 0000101” e não deveria reconhecer as palavras “0, 10, 1000, 10100, 1010”.

Curso Básico de Linguagens Formais Você acessou como B 2 (Sair)

LabLF > Curso Básico de Linguagens Formais > Questionários > Teste > Tentativa 1

Teste - Tentativa 1

1 Construa um AFND que aceite todas as palavras sob o alfabeto $\{0,1\}$ terminadas em 1.

Acertos: 0/1

Você acertou **0%** da questão.
 Sua resposta ainda não é a solução esperada pelo professor. Verifique os estados finais.
 O autômato finito deveria reconhecer a(s) palavra(s): 1, 11, 10001, 00001, 0000101,
 O autômato finito **não** deveria reconhecer a(s) palavra(s): 0, 10, 1000, 10100, 1010,
 X

Responder e enviar

Errado
 Notas relativas a este envio: 0/1. Penalidade de 0.1 neste envio.

Salvar sem enviar Enviar página Enviar tudo e terminar

Você acessou como B 2 (Sair)

Curso Básico de Linguagens Formais

Figura 3.8 – Tela de feedback de solução completamente incorreta.

Assim, supõe-se que com estas informações o aluno modifique sua resposta a fim de, novamente, tentar fazer a questão e acertá-la. A Figura 3.9 apresenta a informação exibida ao aluno quando este submete uma resposta parcialmente correta.

The screenshot shows a web interface for a course titled "Curso Básico de Linguagens Formais". The user is logged in as "B 2". The page is titled "Teste - Tentativa 1". The question number is "1" and asks to construct an AFND that accepts all words over the alphabet {0,1} ending in 1. The user's score is 0.9/1. The feedback text states: "Você acertou 70% da questão. Sua resposta ainda não é a solução esperada pelo professor. Verifique os estados finais. O autômato finito deveria reconhecer a(s) palavra(s): 11, 10001, 0000101,". A yellow bar highlights the text "Resposta parcialmente correta." Below this, it says "Notas relativas a este envio: 0/1. Penalidade de 0.1 neste envio". There are buttons for "Responder e enviar", "Salvar sem enviar", "Enviar página", and "Enviar tudo e terminar".

Figura 3.9 – Tela de feedback de solução correta.

A tela da Figura 3.10 é apresentada ao aluno quando ele submete a solução esperada pelo professor.

The screenshot shows the same web interface as Figure 3.9. The question is the same. The user's score is 0.9/1. The feedback text states: "Resposta Correta. Você acertou 100% da questão!!!". A green bar highlights the text "Correto". Below this, it says "Notas relativas a este envio: 1/1. Considerando as penalidades: 0.9/1.". There are buttons for "Responder e enviar", "Salvar sem enviar", "Enviar página", and "Enviar tudo e terminar".

Figura 3.10 – Tela de feedback de solução correta.

Supondo agora que a questão disponibilizada ao aluno seja do tipo Autômato Finito com Pilha. A Figura 3.11 apresenta a tela exibida ao aluno.

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

LabLF > Linguagens Formais 2/2009 > Questionários

Respondendo Questao 4

Feedback:

Construa um Auômato Finito com Pilha sob o alfabeto $\{0,1\}$ que reconheça todas as palavras terminadas em 0.

Resposta

```

graph LR
    q0((q0)) -- "0, A; A" --> q0
    q0 -- "1, A; A" --> q1((q1))
    style q1 fill:#ffff00
  
```

[Documentação de Moodle relativa a esta página](#)
 Você acessou como Robson Aguiar (Sair)

Figura 3.11 – Tela para tentativa de solução de questão de AFP pelo aluno.

Note que a disposição do enunciado, o editor e os botões é idêntica à questão anterior sobre Autômato Finito Determinístico e Não-Determinístico, porém existe uma diferença entre os processos de correção para estes dois tipos de questões. O processo de correção para questões de Autômato Finito Determinístico e Não-determinístico é chamado automático e o mecanismo de correção de questões do tipo Autômato Finito com Pilha é chamado semi-automático.

Esta diferença nos processos de correção decorre da possibilidade de comparar Autômatos Finitos Determinísticos mínimos construídos internamente e de maneira transparente (quando o professor cadastra sua solução e quando o aluno elabora uma resposta) ao usuário. Assim, as questões dos tipos Autômato Finito com Pilha e Máquinas de Turing por não possuírem um mecanismo implementado de obtenção do Autômato Finito Determinístico mínimo são corrigidas de maneira semi-automática.

Esta correção semi-automática leva em conta exclusivamente a solução dada pelo professor e a solução submetida pelo aluno. Caso estas soluções não coincidam, uma porcentagem de acerto será informada como *feedback* ao aluno, porcentagem esta calculada através das palavras de teste informadas pelo professor no ato do cadastramento da questão.

Devido as interfaces para tentativa de solução das questões específicas de TC: GR, ER e MT serem semelhantes às interfaces de questões do tipo AFD apresentadas a pouco, diferenciando-se apenas no editor para inserção da resposta pelo aluno, elas não serão discutidas nesta seção. O leitor caso tenha interesse, pode visualizá-las no Apêndice B desta dissertação.

3.3 Integração do JFLAP no Moodle

Uma das primeiras etapas do processo de integração do JFLAP no ambiente Moodle foi o estudo de suas arquiteturas. O Moodle, desenvolvido sob a linguagem de programação PHP, tem uma arquitetura do tipo cliente/servidor e o JFLAP, codificado em Java, em sua versão 4.0 é composto de um cliente *desktop* e um cliente *applet*.

Grande parte do estudo da arquitetura do Moodle foi possível a partir de suas documentações⁶ disponíveis na *Internet* e o mesmo ocorreu no estudo da arquitetura do JFLAP⁷. Passado este período de estudos de arquitetura, durante a implementação das facilidades para o professor e para o aluno (Seções 3.1 e 3.2) foi necessário desenvolver um mecanismo para salvar as soluções dadas pelo professor e pelos alunos a uma questão específica de TC. Essas soluções são diagramas de modelos computacionais desenhados no editor da ferramenta JFLAP, embutido em uma página PHP do Moodle (região 2 da Figura 3.2 e Figura 3.7).

Para salvar estas soluções foi necessário:

- 1°. utilizar as tags `<applet></applet>` da linguagem PHP para exibir os editores do JFLAP;
- 2°. acrescentar métodos na ferramenta JFLAP para converter o diagrama do modelo computacional em *String* e de *String* em diagrama;

⁶ Disponível em <http://xref.moodle.org/nav.html?lib/formslib.php.html>.

⁷ Disponível em <http://www.cs.duke.edu/csed/jflap/log/form.php>.

3º. usar a passagem de parâmetros, disponível nas tags `<applet></applet>`, para:

- armazenar em BD a *String* correspondente ao diagrama desenhado no editor do JFLAP;
- recuperar do BD a *String* e exibir o diagrama do modelo.

Realizada essa etapa buscou-se utilizar outros recursos do JFLAP, como a conversão de modelos computacionais em seus equivalentes e de testes de palavras, para implementar a correção automática de alguns tipos de exercícios e a semi-automática para outros.

Assim, as questões específicas de TC relacionadas à criação de modelos computacionais que possuem equivalência entre máquinas (AFND, AFD, GR e ER), a solução dada pelo usuário passa internamente pelos processos de conversão em AFD e minimização deste, chegando a um AFD que é único por definição (MENEZES, 2000, p. 66). Após essas conversões, a correção poderá ser automática, pois basta comparar os autômatos mínimos da solução do professor e do aluno, bem como a comparação entre as palavras de testes aceitas pela solução do professor com as palavras de testes aceitas pela solução do aluno.

Já para questões específicas, as quais suas soluções são os modelos AFP ou MT, que não possuem equivalência entre máquinas, a correção é semi-automática, isto é, não são realizadas conversões internas, apenas verificações com palavras de testes, necessitando posteriormente revisão pelo professor desta correção. Detalhes sobre a implementação desta integração em uma nova versão do Moodle podem ser consultados no Apêndice C.

3.4 Materiais didáticos para o ensino e a aprendizagem em TC

Como relatado na Introdução desta dissertação, o professor principiante pode encontrar dificuldades adicionais em relação ao professor experiente de disciplinas de TC devido sua pouca experiência em ministrar aulas, elaborar material didático a ser apresentado em sala (e disponibilizado aos alunos) e encontrar rapidamente estratégias (dispersas na literatura) de ensino e aprendizagem de TC que possa aplicar.

Devido a esses problemas, um curso foi construído no ambiente computacional Moodle com vários materiais utilizados por professores de disciplinas de TC da UFPA como: conjuntos de slides, artigos que sugerem estratégias para o ensino e a aprendizagem de TC, listas de exercícios, ferramentas computacionais, apostilas etc. Portanto, esses recursos podem auxiliar tanto a tarefa de ensino realizada pelo professor (principiante ou não) como a aprendizagem dos alunos. Esse curso criado no Moodle é denominado neste trabalho de “curso básico” e é apresentado a partir de suas partes constituintes nas Subseções de 3.4.1 a 3.4.7.

3.4.1 Sistema de arquivos

No curso básico deste trabalho, o professor tem disponível vários materiais no sistema de arquivos do ambiente Moodle, que pode ser acessado clicando-se em “Arquivos” (envolto pelo retângulo com bordas em vermelho) na seção Administração à esquerda na tela principal do ambiente (ver Figura 3.12).

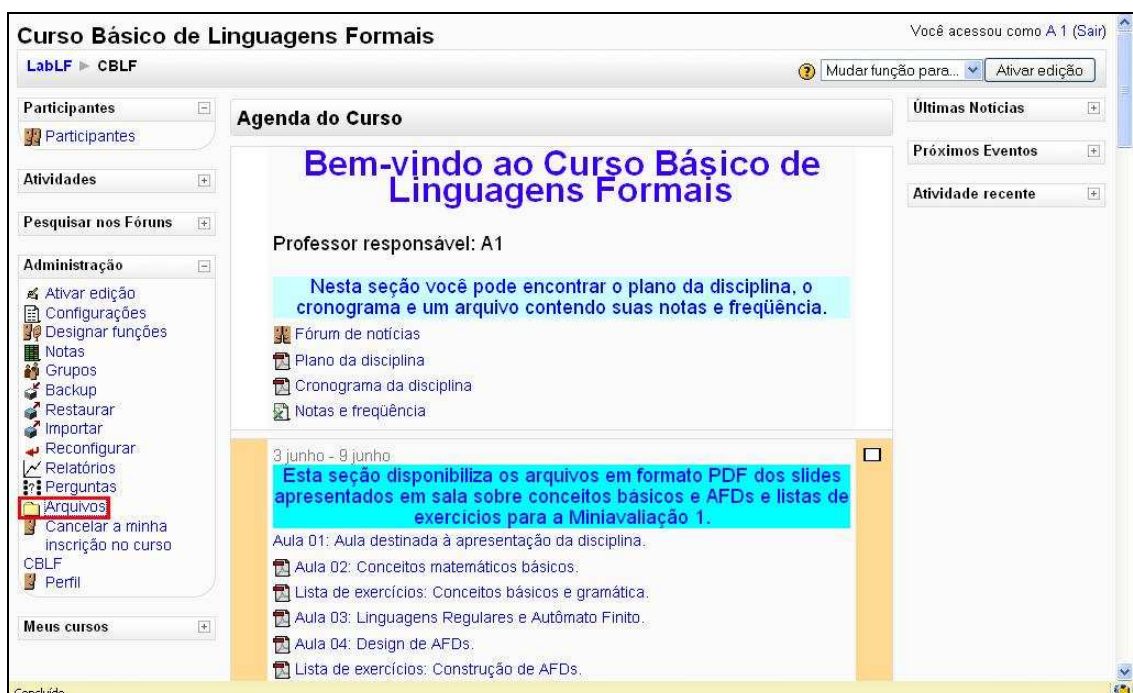


Figura 3.12 – Tela principal do Moodle.

A tela que se segue ao clicar em “Arquivos” é vista na Figura 3.13 em que se pode verificar que existem oito pastas criadas que serão nas próximas Subseções analisadas.

Curso Básico de Linguagens Formais Você acessou como A 1 (Sair)

LabLF > CBLF > Arquivos

Nome	Tamanho	Modificado	Ação
<input type="checkbox"/> Apostilas	4.1Mb	3 junho 2010, 11:17	Renomear
<input type="checkbox"/> Artigos_do_aluno	1.2Mb	2 junho 2010, 22:29	Renomear
<input type="checkbox"/> Artigos_do_professor	2.3Mb	2 junho 2010, 22:25	Renomear
<input type="checkbox"/> Documentos_auxiliares	284.8Kb	3 junho 2010, 10:44	Renomear
<input type="checkbox"/> Ferramentas_computacionais	543.7Kb	2 junho 2010, 22:31	Renomear
<input type="checkbox"/> Imagens	18.8Kb	2 junho 2010, 19:36	Renomear
<input type="checkbox"/> Listas_de_exercicios	440.6Kb	2 junho 2010, 22:16	Renomear
<input type="checkbox"/> Material_de_sala	6.6Mb	2 junho 2010, 22:08	Renomear

Com arquivos escolhidos... ▾

[Documentação de Moodle relativa a esta página](#)

Você acessou como A 1 (Sair)

Figura 3.13 – Sistema de arquivos do Moodle.

3.4.2 Documentos auxiliares

É esperado do professor que alguns documentos, como o cronograma da disciplina e a forma de avaliação, sejam disponibilizados aos alunos. O curso básico em sua pasta “Documentos_auxiliares” do sistema de arquivos disponibiliza quatro documentos básicos para que o professor organize sua disciplina a Figura 3.14 apresenta o conteúdo desta pasta.

Curso Básico de Linguagens Formais Você acessou como A 1 (Sair)

LabLF > CBLF > Arquivos > Documentos_auxiliares

Nome	Tamanho	Modificado	Ação
Diretório pai			
<input type="checkbox"/> Cronograma.doc	80Kb	2 junho 2010, 21:22	Renomear
<input type="checkbox"/> Cronograma.pdf	26.2Kb	2 junho 2010, 21:23	Renomear
<input type="checkbox"/> Notas_e_frequencia.xls	45Kb	2 junho 2010, 21:38	Renomear
<input type="checkbox"/> Plano_da_disciplina.doc	54.5Kb	2 junho 2010, 20:33	Renomear
<input type="checkbox"/> Plano_da_disciplina.pdf	24.5Kb	2 junho 2010, 20:34	Renomear
<input type="checkbox"/> seminario.doc	37.5Kb	3 junho 2010, 10:43	Renomear
<input type="checkbox"/> seminario.pdf	17.2Kb	3 junho 2010, 10:44	Renomear

Com arquivos escolhidos... ▾

[Documentação de Moodle relativa a esta página](#)

Você acessou como A 1 (Sair)

Figura 3.14 – Conteúdo de “Documentos_auxiliares”.

Nesta pasta o professor conta o arquivo “Cronograma.doc”, que permite ao professor elaborar o cronograma da disciplina e disponibilizá-lo no ambiente aos alunos por meio de um *link* (ver Figura 3.15). O Apêndice D apresenta o cronograma utilizado em uma turma real de Linguagens Formais na UFPA.

Curso Básico de Linguagens Formais Você acessou como A 1 (Sair)

LabLF > CBLF > Recursos > Modificando um Recurso

Acrescentando um(a) novo(a) Recurso

Geral

Nome*

Sumário

Trebuchet 1 (8 pt) Língua **B I U S**

Aqui deve ser inserido um *link* para o Cronograma da disciplina em formato PDF do documento "Cronograma.doc" disponível na pasta "Documentos_auxiliares".

Caminho: body

Link a um arquivo ou site

Localização

Janela

Forçar download

Janela

Parâmetros

Configuração de módulos comuns

Visível

Número ID

Este form contém campos obrigatórios

[Documentação de Moodle relativa a esta página](#)

Você acessou como A 1 (Sair)

Figura 3.15 – *Link* do arquivo contendo o cronograma na disciplina na tela principal.

Além do cronograma há outros três arquivos: “Notas_e_frequencia.xls”, “Plano_da_disciplina.doc” e “seminário.doc”, que também podem ser disponibilizados aos alunos na tela principal do Moodle. Estes arquivos ajudam o professor a organizar as notas das avaliações e a frequência dos alunos (“Notas_e_frequencia.xls”), o plano da disciplina (“Plano_da_disciplina.doc”) e as orientações de seminários (“seminário.doc”).

As atividades administrativas do professor organizadas por meio dos arquivos contidos nesse diretório também podem ser realizadas por meio de funcionalidades do próprio ambiente Moodle (por exemplo, o lançamento de notas), ficando a cargo do professor escolher entre o uso destes arquivos ou usar as ferramentas do Moodle.

3.4.3 Material a ser usado em sala

A depender de recursos computacionais (computador e projetor) disponíveis ao professor em sala de aula, os assuntos de suas disciplinas poderão ser apresentados de diferentes maneiras, por exemplo, a partir de vídeos, slides etc. No curso básico o professor conta com 14 conjuntos de *slides* que devem ser utilizados na(s) disciplina(s) de TC relacionada(s) ao estudo da Teoria dos Autômatos. Na maior parte das vezes o nome dado a tal disciplina é “Linguagens Formais”.

A Figura 3.16 apresenta o conteúdo da pasta “Material_de_sala” disponível ao professor no sistema de arquivos Moodle. Nela estão contidos 14 conjuntos de *slides* (extensão “.ppt”) que contêm assuntos relativos ao estudo de modelos computacionais desde os conceitos matemáticos básicos necessários para a disciplina, passando por Autômatos Finitos Determinísticos, Gramáticas, Expressões Regulares, Autômatos Finitos com Pilha e terminando com os estudos de Máquinas de Turing.

Curso Básico de Linguagens Formais Você acessou como A 1 (Sair)

LabLF > CBLF > Arquivos > Material_de_sala

Nome	Tamanho	Modificado	Ação
Diretório pai			
<input type="checkbox"/> LF_aula01_conceitos_basicos.pdf	111Kb	2 junho 2010, 22:05	Renomear
<input type="checkbox"/> LF_aula01_conceitos_basicos.ppt	229.5Kb	2 junho 2010, 21:58	Renomear
<input type="checkbox"/> LF_aula02_AF.pdf	129.4Kb	2 junho 2010, 22:06	Renomear
<input type="checkbox"/> LF_aula02_AF.ppt	343.5Kb	2 junho 2010, 21:58	Renomear
<input type="checkbox"/> LF_aula03_design_AFD_AFND.pdf	160.3Kb	2 junho 2010, 22:06	Renomear
<input type="checkbox"/> LF_aula03_design_AFD_AFND.ppt	351.5Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula04_equiv_AFD_AFND.pdf	120Kb	2 junho 2010, 22:06	Renomear
<input type="checkbox"/> LF_aula04_equiv_AFD_AFND.ppt	284Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula05_equiv_AFD_eAFND.pdf	137.5Kb	2 junho 2010, 22:06	Renomear
<input type="checkbox"/> LF_aula05_equiv_AFD_eAFND.ppt	347Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula06_ER.pdf	53.3Kb	2 junho 2010, 22:06	Renomear
<input type="checkbox"/> LF_aula06_ER.ppt	102.5Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula07_equiv_ER_AF.pdf	131.4Kb	2 junho 2010, 22:06	Renomear
<input type="checkbox"/> LF_aula07_equiv_ER_AF.ppt	384Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula08_minimizacao.pdf	183.8Kb	2 junho 2010, 22:06	Renomear
<input type="checkbox"/> LF_aula08_minimizacao.ppt	455.5Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula09_bombeamento.pdf	92.6Kb	2 junho 2010, 22:07	Renomear
<input type="checkbox"/> LF_aula09_bombeamento.ppt	264Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula10_GLC.pdf	90.4Kb	2 junho 2010, 22:07	Renomear
<input type="checkbox"/> LF_aula10_GLC.ppt	221Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula11_FNC.pdf	75.9Kb	2 junho 2010, 22:07	Renomear
<input type="checkbox"/> LF_aula11_FNC.ppt	207Kb	2 junho 2010, 21:59	Renomear
<input type="checkbox"/> LF_aula12_PDA.pdf	176.1Kb	2 junho 2010, 22:07	Renomear
<input type="checkbox"/> LF_aula12_PDA.ppt	346.5Kb	2 junho 2010, 22:00	Renomear
<input type="checkbox"/> LF_aula13_Maquinas_Turing_I.pdf	276.2Kb	2 junho 2010, 22:07	Renomear
<input type="checkbox"/> LF_aula13_Maquinas_Turing_I.ppt	912Kb	2 junho 2010, 22:00	Renomear
<input type="checkbox"/> LF_aula14_Maquinas_Turing_II.pdf	142.8Kb	2 junho 2010, 22:08	Renomear
<input type="checkbox"/> LF_aula14_Maquinas_Turing_II.ppt	393.5Kb	2 junho 2010, 22:00	Renomear

Com arquivos escolhidos...

[Documentação de Moodle relativa a esta página](#)

Você acessou como A 1 (Sair)

Figura 3.16 – Conteúdo de “Material_de_sala”.

Para disponibilizar esses materiais apresentados em sala de aula aos alunos, o professor pode criar um *link* entre o arquivo PDF, correspondente aos slides apresentados, na tela principal do Moodle como mostra a Figura 3.16. Nela se pode ver a criação do *link* para o arquivo “LF_aula01_conceitos_basicos.pdf” gerado a partir de “LF_aula01_conceitos_basicos.ppt” para disponibilizar os conceitos matemáticos básicos tratado em sala aos alunos.

Curso Básico de Linguagens Formais Você acessou como A 1 (Sair)

LabLF ► CBLF ► Recursos ► Modificando um Recurso

Acrescentando um(a) novo(a) Recurso em semana 1 ?

Geral

Nome*

Sumário ?

Trebuchet 1 (8 pt) Língua **B** **I** **S** **x** **x** **W** **C**

Esta aula discorre sobre os principais conceitos matemáticos necessários para o início dos assuntos da disciplina de Linguagens Formais como: definição de alfabeto, operações em conjuntos etc.

Caminho: `body`

Link a um arquivo ou site

Localização

Janela

Forçar download ?

Janela

Parâmetros

Configuração de módulos comuns

Visível

Número ID ?

Este form contém campos obrigatórios

? [Documentação de Moodle relativa a esta página](#)
 Você acessou como A 1 (Sair)

Figura 3.17 – Link do arquivo “LF_aula01_conceitos_basicos.pdf” na tela principal.

3.4.4 Listas de exercícios

Embora os questionários online tenham suas vantagens e possam substituir as listas de exercícios, uma indisponibilidade temporária do Moodle (pode prejudicar a aprendizagem dos alunos, pois eles não mais terão acesso aos questionários). Assim, é necessário que sempre existam a versão dessas listas de exercícios, como arquivos em formato PDF, para que os alunos possam obtê-las e respondê-las da maneira tradicional com lápis e papel.

No curso básico as listas de exercícios estão inseridas na pasta “Listas_de_exercicios” no sistema de arquivos. Estas listas contêm exercícios de todos os assuntos contidos nos *slides* disponibilizados no curso (Figura 3.18).

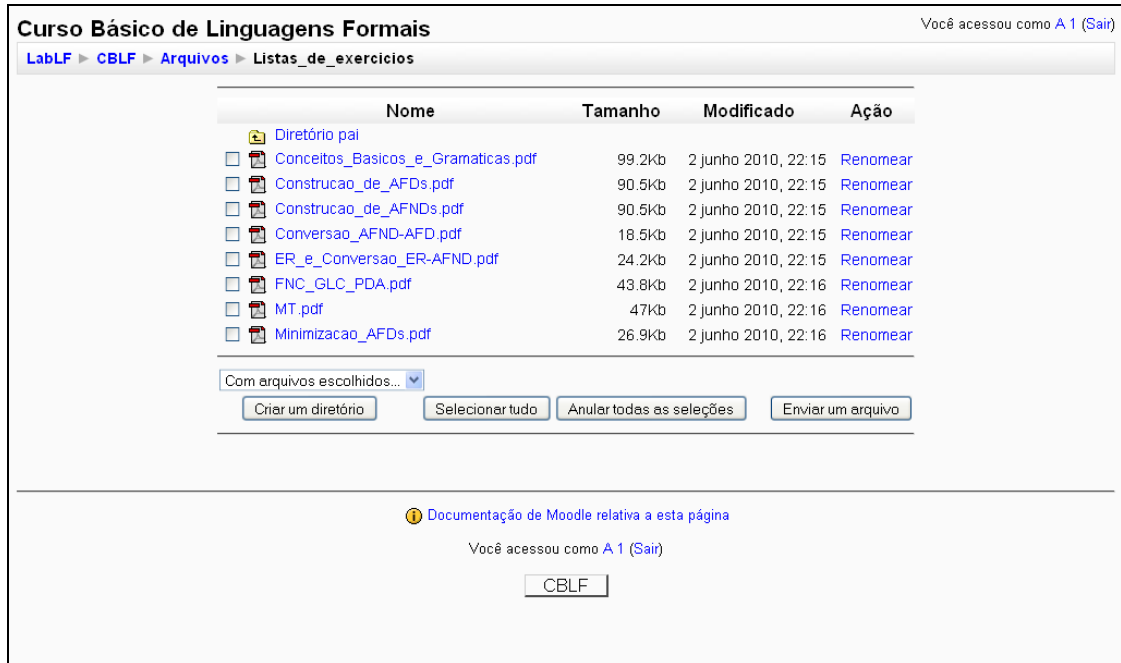


Figura 3.18 – Conteúdo de “Listas_de_exercicios”.

Para disponibilizá-las aos alunos o professor pode utilizar o processo de criação de um *link* desses arquivos na tela principal como visto na Subseção 3.4.3.

3.4.5 Apostilas

As apostilas são um material didático importante para a aprendizagem dos alunos. Elas permitem o estudo dos vários assuntos de TC pela visão de outros autores. A coleta destas apostilas se deu por meio de pesquisas realizadas na *Internet*.

A Figura 3.19 apresenta o conteúdo da pasta de “Apostilas” no curso básico. Estas apostilas contêm assuntos referentes a todos os tópicos estudados na disciplina de Linguagens Formais e, em algumas delas, outros tópicos que servirão para disciplinas de TC posteriores.

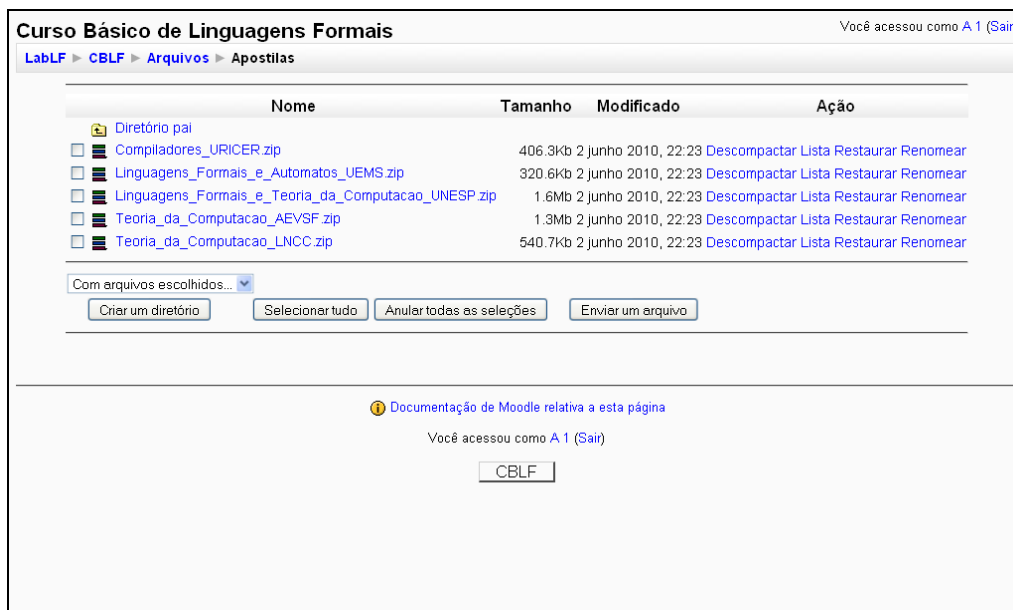


Figura 3.19 – Conteúdo de “Apostilas”.

Ainda na Figura 3.19, o arquivo intitulado:

- “Compiladores_URICER.zip” contém um arquivo com 61 folhas da Universidade Regional Integrada do Alto Uruguai e das Missões que trata de modelos computacionais inicialmente e, parte para o estudo de compiladores;
- “Linguagens_Formais_e_Automatos_UEMS.zip” contém uma apostila com 60 páginas, disponibilizado pela Universidade Estadual do Mato Grosso do Sul, que discorre sobre modelos computacionais, os processos de equivalência entre modelos e trata dos problemas computáveis e não-computáveis;
- “Linguagens_Formais_e_Teoria_da_Computacao_UNESP.zip” contém, em formato PDF, 6 pequenas apostilas utilizadas por professores da Universidade Estadual Paulista, que discorre sobre os modelos computacionais AFD, AFND, AFP, MT , sobre Gramáticas e ERs e ainda inicia os estudos de Computabilidade;
- “Teoria_da_Computacao_AEVSF.zip” contém 86 páginas e apresenta todos os modelos computacionais estudados em Linguagens Formais (AFD, AFND, Gramática, ER, AFP e MT) além de exercícios com gabarito; e
- “Teoria_da_Computacao_LNCC.zip” discorre sobre Computabilidade.

Estas apostilas estão disponíveis aos alunos na seção Apostilas (envolta por um retângulo de bordas em vermelho) do curso básico como pode ser vista na Figura 3.20.



Figura 3.20 – Apostilas disponibilizadas aos alunos.

3.4.6 Artigos

Os artigos coletados estão divididos em dois grupos: a) os artigos destinados a auxiliar a tarefa de ensino do professor (Subseção 3.4.6.1); e b) os artigos que apresentam aplicações dos assuntos estudados nas disciplinas de TC para auxiliar a aprendizagem dos alunos (Subseção 3.4.6.2).

3.4.6.1 Artigos para o professor

A importância destes artigos para a tarefa de ensino realizada pelo professor está em apresentar estratégias (boas práticas) para conduzir uma disciplina de TC. No curso básico estão disponíveis cinco artigos na pasta “Artigos_do_professor” no sistema de arquivos Moodle. A Figura 3.21 apresenta o conteúdo desta pasta.

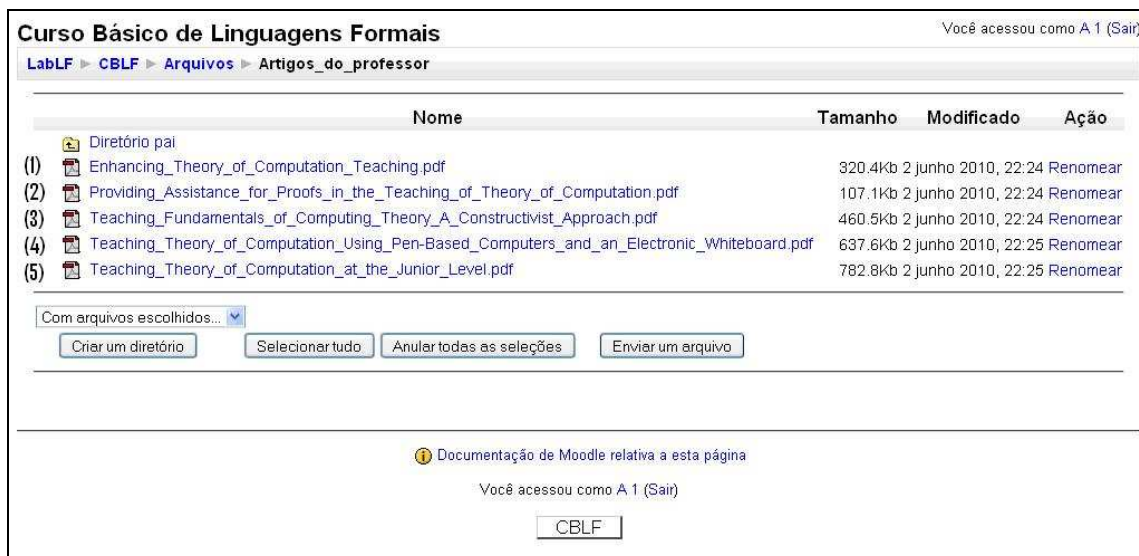


Figura 3.21 – Conteúdo de “Artigos_do_professor”.

O artigo visto em (1) apresenta quatro estratégias para alcançar a meta de ensino dos professores de TC. As estratégias são:

- integrar os fundamentos de TC a assuntos da mesma área (por exemplo Compiladores),
- integrar os fundamentos de TC a assuntos de áreas paralelas (por exemplo Engenharia de Software),
- incentivar a leitura de artigos de pesquisa em TC, e
- utilizar ferramentas computacionais de auxílio à aprendizagem em TC.

O artigo visto em (2) é importante porque ele aborda os benefícios em se utilizar um provador de teoremas e tenta mostrar ao aluno que não existe uma maneira de automatizar o processo de provas.

O artigo em (3) discorre sobre as estratégias de:

- ensinar a partir de contextualização histórica,
- utilizar de ferramentas computacionais para auxiliar professores e alunos,
- estabelecer um *link* entre os fundamentos de TC com linguagens de programação, e
- propor leitura de artigos de pesquisa de TC.

O artigo em (4) se resume a apresentar os resultados obtidos com o uso de computadores *pen-based* e um quadro branco eletrônico em disciplinas de TC. Por fim, o artigo visto em (5) apresenta fundamentalmente o uso de ferramentas para auxiliar o ensino e a aprendizagem de TC, bem como enuncia a necessidade de oferecer (o quanto antes) na grade curricular dos alunos uma disciplina preparatória para as disciplinas de TC.

3.4.6.2 Artigos para o aluno

No curso básico a pasta “Artigos_do_aluno” contém artigos relevantes a aprendizagem dos alunos em disciplinas de TC. Em geral, estes artigos apresentam aplicações dos conceitos estudados em TC à problemas reais. A Figura 3.22 exhibe o conteúdo desta pasta.

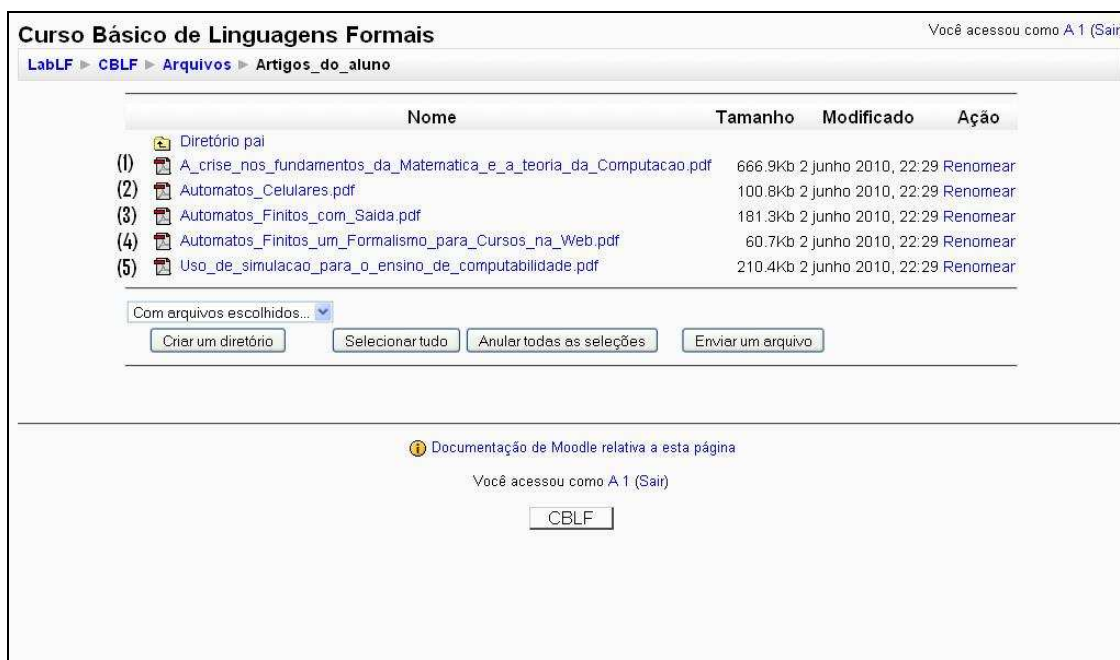


Figura 3.22 – Conteúdo de “Artigos_do_aluno”.

O artigo indicado por:

- (1) apresenta o contexto de nascimento de Teoria da Computação como uma nova linha em Computação;
- (2) aborda uma implementação em Prolog de Autômatos Celulares.
- (3) discorre sobre desenvolvimento de animações computacionais por meio de Autômatos Finitos com Saída;
- (4) trata da aplicação do modelo computacional Autômato Finito como um formalismo para cursos na Web; e

(5) aborda a importância da simulação para o ensino de Computabilidade.

Para disponibilizar aos alunos estes artigos, o curso básico possui uma seção chamada “Artigos” (envolta por um retângulo com bordas em vermelho) em que há um *link* direto aos arquivos da pasta “Artigos_do_aluno” (ver Figura 3.23).



Figura 3.23 – Artigos disponibilizados aos alunos.

3.4.7 Ferramentas computacionais

As ferramentas computacionais estão disponíveis na pasta “Ferramentas_computacionais” e foram coletadas da *Internet*. A Figura 3.23 apresenta o conteúdo desta pasta e pode-se observar que há apenas duas ferramentas: Language Emulator (VIEIRA, J. F. M.; VIEIRA, M. A. M. e VIEIRA, N. J., 2002) e VAS (BOVET, 2006), embora nesta dissertação tenha-se relatado a existência de pelo menos mais três como: JFLAP (RODGER, 2010), GAM (JUKEMURA, NASCIMENTO e UCHÔA, 2005) e SCTMF (COSTA, MENESES e UBER, 2008) as quais estão disponíveis para *download*.

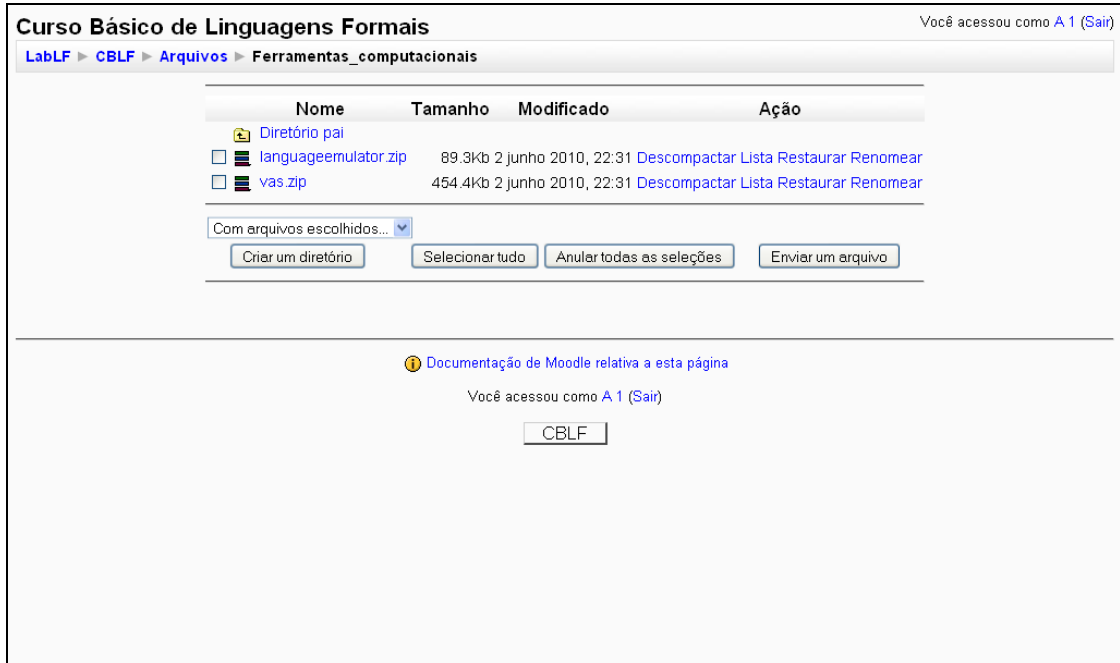


Figura 3.23 – Conteúdo de “Ferramentas_computacionais”.

A inserção no ambiente destas três ferramentas não foi possível pelo fato de JFLAP, mesmo compactada, exceder o tamanho máximo de arquivo permitido para *upload* no Moodle e as ferramentas GAM e SCTMF por não possuírem um arquivo executável disponível para *download*.

Deste modo, cabe ao professor direcionar seus alunos a realizarem o *download* do JFLAP de sua página oficial⁸ (*link* disponível em (1) na Figura 3.24) e, para as ferramentas GAM e SCTMF, entrarem em contato com os autores a partir de seus *e-mails* disponibilizados em seus respectivos artigos (os nomes das ferramentas disponíveis em (2) na Figura 3.24), que podem ser encontrados na *Internet*.

⁸ www.jflap.org

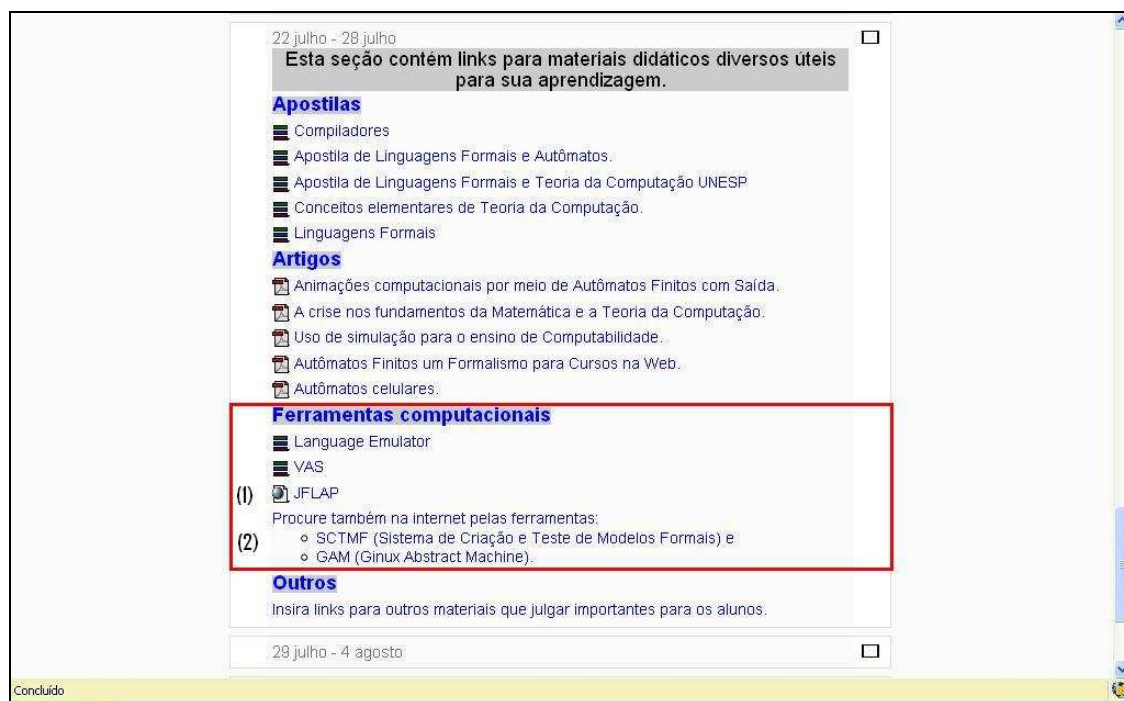


Figura 3.24 – Ferramentas computacionais disponibilizadas aos alunos.

3.5 Considerações do capítulo

O Moodle com adaptações que foi apresentado neste capítulo é sugerido como alternativa a auxiliar, principalmente, o professor principiante e o aluno que deseja saber rapidamente se sua solução a um problema proposto pelo professor é a solução correta ou não. Além disso, com a reunião de **artigos para o professor** sobre estratégias condução de disciplinas de TC, **artigos para os alunos** sobre aplicações dos conceitos estudados em sala à resolução de problemas reais, **apostilas** de outros professores sobre TC, **ferramentas computacionais** para auxiliar a aprendizagem de modelos computacionais, bem como de **templates de documentos** necessários à organização da disciplina, em um curso básico de TC facilita a tarefa do professor de ministrar uma disciplina e, ao mesmo tempo a aprendizagem dos alunos.

CAPÍTULO 4

A UTILIZAÇÃO DO AMBIENTE MOODLE EM UMA DISCIPLINA DE LINGUAGENS FORMAIS

Neste Capítulo 4 será descrita a experiência de uso do ambiente computacional Moodle com adaptações em uma turma da disciplina de Linguagens Formais do curso de Bacharelado em Ciência da Computação da Universidade Federal do Pará no 2º Período Letivo⁹ de 2010 com 18 alunos matriculados. As Seções 4.1 e 4.2 apresentam cada um dos testes feitos.

4.10 uso das funcionalidades de correção

Os testes realizados para observar a adequação das funcionalidades de correção automática e semi-automática de questões específicas de TC em uma disciplina real foram feitos sob dois aspectos: o ensino realizado pelo professor (Subseção 4.1.1) e a aprendizagem do aluno (Subseção 4.1.2).

⁹ O calendário acadêmico da UFPA é constituído de quatro períodos letivos anualmente. O primeiro e o terceiro períodos letivos são mais curtos e ocorrem nos meses de recesso para atender cursos de graduação do interior do estado. O segundo e o quarto períodos letivos têm duração de cerca de quatro meses, entre março e junho e agosto e dezembro, respectivamente.

4.1.1 No ensino

A possibilidade de criar exercícios de questões específicas de TC com correção automática e semi-automática no Moodle é um dos pontos fortes deste trabalho. O professor assim pode acompanhar a aprendizagem de seus alunos, principalmente quando o número de alunos de cada uma de suas turmas é grande. Outro benefício do uso destas correções é simular para os alunos a ajuda do professor a todo o momento, mesmo quando sua carga de trabalho o impossibilita de assistir pessoalmente todos eles.

É importante salientar que os recursos de correção automática e semi-automática não têm por objetivo substituir o professor e sim, auxiliá-lo em sua tarefa de ensino, permitindo a ele acompanhar a aprendizagem de seus alunos e, se for o caso, incentivá-los a “tirarem dúvidas” sobre suas soluções quando o *feedback* não os ajudar.

Ao longo de uma disciplina de Linguagens Formais que ocorreu no 2º Semestre de 2009 na UFPA, os alunos (no total de 15) foram submetidos às funcionalidades de correção automática e semi-automática de questões por meio de trabalhos extraclases. Ao fim de tal disciplina, observou-se que a percentagem de reprovação permaneceu ainda em torno de 30% (como nos semestres anteriores). Este fato mostra que a simples inserção de uma ferramenta na prática não tem por si o poder de mudança. Isto levou à reflexão de que era necessária também uma reorganização da disciplina em termos de avaliações. Basicamente, os alunos eram submetidos a:

- duas avaliações individuais, sendo que na:
 - 1ª Avaliação os alunos respondiam questões relativas aos assuntos:
 - os modelos computacionais AFD e AFND,
 - os tipos de Gramáticas,
 - as Expressão Regulares e
 - o processo de minimização de AFDs.
 - 2ª Avaliação os alunos necessitavam estudar:
 - Gramáticas Livre de Contexto,
 - Forma Normal de Chomsky,
 - Autômatos Finitos com Pilha e

4.1 O uso das funcionalidades de correção

- Máquinas de Turing.

- um seminário em grupo; e
- aulas de resolução de exercícios e “tirar dúvidas” em sala de aula.

Para repensar a organização da disciplina foi realizada uma pesquisa sobre como os outros professores trabalham, um questionário foi aplicado a professores de dez universidades brasileiras e foram coletados dados das páginas das suas disciplinas de TC.

4.1.1.1 Os dados

Para coletar as informações sobre como os professores de TC ministram suas aulas e como avaliam seus alunos, preferiu-se realizar a pesquisa somente com os professores de disciplinas de TC das dez melhores universidades do Brasil em Computação, segundo a revista Info Exame (Brasil). A Tabela 4.1 apresenta as dez universidades melhores classificadas.

Tabela 4.1 – As dez melhores universidades do Brasil segundo a revista Info Exame.

Colocação	Universidade
1	Universidade Estadual de Campinas
2	Universidade Federal do Rio Grande do Sul
3	Universidade de São Paulo / Instituto de Ciências Matemáticas e de Computação
4	Instituto Tecnológico de Aeronáutica
5	Universidade Federal de Minas Gerais
6	Universidade Federal do Rio de Janeiro
7	Universidade Federal de Pernambuco
8	Instituto Militar de Engenharia
9	Universidade Federal de São Carlos
10	Pontifícia Universidade Católica do Rio de Janeiro

A partir dos dados da Tabela 4.1, um questionário (ver Apêndice F) foi enviado por *e-mail* a cada um destes professores para coletar informações acerca da maneira como eles ministram suas disciplinas de TC e como avaliam seus alunos.

As principais respostas deste questionário foram:

- todos os professores disponibilizam listas de exercícios a seus alunos,
- 70% (7/10) dos professores realizam três avaliações individuais,
- 100% (10/10) dos professores propõem um trabalho e/ou seminário como outra maneira de avaliar seus alunos,
- as turmas tem por volta de 50 alunos na graduação e

4.1 O uso das funcionalidades de correção

- o índice de reprovação mais alto informado foi de 20%.

Desse modo, pôde-se verificar que a disciplina ofertada na UFPA tinha sua organização semelhante à das universidades entrevistadas. Ainda assim, a disciplina, a ser reofertada no 2º período letivo de 2010, foi reorganizada e foram incluídas as seguintes avaliações:

- seis miniavaliações individuais em que as três primeiras correspondem à antiga 1ª Avaliação e as três últimas à 2ª Avaliação.
- um seminário e
- aulas especiais para “tirar dúvidas” e resolver exercícios de listas propostas pelo professor em laboratório.

O Apêndice D apresenta uma sugestão de cronograma a ser seguido pelo professor nesta organização proposta.

4.1.1.2 Avaliação da reorganização da disciplina

As respostas dadas pelos vários professores de disciplinas de TC apontam para a aplicação de provas individuais e listas de exercícios, bem como a realização de trabalhos e seminários como estratégias para o ensino de TC. Mesmo sabendo que estas estratégias são realizadas na UFPA, ainda se têm percentuais de reprovação nestas disciplinas em torno de 30%, superior ao maior percentual de reprovação informado por uma das dez melhores universidades brasileiras que foi de 20%.

Nesta etapa esperava-se, portanto, que a nova organização da disciplina de Linguagens Formais da UFPA em seis miniavaliações, um seminário e aulas especiais para “tirar dúvidas” em laboratório usando o Moodle pudesse trazer resultados positivos para a tarefa de ensinar do professor e, conseqüentemente para a aprendizagem dos alunos.

Assim, para avaliar os efeitos da organização da disciplina, o grupo de alunos (seis) que não alcançou média suficiente na disciplina de Linguagens Formais no 4º período letivo de 2009 (em que havia duas avaliações individuais) e que estão matriculados na disciplina de Linguagens Formais do 2º Período Letivo de 2010 (em que há seis miniavaliações) foi entrevistado (veja perguntas no Apêndice G).

A estes alunos foi solicitado que apontassem os pontos fortes e fracos da divisão das duas avaliações individuais em seis miniavaliações. Dos seis alunos entrevistados apenas um apontou o tempo para realização de algumas miniavaliações ser reduzido (em torno de 50 minutos em relação a outras em que foram dados 110 minutos). Em contrapartida:

- dois alunos relataram como ponto forte o menor número de assuntos a serem estudados para cada uma das miniavaliações e
- outros dois alunos disseram que o ponto forte da divisão das duas avaliações em seis miniavaliações é a assimilação maior de informações mais facilmente.

Outros pontos fortes relatados foram: a partição dos assuntos considerados difíceis, a sensação de mais tempo para estudar e “tirar dúvidas” e o estudo contínuo dos alunos, devido ao intervalo entre uma miniavaliação e outra ser pequeno.

4.1.2 Uso da correção automática em aulas em laboratório

Para se observar o uso das funcionalidades de correção automática e semi-automática na aprendizagem dos fundamentos de TC, no cronograma da disciplina foram inseridas, durante o período de março a junho de 2010, aulas especiais de “tirar dúvidas” e de resolver problemas específicos de TC no Moodle, permitindo aos alunos utilizarem as funcionalidades de correção. Nessas aulas especiais que ocorrem em laboratório, os alunos deveriam acessar o Moodle para resolver problemas específicos de TC propostos pelo professor.

Assim, inicialmente, se observou a surpresa dos alunos quando eles descobriram que poderiam saber rapidamente se suas soluções aos problemas específicos de TC estavam corretas ou não. Outro aspecto observado foi que, num primeiro momento, muitas “dúvidas” surgiram a respeito da solução submetida e do *feedback* informado em seguida. O professor e o monitor da disciplina, que sempre estavam presentes a estas aulas especiais, foram chamados pelos alunos várias vezes, pois havia descrédito em relação ao sistema. A principal dúvida era saber, realmente, se a solução dada por eles estavam corretas, mesmo que os *feedbacks* informassem que sim. Considerou-se então que este primeiro contato com as funcionalidades de correção serviu para fazer os alunos acreditarem que o *feedback* gerado informa se a solução construída por eles é a solução esperada pelo professor ou, se não alcançada esta, o que precisa ser reconhecido ou não em sua resposta para que alcance aquela resposta esperada pelo professor.

Portanto, a adequação das funcionalidades de correção automática e semi-automática de questões específicas de TC aliada à mudança no processo de avaliação pode ter influenciado no desempenho melhor destes alunos nestas questões em específico na avaliação individual.

Os dados para se observar a adequação das funcionalidades de correção automática e semi-automática na aprendizagem foram obtidos a partir do desempenho na 1ª avaliação (dividida em 3 miniavaliações) de cada aluno. Em cada uma das provas, os valores obtidos pelo aluno na resolução de cada questão foram coletados, bem como informações a respeito da tentativa em resolver uma questão ou não.

A partir desta coleta, os dados foram organizados sobre cinco categorias:

- “100%”: o aluno acertou a questão na íntegra;
- “50% a 99,9%”: o aluno não acertou completamente a questão;
- “0,1% a 49,9%”: o aluno não errou totalmente a questão;
- “0%”: o aluno errou completamente a questão; e
- “Não fez”: o aluno não tentou responder a questão.

Assim, a Tabela 4.2 relaciona os assuntos das questões das três miniavaliações, os pontos que poderiam ser obtidos em cada questão e as frequências das categorias conforme os valores obtidos por 15 alunos em cada pergunta. Veja também os mesmo dados da Tabela 4.2 no gráfico da Figura 4.1 em seguida.

Tabela 4.2 – Assuntos das questões e frequência das categorias.

1ª Avaliação	Nº da questão	Assunto	Pontos	Acerto				Não Fez
				100%	50% a 99,9%	0,1% a 49,9%	0%	
Miniavaliação 1	1	Classificação H. Chomsky	1,0	2	3	5	3	2
	2	Derivação	0,5	14	0	0	0	1
	3	Modelo formal de AFDs	0,75	5	1	1	3	5
	4	Diagrama de AFDs	0,75	11	1	3	0	0
Miniavaliação 2	5	Diagrama de AFNDs	1,5	5	7	1	0	2
	6	Conversão AFND - AFD	1,5	1	6	6	0	2
Miniavaliação 3	7	Expressão Regular	1,0	4	9	0	0	2
	8	Conversão ER - AFND	1,0	10	3	0	0	2
	9	Minimização de AFDs	2,0	9	3	0	0	3

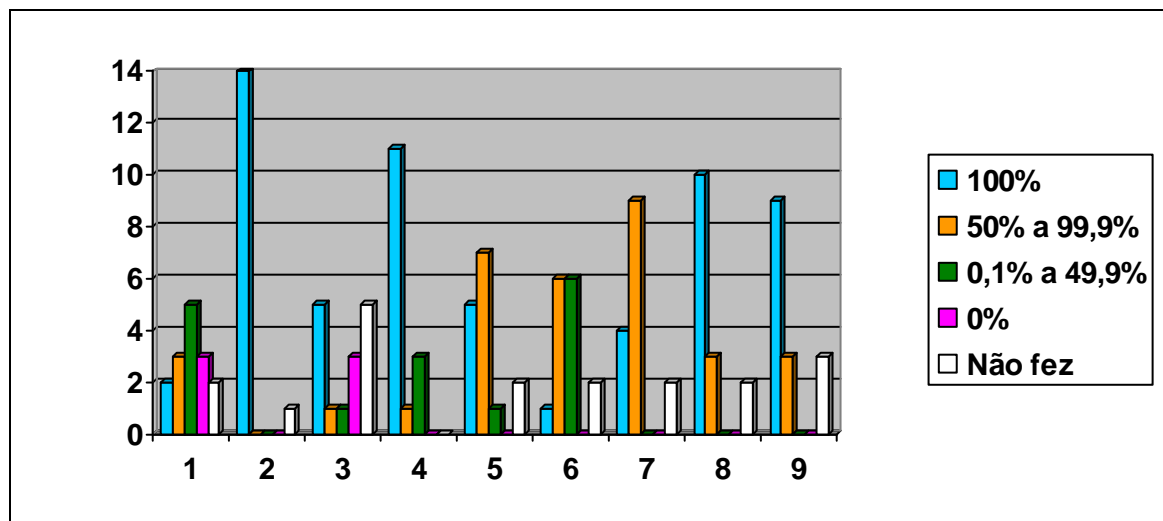


Figura 4.1 – Questões da avaliação e frequência.

Pela tabela 4.2 se observa que a 1ª Avaliação abordou assuntos como: classificação de linguagens segundo a hierarquia de Chomsky, o processo de derivação em gramáticas, a definição formal de AFDs, o design de AFDs e AFNDs, os processos de conversão de AFNDs em AFDs e de ERs em AFNDs, Expressão Regular e a minimização de AFDs.

Nos assuntos sobre design de AFDs (“Diagrama de AFDs”) e de AFNDs (“Diagrama de AFNDs”) e Expressão Regular (“Expressão Regular”), que foram conteúdos abordados nas aulas de laboratório com o Moodle, se observou que:

- 80% (12 alunos) conseguiram acertar pelo menos metade da questão sobre design de AFDs (“Diagrama de AFDs”) (11 alunos acertaram 100% da questão e 1 aluno acertou entre 50% e 99,9% dela);
- 80% (12 alunos) conseguiram acertar pelo menos metade da questão sobre design de AFNDs (“Diagrama de AFNDs”) (5 alunos acertaram 100% da questão e 7 alunos acertaram entre 50% e 99,9% dela); e
- 86,66% (13 alunos) conseguiram acertar pelo menos metade da questão sobre Expressão Regular (“Expressão Regular”) (4 alunos acertaram 100% da questão e 9 alunos acertaram entre 50% e 99,9% dela).

Ao mesmo tempo, em questões sobre a classificação de linguagens segundo a hierarquia de Chomsky (“Classificação H. Chomsky”), a definição formal de AFDs (“Modelo formal de AFDs”) e o processo de conversão de AFNDs em AFDs (“Conversão AFND - AFD”), se observam que:

- 66,66% (10 alunos) não conseguiram acertar pelo menos metade da questão sobre classificação de linguagens segundo a hierarquia de Chomsky (“Classificação H. Chomsky”) (5 alunos acertaram entre 0,1% e 49,9%, 3 alunos erraram completamente a questão e 2 alunos não a fizeram);
- 60% (9 alunos) não conseguiram acertar pelo menos metade da questão sobre definição formal de AFDs (“Modelo formal de AFDs”) (1 aluno acertou entre 0,1% e 49,9%, 3 alunos erraram completamente a questão e 5 alunos não a fizeram).
- 53,33% (8 alunos) não conseguiram acertar pelo menos metade da questão sobre o processo de conversão de AFNDs em AFDs (“Conversão AFND - AFD”) (6 alunos acertaram entre 0,1% e 49,9%, nenhum aluno errou completamente a questão e 2 alunos não a fizeram).

Neste contexto, pode-se observar que há mais alunos acertando pelo menos metade das questões da prova, as quais abordavam assuntos que foram alvo das aulas de “tirar dúvidas” em laboratório. Veja no gráfico da Figura 4.1 que as barras em azul e as em laranja detêm juntas maior número de alunos nas questões sobre design de AFDs (representada por 4), de AFNDs (representada por 5) e Expressão Regular (representada por 7).

Ainda na Figura 4.1, as questões sobre hierarquia de Chomsky (representada por 1), definição formal de AFDs (representada por 3) e conversão de AFNDs em AFDs (representada por 6), que não foram tão praticadas em aulas de “tirar dúvidas” apresentam menores quantidades de alunos que acertaram pelo menos metade da questão.

Comparativamente à disciplina de Linguagens Formais do 4º período letivo de 2009 (com aulas de “tirar dúvidas” em sala) em que a média das notas da 1ª Avaliação foi **5,958**, a disciplina de Linguagens Formais do 2º período letivo de 2010 (com aulas de “tirar dúvidas” em laboratório e nova organização das avaliações) a média das notas da 1ª Avaliação sobre os mesmos assuntos foi **6,83**. Na 2ª Avaliação as médias foram de **4,563** para a disciplina em 2009 e **6,539** na disciplina em 2010.

É cedo ainda para fazer afirmações a respeito da influência dessa nova organização e das aulas em laboratório na aprendizagem dos alunos (o semestre ainda está em andamento), mas o aumento da média das notas de **5,958** para **6,83** na 1ª Avaliação e de **4,563** para **6,539** na 2ª Avaliação de Linguagens Formais pode se refletir em um índice de reprovação abaixo de 30%.

4.2 As entrevistas com os alunos

Os alunos que finalizaram a disciplina de Linguagens Formais do 2º Período Letivo de 2010 da UFPA foram submetidos a uma entrevista semi-estruturada (Apêndice G) para levantar suas opiniões, idéias, atitudes, juízos (VIEIRA, 2009) acerca da adequação do curso básico à aprendizagem de TC. E para uma melhor análise resolveu-se dividir os alunos em dois grupos: os alunos que estão estudando Linguagens Formais pela primeira vez (“1ª vez”) e os alunos que estão estudando por pelo menos mais de uma vez (“pelo menos mais de uma vez”).

Assim, após a elaboração das perguntas a serem feitas pelo entrevistador a cada um dos alunos, a escolha do melhor dia para a entrevista se baseou nos dias em que a disciplina é ministrada. Desta maneira, a coleta de informações de nove destes alunos aconteceu no dia 11 de maio de 2010 e de outros três no dia 13 de maio de 2010.

Analisando as informações sob os dois grupos de alunos da disciplina: “1ª vez” e “pelo menos mais de uma vez”, dos alunos de “1ª vez”:

- 83,33% (5/6) dizem ter suprido suas necessidades na disciplina sobre Linguagens Formais,
- 100% (6/6) afirmam que a verificação pelo ambiente de uma solução a um problema proposto os ajuda muito em sua aprendizagem,
- 33,33% (2/6) apontam a associação que fazem entre teoria e prática como maneira de resolver um problema qualquer utilizando os conceitos de TC,
- todos (6/6) afirmam que o *feedback* gerado na correção de questões específicas de TC os ajudam a saberem se suas soluções estão corretas, e
- 66,66% (4/6) dizem que os exercícios realizados no ambiente os ajudaram em grau 9, numa escala de 0 a 10, a resolver questões de avaliações.

Quanto aos alunos de “pelo menos mais de uma vez”:

- 100% (6/6) dizem agora terem suprido suas necessidades na disciplina sobre Linguagens Formais,

- 83,33% (5/6) afirmam que a verificação pelo ambiente de uma solução a um problema proposto os ajuda muito em sua aprendizagem e 16,67 % (1/6) diz tê-lo ajudado razoavelmente,
- 33,33% (2/6) apontam para o entendimento do assunto como a principal maneira de aplicar os conceitos TC a outros problemas e 33,33% (2/6) afirmam ser o raciocínio lógico,
- 66,67 % (4/6) afirmam que o *feedback* gerado na correção de questões específicas de TC os ajudam em parte a saberem se suas soluções estão corretas, e
- 50% (3/6) dizem que os trabalhos realizados no ambiente os ajudaram em grau 7, numa escala de 0 a 10, a resolver questões de avaliações.

A análise das informações sobre este grupo também revelou que as causas para o insucesso de cada um dos alunos em turmas anteriores foram variadas, por exemplo: problemas pessoais, compreensão dos assuntos e desinteresse na disciplina.

4.3 Considerações do capítulo

Inicialmente, o uso pelos alunos das funcionalidades de correção de questões específicas de TC no ambiente Moodle trouxe dúvidas e desconfiança a eles a respeito do *feedback* gerado pelo ambiente. Além disso, observou-se que o Moodle, ao exibir respostas, sinalizava com a cor vermelha que a solução dos alunos estava errada. Porém, ao olhar a porcentagem de acerto exibida pelo ambiente, os alunos viam que parte da solução estava correta. Isto na verdade não era um erro de correção e sim um erro de sinalização do Moodle, que “assustava” os alunos.

À medida que os usuários utilizam o Moodle, alguns problemas de interface começaram a aparecer, por exemplo, o caso de erro de sinalização e as interfaces de relatórios que não permitem a consulta das tarefas realizadas em um período. Estes problemas abrem margem para o estudo e mudanças no projeto de interfaces do Moodle para atender melhor as atividades do aluno e do professor.

A partir dos dados coletados observou-se que o grupo de alunos, que realizam a disciplina por pelo menos mais de uma vez, ainda têm dificuldades, mas a organização da disciplina e a inserção das funcionalidades de correção de questões específicas no Moodle os têm ajudado.

Também se observou que houve um maior número de alunos com acertos abaixo de 50% em algumas questões que não foram tão exercitadas nas aulas especiais de “tirar dúvidas”, o que sugere uma mudança na quantidade de exercícios sobre os assuntos destas questões em listas e em questionários a serem respondidos pelos alunos nessas aulas especiais. A seguir no Capítulo 5 são apresentadas as conclusões desta dissertação de mestrado.

CAPÍTULO 5

CONCLUSÕES

A Introdução e o Capítulo 2 apresentaram ao leitor dificuldades encontradas por professores e alunos em ensinar e em aprender fundamentos de TC, respectivamente. Algumas destas dificuldades são: a) descritas em trabalhos da literatura e/ou b) obtidas a partir de comentários de professores e alunos de TC da UFPA.

Em meio a isto, uma dificuldade é **o professor acompanhar a aprendizagem dos alunos** quando este tem uma sobrecarga de trabalho e um grande número de alunos por turma. Assim, para auxiliar o professor, optou-se por adaptar no ambiente computacional Moodle novos tipos de questões direcionadas, especificamente, para os estudos de modelos computacionais de TC, como pode ser visto na Seção 3.1, e utilizar suas funcionalidades de relatório para verificar se os alunos realizaram ou não as tarefas (atividades) propostas pelo professor.

Outro problema apresentado nesta dissertação é relativo à dificuldade que o professor principiante tem em elaborar material didático a ser apresentado em sala para as disciplinas de TC, haja vista todas as dificuldades já mencionadas do ensino e aprendizagem de TC. Logo, uma maneira encontrada para auxiliar este professor principiante é disponibilizar no ambiente Moodle um curso básico com um conjunto de materiais didáticos para tal, como pode ser vista por toda a Seção 3.4 e no Apêndice E.

Um segundo problema do professor principiante é tender a conduzir as disciplinas de TC de maneira tradicional. No entanto, isto pode gerar outras dificuldades como a desmotivação do professor em ensinar e dos alunos em aprender. A origem deste problema pode ser o desconhecimento do professor da existência de estratégias para o ensino e aprendizagem de TC, bem como da dificuldade de encontrá-las dado sua dispersão pela literatura. Igualmente ao caso anterior dos materiais didáticos, escolheu-se disponibilizar a este professor, artigos envolvendo estratégias disponíveis na literatura para condução de disciplinas de TC como podem ser vistos na Subseção 3.4.6.1.

Outro objetivo definido neste trabalho era relativo ao processo de aprendizagem dos alunos de TC. Como visto anteriormente, um dos problemas dos alunos de TC na UFPA é saber rapidamente se sua solução a um problema proposto pelo professor é a solução esperada por este ou não. O ambiente Moodle com novos tipos de questões específicas de TC, adicionadas de mecanismos de correção automática e semi-automática buscam auxiliar estes alunos nesse problema. A Seção 3.2 discorre sobre estas novas questões específicas e como os alunos podem reduzir estas dificuldades em TC. E a Subseção 4.2.2 apresenta uma análise do desempenho dos alunos da turma do 2º período letivo de 2010 em Linguagens Formais na 1ª Avaliação.

Outro objetivo desta dissertação foi observar a adequação do curso básico contendo materiais didáticos diversos, funcionalidades para correção automática e semi-automática de questões específicas e possibilidade de acompanhamento da aprendizagem dos alunos. As Seções 4.1 e 4.2 apresentaram dados coletados a partir de entrevistas com os alunos da disciplina de Linguagens Formais do 2º Período de 2010 a respeito de suas opiniões sobre a condução do curso e da ajuda dada pelo mesmo na aprendizagem de fundamentos de TC. O que se obteve são relatos de que este curso básico auxilia os alunos em sua aprendizagem em TC.

Uma lição aprendida durante os últimos dois anos de pesquisa neste trabalho é que apenas sugerir o uso de ferramentas de auxílio à aprendizagem de TC entre os alunos não é o melhor caminho quando se deseja que eles as utilizem. O professor deve criar estratégias para que os alunos façam uso de fato desses recursos. Até então eram propostas listas de exercícios e deixado livre o uso do JFLAP. Desse modo o professor não tinha como acompanhar se os alunos estavam fazendo as listas e se estavam usando a ferramenta como apoio. A introdução das aulas de exercícios com o uso do Moodle em laboratório contribui para que o professor possa acompanhar melhor os alunos. O professor pode observar o raciocínio empregado por eles nesse momento e fazer as intervenções necessárias para ajudá-los. Essas aulas de certo modo asseguram um momento em que os alunos tentarão fazer os exercícios.

5.1 As contribuições deste trabalho

A principal contribuição deste trabalho é a possibilidade de auxiliar a aprendizagem dos alunos com questões específicas de TC em que algumas podem ser corrigidas automaticamente, enquanto as outras semi-automaticamente.

Outra contribuição que decorre destas questões específicas de TC implementadas no ambiente Moodle é proporcionar o acompanhamento da aprendizagem dos alunos em TC pelo professor e, a partir das funcionalidades deste ambiente de verificação de atividades (tarefas) realizadas pelos alunos, permitir ao professor direcionar esforços na tentativa de ajudar os alunos que não realizaram as atividades a superarem suas dificuldades.

Uma terceira contribuição se refere ao ambiente Moodle adaptado que, ao ser usado pelo professor, tem o potencial de ajudar na aprendizagem dos alunos. Como visto no Capítulo 2, ferramentas computacionais são recursos integrantes de várias estratégias que facilitam a aprendizagem em TC. Isto abre margem para que modificações similares as deste trabalho sejam feitas em outros ambientes EAD, como Amadeus, TelEduc, WebCT e outros.

Além dessas, a reunião de várias estratégias, que estavam dispersas na literatura, e a organização de diversos materiais didáticos em um curso básico para ajudar professores e alunos é outra contribuição deste trabalho.

5.2 Os trabalhos futuros

No decorrer desta dissertação o leitor se deparou com a apresentação de várias dificuldades tanto no ensino quanto na aprendizagem de fundamentos de TC, e diversas estratégias disponíveis na literatura para ajudar a reduzir tais dificuldades em ambos os lados.

Assim, sugerem-se como propostas de trabalhos nesta linha:

- Desenvolver estratégias que possam auxiliar a aprendizagem dos alunos em TC quando estes têm dificuldades com o embasamento matemático exigido pelas disciplinas de TC e com a abstração dos fundamentos de TC.
- Fazer um novo projeto da interação com o sistema Moodle para facilitar o uso pelo professor das funcionalidades de “Relatório” e aperfeiçoar os *feedback* informados aos alunos.
- Repetir as experiências deste trabalho em outras turmas, visando aperfeiçoar este curso, propondo novos tipos de questões específicas, inserindo novos materiais para o professor e para o aluno.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGUIAR, Robson da Silva; OEIRAS, Janne Yukiko Yoshikawa. Laboratório de Linguagens Formais - LabLF. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 20., 2009, Florianópolis. *Anais...*. Florianópolis, SC: Sociedade Brasileira de Computação, 2009. p. 1 - 8. Disponível em: <<http://wwwexe.inf.ufsc.br/~sbie2009/anais/conteudo/artigos/ctdietcc/61694.pdf>>. Acesso em: 02 jun. 2010.
- AGUIAR, Robson da Silva; OEIRAS, Janne Yukiko Yoshikawa. Ambiente Moodle de auxílio ao ensino e aprendizagem em Linguagens Formais. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 18., 2010, Belo Horizonte. *Anais...*. Belo Horizonte, MG: Sociedade Brasileira de Computação, 2010. p. 1 - 10.
- AMADEUS. Disponível em: < <http://amadeus.cin.ufpe.br/blog/?p=37>>. Acesso em: 02 mar. 2010.
- AULANET. Disponível em: < <http://www.aulanet.pt/>>. Acesso em: 28 mai. 2010.
- BERQUE, Dave; JOHNSON, David K.; JOVANOVIC, Larry. Teaching Theory of Computation Using Pen-Based Computers and an Electronic Whiteboard. In: ITICSE - INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION, 6., 2001, Canterbury. *Anais...*. Canterbury: ITiCSE, 2001. p. 169 - 172.
- 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: 01 mai. 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. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.1251&rep=rep1&type=pdf>>. Acesso em: 20 mar. 2010.
- CHUA, Y. S.; WINTON, C. N.. Teaching Theory of Computation at the Junior level. In: INTERNATIONAL CONFERENCE ON APL, 87., 1987, Dallas. **Proceedings...** . Dallas: ACM, 1987. p. 69 - 78.
- 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 SBC - WEI - WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 28., 2008, Belém. **Arq0119.pdf**. Belém: SBC, 2008. p. 208 - 217. Disponível em: <www.prodepa.gov.br/sbc2008/anais/pdf/arc0119.pdf>. Acesso em: 26 abr. 2010.
- EPROINFO. Disponível em: <<http://eproinfo.mec.gov.br/>>. Acesso em: 02 mar. 2010.
- ESMORIS, Agustín; CHESÑEVAR, Carlos Iván. Una Herramienta para la Simulación de Autómatas Traductores en la Enseñanza de Teoría de la Computación. In: ARGENTINIAN CONGRESS IN COMPUTER SCIENCE, 9., 2003, La Plata. **2003_cacic_automata.pdf**. La Plata: Redunci, 2003. p. 287 - 295. Disponível em: <http://cs.uns.edu.ar/~cic/2003/2003_cacic_automata.pdf>. Acesso em: 30 abr. 2010.
- ESTREBOU, F.; LANZA, M.; MAUCO, V.; BARBUZZA, R.; FABRE, L.. Minerva: Una Herramienta para um curso de lenguajes formales y autómatas. In: PROC. OF THE LAT. CONF. IN INFORMATICS, 2002.
- GOYAL, Mukta; SACHDEVA, Shelly. 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, Eric; RODGER, Susan H.. Using JFLAP to Interact with Theorems in Automata Theory. In: SIGCSE TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, 30., 1999, New Orleans. **Cse99.pdf**. New Orleans: IEEE, 1999. p. 336 - 340. Disponível em: <<http://www.cs.duke.edu/csed/rodger/papers/cse99.pdf>>. Acesso em: 26 abr. 2010.

- JUKEMURA, Anibal S.; NASCIMENTO, Hugo A. D. do; UCHÔA, Joaquim Q.. GAM - Um Simulador para Auxiliar o Ensino de Linguagens Formais e de Autômatos. In: CONGRESSO DA SBC - WEI - WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 13., 2005, Unisinos. **Arq0043.pdf**. São Leopoldo: Congresso da SBC, 2005. p. 2432 - 2443. Disponível em: <http://www.unisinos.br/_diversos/congresso/sbc2005/_dados/anais/pdf/arq0043.pdf>. Acesso em: 01 mai. 2010.
- KRISHNAN, Padmanabhan. Providing Assistance for Proofs in the Teaching of Theory of Computation. In: INTERNATIONAL CONFERENCE ON COMPUTERS IN EDUCATION, 10., 2002, Auckland. **15091083.pdf**. Auckland: IEEE, 2002. v. 2, p. 1083 - 1084. Disponível em: <<http://www.computer.org/plugins/dl/pdf/proceedings/icce/2002/1509/00/15091083.pdf>>. Acesso em: 26 abr. 2010.
- LEARNINGSPLACE. Disponível em: <<http://openlearn.open.ac.uk/>>. Acesso em: 02 mar. 2010.
- MOODLE (Org.). **Moodle**. Disponível em: <<http://moodle.org/>>. Acesso em: 01 mai. 2010.
- OWRE, Sam (Org.). **PVS Specification and Verification System**. Disponível em: <<http://pvs.csl.sri.com/>>. Acesso em: 4 jun. 2010.
- ROCHA, Heloísa V. **O Ambiente para Ensino a Distância Teleduc**. Disponível em: <http://www.teleduc.org.br/artigos/ccuec_dez_2000_heloisa/sld001.htm>. Acesso em: 28 mai. 2010.
- RODGER, Susan H.. **JFLAP Version 7.0 RELEASED August 28, 2009**. Disponível em: <<http://www.jflap.org/>>. Acesso em: 01 mai. 2010.
- TAYLOR, R. Gregory. **Models of Computation and Formal Languages (Oxford University Press: New York, 1998)**. Disponível em: <<http://home.manhattan.edu/~gregory.taylor/thcomp/>>. Acesso em: 30 abr. 2010.
- TELEDUC. Disponível em: <<http://www.teleduc.org.br/>>. Acesso em: 02 mar. 2010.
- VIEIRA, Luiz Filipe Menezes; VIEIRA, Marcos Augusto Menezes; VIEIRA, Newton José. **Fundamentos da Teoria da Computação: 1º Semestre de 2002**. Disponível em: <<http://homepages.dcc.ufmg.br/~lfvieira/ftc.html>>. Acesso em: 01 mai. 2010.
- VIEIRA, Sonia. **Como elaborar questionários**. São Paulo: Atlas, 2009. 159 p.

WEBCT. Disponível em: < <https://webct.uga.edu/>>. Acesso em: 02 mar. 2010.

APÊNDICE A

INTERFACES PARA CRIAÇÃO DE QUESTÕES ESPECÍFICAS DE TC

A Figura 6.1 apresenta a tela para a criação de uma pergunta em um questionário Moodle (MOODLE, 2010) do tipo Gramática Regular.

Curso Básico de Linguagens Formais Você acessou como A 1 (Sair)

LabLF > Curso Básico de Linguagens Formais > Questionários > Anterior > Modificando um Questionário > Editando uma pergunta de Gramática Regular

Adicionando uma questão de Gramática Regular

Geral

Categoria: Padrão para Curso Básico de Linguagens Formais

Nome da pergunta*: Pergunta 2

Texto da pergunta

Trebuchet 1 (8 pt) Língua

Construa uma GR sob o alfabeto $\{0,1\}$ em que todas as palavras geradas terminem em 0.

Caminho: body

Formato: Formato HTML

Avaliação predefinida*: 1

Fator de penalidade*: 0,1

Feedback geral

Trebuchet 1 (8 pt) Língua

Acertou.

Caminho: body

Rosposta correta. Preencha a resposta a seguir.

Resposta

Resposta

S	→ 0
S	→ T
T	→ T0
T	→ TT
T	→ A

Palavras de teste

0, 1, 10, 11, 10010, 1100, 1010, 1001, 100, 111

Este form contém campos obrigatórios

[Documentação de Moodle relativa a esta página](#)
 Você acessou como A 1 (Sair)

Figura 6.1 – Tela da criação de questões do tipo Gramática Regular.

A Figura 6.2 apresenta a tela para a criação de uma pergunta em um questionário Moodle do tipo Expressão Regular.

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

LabLF > Linguagens Formais 2/2009 > Questionários > Anterior > Modificando um Questionário > Editando uma pergunta de Expressão Regular

Adicionando uma questão de Expressão Regular ?

Geral

Categoria: Padrão para Linguagens Formais 2/2009

Nome da pergunta*: Questao 3

Texto da pergunta ?

Trebuchet 1 (8 pt) Língua B I U S

Construa uma Expressão Regular sob o alfabeto {0,1} e que todas as palavras terminem com um 0.

Caminho: body

Formato ? Formato HTML

Avaliação predefinida* 1

Fator de penalidade* ? 0.1

Feedback geral ?

Trebuchet 1 (8 pt) Língua B I U S

Você acertou.

Caminho: body

Resposta correta. Preencha a resposta a seguir.

Resposta

Resposta

Insira a expressão regular abaixo:

Palavras de teste ?

Este form contém campos obrigatórios

[Documentação de Moodle relativa a esta página](#)

Você acessou como Robson Aguiar (Sair)

Figura 6.2 – Tela da criação de questões do tipo Expressão Regular.

A Figura 6.3 apresenta a tela para a criação de uma pergunta em um questionário Moodle do tipo Autômato Finito com Pilha.

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 com pilha

Adicionando uma questão de Autômato Finito com Pilha

Geral

Categoria: Padrão para Linguagens Formais 2/2009

Nome da pergunta*: Questao 4

Texto da pergunta

Trebuchet 1 (8 pt) Língua B I U S x² x³

Construa um Autômato Finito com Pilha sob o alfabeto {0,1} que reconheça todas as palavras terminadas em 0.

Caminho: body

Formato: Formato HTML

Avaliação predefinida*: 1

Fator de penalidade*: 0.1

Feedback geral

Trebuchet 1 (8 pt) Língua B I U S x² x³

Você acertou.

Caminho: body

Resposta correta. Preencha a resposta a seguir.

Resposta

Resposta

```

    graph LR
      q0((q0)) -- "1, A; A  
0, A; A" --> q0
      q0 -- "0, A; A" --> q1((q1))
      style q0 fill:#ffff00
      style q1 fill:#ffff00
  
```

Palavras de teste

0, 1, 10, 11, 1000, 11111, 10101, 01010

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 6.3 – Tela da criação de questões do tipo Autômato Finito com Pilha.

A Figura 6.4 apresenta a tela para a criação de uma pergunta em um questionário Moodle do tipo Máquina de Turing.

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

LabLF > Linguagens Formais 2/2009 > Questionários > Anterior > Modificando um Questionário > Editando uma pergunta de Máquina de Turing

Adicionando uma questão de Máquina de Turing

Geral

Categoria: Padrão para Linguagens Formais 2/2009

Nome da pergunta*: Questao 5

Texto da pergunta

Trebuchet 1 (8 pt) Língua B I U S x x

Construa uma Máquina de Turing para reconhecer sob o alfabeto {0,1} todas as palavras terminadas em 0.

Caminho: body

Formato: Formato HTML

Avaliação predefinida*: 1

Fator de penalidade*: 0,1

Feedback geral

Trebuchet 1 (8 pt) Língua B I U S x x

Você acertou.

Caminho: body

Rosposta correta. Preencha a resposta a seguir.

Resposta

Resposta

```

graph LR
    Start(( )) --> q0((q0))
    q0 -- "1: 1, R" --> q0
    q0 -- "0: 0, R" --> q0
    q0 -- "0: 0, R" --> q1((q1))
    q1 --> End(( ))
    
```

Palavras de teste

0, 1, 1000, 10101, 0001, 00000, 101011

Este form contém campos obrigatórios

Documentação de Moodle relativa a esta página
 Você acessou como Robson Aguiar (Sair)

Figura 6.4 – Tela da criação de questões do tipo Máquina de Turing.

APÊNDICE B

INTERFACES PARA SUBMISSÃO DE SOLUÇÕES

A Figura 7.1 apresenta a tela para submissão de solução pelo aluno a uma questão em um questionário Moodle (MOODLE, 2010) do tipo Gramática Regular.

The screenshot shows a Moodle question submission page. At the top, it says "Linguagens Formais 2/2009" and "Você acessou como Robson Aguiar (Sair)". Below that is a breadcrumb trail: "LabLF > Linguagens Formais 2/2009 > Questionários". The main heading is "Respondendo Questao 2" with a "Feedback:" label. The question text is: "Construa uma Gramática Regular sob o alfabeto {0,1} em que todas as palavras geradas terminem com um 0." Below the question is a "Resposta" section with a table for defining grammar rules. The table has two rows: the first row has "S" in the first column and "→ 0T" in the second; the second row has "T" in the first column and "→" in the second. Below the table is a large empty text area for the answer. At the bottom of the question area are "Submeter" and "Voltar" buttons. At the very bottom of the page, there is a link for "Documentação de Moodle relativa a esta página", the user name "Você acessou como Robson Aguiar (Sair)", and a search box containing "Linguagens Formais 2/2009".

Figura 7.1 – Tela para submissão de solução de questão do tipo Gramática Regular.

A Figura 7.2 apresenta a tela para submissão de solução pelo aluno a uma questão em um questionário Moodle do tipo Expressão Regular.



The screenshot shows a Moodle interface for a question titled "Respondendo Questao 3". The page header includes "Linguagens Formais 2/2009" and "Você acessou como Robson Aguiar (Sair)". The breadcrumb trail is "LabLF > Linguagens Formais 2/2009 > Questionários". The question is a "Feedback:" type, asking the user to "Construa uma Expressão Regular sob o alfabeto {0,1} em que todas as palavras terminem com um 0." Below the question, there is a "Resposta" label and a large text input area. A smaller input field is labeled "Insira a expressão regular abaixo:" and contains the text "(0+1)". At the bottom of the question area, there are two buttons: "Submeter" and "Voltar". The footer of the page includes a link for "Documentação de Moodle relativa a esta página", the user name "Você acessou como Robson Aguiar (Sair)", and a search box containing "Linguagens Formais 2/2009".

Figura 7.2 – Tela para submissão de solução de questão do tipo Expressão Regular.

A Figura 7.3 apresenta a tela para submissão de solução pelo aluno a uma questão em um questionário Moodle do tipo Máquina de Turing.

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

LabLF > Linguagens Formais 2/2009 > Questionários

Respondendo Questao 5

Feedback:

Construa uma Máquina de Turing para reconhecer sob o alfabeto $\{0,1\}$ todas as palavras terminadas em 0.

Resposta

[Documentação de Moodle relativa a esta página](#)
 Você acessou como Robson Aguiar (Sair)

Figura 7.3 – Tela para submissão de solução de questão do tipo Máquina de Turing.

APÊNDICE C

DETALHAMENTO SOBRE A INTEGRAÇÃO DO JFLAP NO MOODLE

Para que este trabalho possa ser replicado em versões futuras do ambiente Moodle, siga os seguintes passos:

Passo 1: Diretório jflap

Copiar o diretório **jflap** desta versão do trabalho na raiz de diretório da nova versão do ambiente Moodle.

Passo 2: O arquivo funcoes.php

Acessar o diretório **jflap**, abrir o arquivo **funcoes.php** e modificar as variáveis da função **conexão()** (**\$link** e **\$bool**) conforme as configurações de nome, endereço, login e senha de seu Banco de Dados.

Passo 3: Tabelas de questões especiais

Acessar o Banco de Dados e criar 5 tabelas, uma para cada tipo de questão especial. Os cinco **scripts** estão listados abaixo.

script: Criar tabela afdafnd

```
CREATE TABLE `mdl_question_afdafnd` (  
  `id` bigint(10) unsigned NOT NULL auto_increment,  
  `question` bigint(10) unsigned NOT NULL default '0',  
  `answers` text NOT NULL,  
  `usecase` tinyint(2) NOT NULL default '0',
```

```

PRIMARY KEY (`id`),
KEY `mdl_quesshor_que_ix` (`question`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='Options for afd afnd questions'
AUTO_INCREMENT=192 ;

```

script: Criar tabela afp

```

CREATE TABLE `mdl_question_afp` (
  `id` bigint(10) unsigned NOT NULL auto_increment,
  `question` bigint(10) unsigned NOT NULL default '0',
  `answers` text NOT NULL,
  `usecase` tinyint(2) NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `mdl_quesshor_que_ix` (`question`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='Options for afp questions'
AUTO_INCREMENT=158 ;

```

script: Criar tabela er

```

CREATE TABLE `mdl_question_er` (
  `id` bigint(10) unsigned NOT NULL auto_increment,
  `question` bigint(10) unsigned NOT NULL default '0',
  `answers` text NOT NULL,
  `usecase` tinyint(2) NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `mdl_quesshor_que_ix` (`question`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='Options for afd afnd questions'
AUTO_INCREMENT=170 ;

```

script: Criar tabela g

```

CREATE TABLE `mdl_question_g` (
  `id` bigint(10) unsigned NOT NULL auto_increment,
  `question` bigint(10) unsigned NOT NULL default '0',
  `answers` text NOT NULL,
  `usecase` tinyint(2) NOT NULL default '0',

```

```

PRIMARY KEY (`id`),
KEY `mdl_quesshor_que_ix` (`question`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='Options for g questions'
AUTO_INCREMENT=209 ;

```

script: Criar tabela mt

```

CREATE TABLE `mdl_question_mt` (
  `id` bigint(10) unsigned NOT NULL auto_increment,
  `question` bigint(10) unsigned NOT NULL default '0',
  `answers` text NOT NULL,
  `usecase` tinyint(2) NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `mdl_quesshor_que_ix` (`question`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='Options for afp questions'
AUTO_INCREMENT=162 ;

```

Passo 4: Arquivo quiz.php

Acessar o diretório `/moodledata/lang/pt_br_utf8` e abrir o arquivo **quiz.php**. Neste arquivo insira as linhas abaixo para especificar o significado das *strings* abreviadas que representam os tipos de questões especiais.

```

$string['afdafnd'] = 'Autômato Finito Determinístico e Não-Determinístico';
$string['afp'] = 'Autômato Finito com Pilha';
$string['er'] = 'Expressão Regular';
$string['g'] = 'Gramática Regular';
$string['mt'] = 'Máquina de Turing';

```

Passo 5: Diretório question/type

Este diretório contém os tipos de questões que estão implementados no Moodle. Assim, copie os diretórios de nomes `afdafnd`, `afp`, `er`, `mt` e `g` para o diretório `/question/type` em sua versão do Moodle.

Passo 6: Arquivo questiontype.php

Acessar o diretório `/question/type` e copie o arquivo **questiontype.php** para a nova versão do Moodle, pois ele contém código modificado para exibir os applets do Jflap no histórico das tentativas do aluno em responder uma questão especial.

Passo 7: Arquivo preview.php

Acessar o diretório `/question` e copiar o arquivo **preview.php** para a nova versão do Moodle.

Passo 8: Arquivo lib.php

Acessar o diretório `/course` e copiar o arquivo **lib.php** para a nova versão do Moodle.

Passo 9: Arquivo questionlib.php

Acessar o diretório `/lib` e copiar o arquivo **questionlib.php** para a nova versão do Moodle.

Após as modificações feitas você poderá utilizar os novos tipos de questões específicas de TC em sua versão Moodle.

APÊNDICE D

CRONOGRAMA

O cronograma (Tabela 9.1) apresentado neste apêndice é utilizado para a disciplina de Linguagens Formais no curso básico. Nada impede que o professor realize adaptações para outras disciplinas de TC.

Tabela 9.1 – Cronograma.

Aula	Conteúdo	Matéria de aula	Outros materiais
01	APRESENTAÇÃO DA DISCIPLINA	-	-
02	CONCEITOS MATEMÁTICOS BÁSICOS	LF_aula01_conceitos_basicos.ppt	Lista-1.pdf
03	LINGUAGENS REGULARES E AUTÔMATO FINITO	LF_aula02_AF.ppt	-
04	DESIGN DE AFDs	LF_aula03_design_AFD_AFND.ppt	Lista-2.pdf
05	AULA ESPECIAL DE "TIRAR DÚVIDAS"	-	-
06	MINIAVALIAÇÃO 1 (realizada em meia aula)	LF_aula03_design_AFD_AFND.ppt	Lista-3.pdf
	DESIGN DE AFNDs		
07	EQUIVALÊNCIA ENTRE AFDs E AFNDs	LF_aula04_equiv_AFD_AFND.ppt	-
08	EQUIVALÊNCIA ENTRE AFDs E ϵ -AFNDs	LF_aula05_equiv_AFD_eAFND.ppt	Lista-4.pdf
09	AULA ESPECIAL DE "TIRAR DÚVIDAS"	-	-
10	MINIAVALIAÇÃO 2 (realizada em meia aula)	LF_aula06_ER.ppt	
	EXPRESSÕES REGULARES		
11	EQUIVALÊNCIA ENTRE ER E AF	LF_aula07_equiv_ER_AF.ppt	Lista-5.pdf
12	MINIMIZAÇÃO DE AFDS	LF_aula08_minimizacao.ppt	Lista-6.pdf

13	AULA ESPECIAL DE "TIRAR DÚVIDAS"	-	-
14	MINIAVALIAÇÃO 3 (realizada em aula inteira)		
15	LEMA DO BOMBEAMENTO E LINGUAGENS LIVRES DE CONTEXTO: GRAMÁTICAS LIVRES DE CONTEXTO, ÁRVORES DE ANÁLISE SINTÁTICA, AMBIGUIDADE.	LF_aula09_bombeamento.ppt LF_aula10_GLC.ppt	
16	FORMA NORMAL DE CHOMSKY	LF_aula11_FNC.ppt	Lista-7.pdf
17	AULA ESPECIAL DE "TIRAR DÚVIDAS"	-	-
18	MINIAVALIAÇÃO 4 (realizada em aula inteira)		
19	AUTÔMATOS DE PILHA	LF_aula12_PDA.ppt	
20	EQUIVALÊNCIA ENTRE PDAs E GLCs	LF_aula12_PDA.ppt	Lista-7.pdf
21	AULA ESPECIAL DE "TIRAR DÚVIDAS"	-	-
22	MINIAVALIAÇÃO 5 (realizada em aula inteira)		
23	LINGUAGENS SENSÍVEIS AO CONTEXTO	LF_aula13_Maquinas_Turing_I.ppt	
24	LINGUAGENS RECURSIVAMENTE ENUMERÁVEIS E RECURSIVAS	LF_aula14_Maquinas_Turing_II.ppt	Lista-8.pdf
25	AULA ESPECIAL DE "TIRAR DÚVIDAS"	-	-
26	MINIAVALIAÇÃO 6 (realizada em aula inteira)		
27	PREPARAÇÃO DO SEMINÁRIO		
28	PREPARAÇÃO DO SEMINÁRIO		
29	Seminário em grupo: - Grupo 1 - Grupo 2 - Grupo 3		
30	Seminário em grupo - Grupo 4 - Grupo 5 - Grupo 6		
31	RESULTADOS		

APÊNDICE E

INTERFACE PRINCIPAL DO CURSO BÁSICO

A interface principal do curso básico¹⁰ quando se acessa como professor pode ser vista na Figura 10.1.

¹⁰ Realize o *backup* em sua versão do Moodle do arquivo “backup-cblf-20101226-0839.zip” disponível no CD desta dissertação. Utilize o login “mestre” e senha “professor” para acesso.

Curso Básico de Linguagens Formais

Participantes: Participantes

Atividades: Atividades

Pesquisar nos Fóruns: Pesquisar nos Fóruns

Administração:

- Adicionar edição
- Configurações
- Designar funções
- Notas
- Grupos
- Backup
- Restaurar
- Importar
- Reconfigurar
- Relatórios
- Perguntas
- Atividades
- Cancelar a minha inscrição no curso
- CELF
- Fórum

Meus cursos: Meus cursos

Modo função: parâmetro | Abrir edição

Últimas Notícias

Próximos Eventos

Atividade recente

Agenda do Curso

Bem-vindo ao Curso Básico de Linguagens Formais

Professor responsável: A1

Nesta seção você pode encontrar o plano da disciplina, o cronograma e um arquivo contendo suas notas e frequência.

- Fórum de notícias
- Plano da disciplina
- Cronograma da disciplina
- Notas e frequência

2 junho - 9 junho

Esta seção disponibiliza os arquivos em formato PDF dos slides apresentados em sala sobre conceitos básicos e AFDs e listas de exercícios para a **Miniavaliação 1**.

- Aula 01: Aula destinada à apresentação da disciplina.
- Aula 02: Conceitos matemáticos básicos.
- Lista de exercícios: Conceitos básicos e gramática.
- Aula 03: Linguagens Regulares e Autômato Finito.
- Aula 04: Design de AFDs.
- Lista de exercícios: Construção de AFDs.
- Aula 05: Aula especial sobre AFDs.

Miniavaliação 1

10 junho - 19 junho

Esta seção disponibiliza os arquivos em formato PDF dos slides apresentados em sala sobre Construção de AFNDs e equivalência entre AFNDs e AFDs e listas de exercícios para a **Miniavaliação 2**.

- Aula 06: Design de AFNDs.
- Lista de exercícios: Construção de AFNDs.
- Aula 07: Equivalência entre AFNDs e AFDs.
- Aula 08: Equivalência entre AFDs e e-AFDs.
- Lista de exercícios: Conversão AFND-AFD.
- Aula 09: Aula especial sobre AFNDs.

Miniavaliação 2

21 junho - 22 junho

Esta seção disponibiliza os arquivos em formato PDF dos slides apresentados em sala sobre Exp. Regular, Equivalência ER-AF e Minimização e listas de exercícios para a **Miniavaliação 3**.

- Aula 10: Expressões Regulares.
- Aula 11: Equivalência entre ER e AF.
- Lista de exercícios: ER e Conversão ER-AF.
- Aula 12: Minimização de AFDs.
- Lista de exercícios: Minimização.
- Aula 13: Aula especial sobre ER.

Miniavaliação 3

29 junho - 30 junho

Esta seção disponibiliza os arquivos em formato PDF dos slides apresentados em sala sobre Lema do Bombeamento e Forma Normal de Chomsky e listas de exercícios para a **Miniavaliação 4**.

- Aula 14: Lema do Bombeamento.
- Aula 15: Anore de Análise sintática.
- Aula 16: Forma Normal de Chomsky.
- Lista de exercícios: FNC, GLC, PDA.
- Aula 17: Aula especial de Gramática.

Miniavaliação 4

1 julho - 7 julho

Esta seção disponibiliza os arquivos em formato PDF dos slides apresentados em sala sobre Autômatos com Pilha e Conversão PDAs-GLC e listas de exercícios para a **Miniavaliação 5**.

- Aula 18: Autômatos com Pilha.
- Aula 19: Autômatos com Pilha.
- Aula 20: Conversão PDA-GLC.
- Lista de exercícios: FNC, GLC, PDA.
- Aula 21: Aula especial sobre PDA.

Miniavaliação 5

08 julho - 14 julho

Esta seção disponibiliza os arquivos em formato PDF dos slides apresentados em sala sobre Máquinas de Turing e listas de exercícios para a **Miniavaliação 6**.

- Aula 22: Linguagem Sensível ao Contexto.
- Aula 23: Linguagem Sensível ao Contexto.
- Aula 24: Linguagem Recursivamente Enumerável.
- Lista de exercícios: Máquinas de Turing.
- Aula 25: Aula especial sobre MT.

Miniavaliação 6

15 julho - 21 julho

Esta seção é relativa aos seminários.

Apresentações da Aula 28

Duplas	Tema
Aluno 1 e 2	Jogo da vida - Autômatos celulares
Aluno 3 e 4	YACC
Aluno 5 e 6	Máquina de Roldignastres
Aluno 7 e 8	Autômatos de Transitão

Apresentações da Aula 30

Duplas	Tema
Aluno 9 e 10	Máquina de Moore
Aluno 11 e 12	Gramática de Grefos
Aluno 13 e 14	Máquina de Mealy
Aluno 15 e 16	Formalismos para cursos Web

- Orientações para o seminário.
- Aula 26: Aula destinada à preparação do seminário.
- Aula 27: Aula destinada à preparação do seminário.
- Aula 28: Aula destinada à preparação do seminário.
- Submeta aqui os slides

22 julho - 29 julho

Esta seção contém links para materiais didáticos diversos úteis para sua aprendizagem.

Apostilas

- Compiladores
- Apostila de Linguagens Formais e Autômatos.
- Apostila de Linguagens Formais e Teoria da Computação UNESP
- Conceitos elementares de Teoria da Computação.
- Linguagens Formais

Artigos

- Animações computacionais por meio de Autômatos Finitos com Salida.
- A crise nos fundamentos da Matemática e a Teoria da Computação.
- Uso de simulação para o ensino de Computabilidade.
- Autômatos Finitos um Formalismo para Cursos na Web.
- Autômatos celulares.

Ferramentas computacionais

- Language Emulator
- VAS
- JFLAP

Procure também na internet pelas ferramentas:

- e GCMF (Sistema de Criação e Teste de Modelos Formais) e
- e GAW (Gíria Abstract Machine).

Outros

Insera links para outros materiais que julgar importantes para os alunos.

29 julho - 4 agosto

5 agosto - 11 agosto

Documentação de Moodle relativa a esta página

Você acessou como A1 (Sai)

Home Page

Figura 10.1 – Tela principal do curso básico.

Para melhor entendê-la dividiu-se em seções numeradas de 1 a 9 entre parênteses e envoltas, cada uma, por um retângulo com bordas em vermelho. Assim, a Figura 10.2 apresenta a seção (1) em que o usuário pode acessar o plano da disciplina e seu cronograma, bem como acessar um arquivo com suas notas e frequência na disciplina.

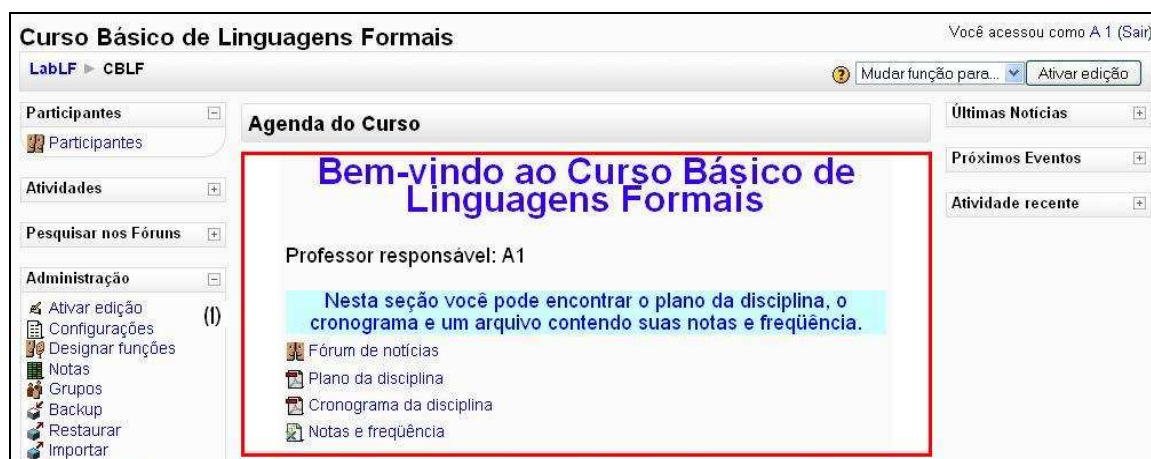


Figura 10.2 – Seção inicial do curso.

A Figura 10.3 apresenta a seção (2) com os materiais de aula disponíveis para *download* e listas de exercícios para a Miniavaliação 1 que ocorrerá na aula seguinte a “Aula 05: Aula especial sobre AFDs.”.

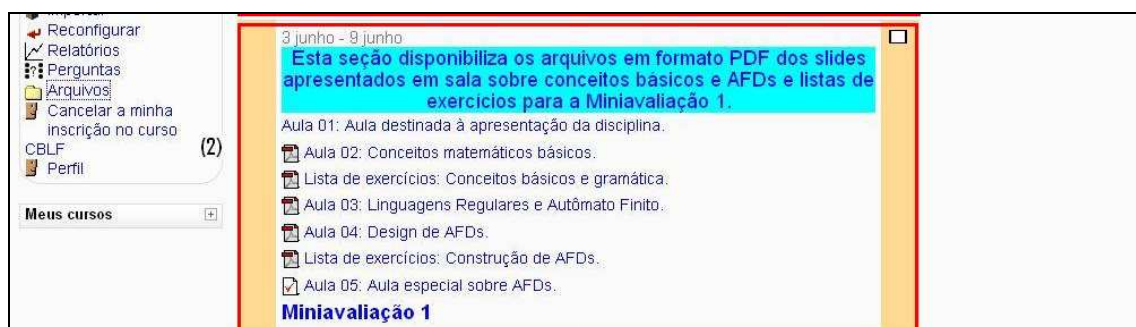


Figura 10.3 – Material para a Miniavaliação 1.

A Figura 10.4 apresenta a seção (3) com os materiais de aula disponíveis para *download* e listas de exercícios para a Miniavaliação 2 que ocorrerá na aula seguinte a “Aula 09: Aula especial sobre AFNDs.”.

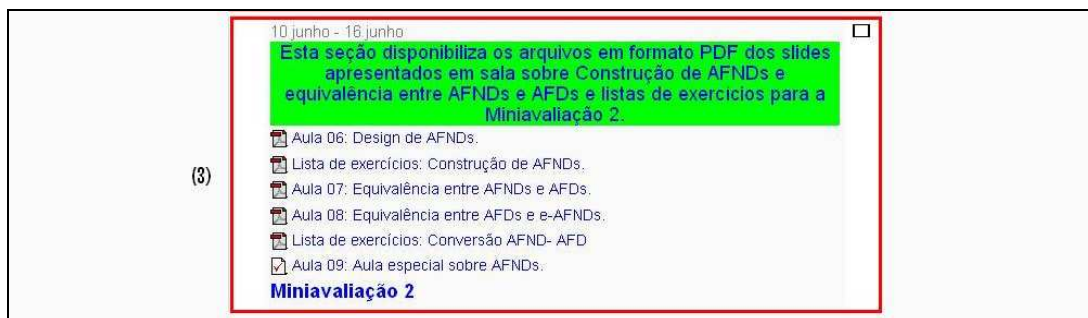


Figura 10.4 – Material para a Miniavaliação 2.

Para as seções (4), (5), (6) e (7) a estrutura apresentada na Figura 10.4 se repete para a realização das Miniavaliações 3, 4, 5 e 6, respectivamente. A Figura 10.5 apresenta a seção (8) relativa aos seminários, onde o professor poderá inserir:

- os nomes dos alunos e os temas que apresentarão no seminário da Aula 29 em (A) e da Aula 30 em (B) do cronograma (ver Apêndice D);
- um *link* (visto em (C)) ao arquivo “seminário.pdf” (disponibilizado na pasta “Documentos_auxiliares”) contendo as orientações para o seminário; e
- um *link* (visto em (D)) para submissão do material apresentado no seminário, conforme planejado no arquivo “seminário.pdf”.



Figura 10.5 – Os seminários.






A seção (9) vista na Figura 10.6 apresenta *links* internos e externos para apostilas, artigos e ferramentas computacionais.

(9)






22 julho - 28 julho

Esta seção contém links para materiais didáticos diversos úteis para sua aprendizagem.




Apostilas

-  Compiladores
-  Apostila de Linguagens Formais e Autômatos.
-  Apostila de Linguagens Formais e Teoria da Computação UNESP
-  Conceitos elementares de Teoria da Computação.
-  Linguagens Formais

Artigos

-  Animações computacionais por meio de Autômatos Finitos com Saída.
-  A crise nos fundamentos da Matemática e a Teoria da Computação.
-  Uso de simulação para o ensino de Computabilidade.
-  Autômatos Finitos um Formalismo para Cursos na Web.
-  Autômatos celulares.

Ferramentas computacionais

-  Language Emulator
-  VAS
-  JFLAP

Procure também na internet pelas ferramentas:

- o SCTMF (Sistema de Criação e Teste de Modelos Formais) e
- o GAM (Ginix Abstract Machine).

Outros

Insira links para outros materiais que julgar importantes para os alunos.

28 julho - 4 agosto

5 agosto - 11 agosto


 Documentação de Moodle relativa a esta página
 Você acessou como A.1 (Sair)

Figura 10.6 – Links a materiais didáticos.

APÊNDICE F

QUESTIONÁRIO PARA PROFESSORES DE TC

Questionário

Este questionário se propõe a coletar informações sobre o ensino e aprendizagem das disciplinas de Teoria da Computação no âmbito das 10 melhores universidades brasileiras segunda a Info Exame.

Apenas os professores ministrantes destas disciplinas nestas universidades deverão responder a este questionário.

O questionário deve ser enviado para robson_cbcc@yahoo.com.br.

Esta pesquisa só necessita de alguns poucos minutos. Agradeço desde já sua compreensão.

1. Quanto à distribuição das disciplinas da linha de Teoria da Computação em sua universidade elas fazem parte da grade de cursos de:

- Graduação.
 - Mestrado.
 - Doutorado.
 - Outros. Qual(is)?
-
-
-

2. Assinale quais disciplinas a sua universidade oferece para a linha de Teoria da Computação?

- Linguagens Formais e Autômatos.
- Teoria da Computação.

Computabilidade.

Análise de Algoritmos.

Outras. Qual(is)?

3. Quais destas disciplinas são obrigatórias e optativas?

4. Quais disciplinas você ministra e a quantos semestres?

5. Qual o número médio de alunos em cada uma de suas turmas?

6. Em média qual o índice de reprovação de cada uma de suas disciplinas?

7. Quanto às tarefas e atividades realizadas na disciplina, como é o seu processo de avaliação? (Ex.: Realiza três provas individuais e trabalhos).

APÊNDICE G

PERGUNTAS DA ENTREVISTA SEMIESTRUTURADA

Questionário

Estas são as questões a serem feitas pelo entrevistador Robson da Silva Aguiar a cada um dos alunos da disciplina de Linguagens Formais 2010/2 ministrada pela professora Janne Y. Y. Oeiras na Universidade Federal do Pará.

O objetivo desta entrevista é coletar informações a cerca adequação do curso básico de TC à aprendizagem de TC.

1. De maneira geral a disciplina supriu suas necessidades sobre Linguagens Formais?

2. O acompanhamento dado pelo(a) professor(a) e pelo(s) monitor(es) foi adequado? Justifique.

3. A inserção de um ambiente para auxílio a sua aprendizagem no que concerne à verificação de uma solução a um problema proposto lhe ajudou:

Pouco

Razoavelmente

Muito

4. Quanto a sua compreensão, é mais fácil vislumbrar uma solução a um problema qualquer com os conceitos aprendidos na disciplina? Justifique.

5. O feedback dado pelo ambiente ao submeter uma solução a um exercício livre lhe ajuda a saber se sua solução está correta? Justifique.

6. Numa escala de 0 (zero) a 10 (dez) quanto você supõe que os trabalhos envolvendo o ambiente lhe ajudaram a resolver as questões das avaliações? Justifique.

7. Qual(is) assunto(s) estudado(s) nesta disciplina faltou(aram) um acompanhamento maior? Justifique.

8. Sugira uma melhor forma de acompanhamento do aprendizado em Teoria da Computação para turmas futuras.

9. Para os alunos que estão realizando a disciplina por mais de uma vez, aponte possíveis dificuldades para seu insucesso na disciplina em semestres anteriores.

10. Para os alunos que estão realizando a disciplina por mais de uma vez, aponte os pontos fortes e fracos da mudança em relação à sua avaliação (6 miniavaliações em vez de apenas 2 avaliações) na disciplina.
