

**UNIVERSIDADE DE CAXIAS DO SUL**  
**Centro de Ciências Exatas e Tecnologia**  
**Curso de Bacharelado em Ciência da Computação**

**Matheus Zenato**

**AMBIENTE DE APRENDIZAGEM DE PROGRAMAÇÃO DE  
COMPUTADORES**

**Caxias do Sul**

**2009**

**Matheus Zenato**

**AMBIENTE DE APRENDIZAGEM DE PROGRAMAÇÃO DE  
COMPUTADORES**

Trabalho de Conclusão de Curso  
para obtenção do Grau de  
Bacharel em Ciência da  
Computação da Universidade de  
Caxias do Sul.

**Carine Geltrudes Webber  
Orientador**

**Caxias do Sul**

**2009**

**Dedico este trabalho a todos  
os alunos e professores que tenham  
interesse em desenvolver novas ferramentas  
que auxiliem a aprendizagem .**

## **AGRADECIMENTOS**

Agradeço ao apoio da minha família porque sem eles nada disso seria possível. A compreensão dos amigos que muito apoiaram. Aos novos amigos e colegas que acabei conhecendo durante esse longo processo de aprendizado. Aos professores pelos seus ensinamentos e apoios em especial à orientadora Carine Webber pelas cobranças e sugestões o que proporcionaram um maior empenho e organização durante as atividades.

## RESUMO

A área da Inteligência Artificial aplicada à Educação busca a utilização das técnicas de Inteligência Artificial para a resolução de problemas educacionais a fim de propiciar ferramentas que auxiliem os alunos e professores na construção de conhecimento. Dentre as áreas das quais a pesquisa em Inteligência Artificial aplicada à Educação envolve merece destaque o estudo dos ambientes de aprendizagem inteligentes. Esse documento apresenta um estudo sobre os ambientes virtuais de aprendizagem e uma análise sobre alguns ambientes existentes a fim de levantar requisitos e funcionalidades para a elaboração de uma interface à um ambiente virtual de aprendizagem em programação de computadores. E, desta forma, desenvolver uma ferramenta que possa minimizar as dificuldades na aprendizagem desta disciplina. Para a elaboração desse projeto seguiu-se uma metodologia baseada em quatro fases sendo a primeira o estudo e a análise da problemática aprendizagem de algoritmos através de ambientes virtuais, a segunda sendo a modelagem da interface e suas funcionalidades, a terceira fase contemplando a construção e a fase final contemplando a realização de testes. A interface se integrará a um sistema de análise de algoritmos a fim de fornecer ao professor um diagnóstico sobre os pontos fracos do aluno referentes ao conteúdo em questão. Por se tratar de um ambiente destinado a construção de conhecimento, esse ambiente buscou agregar recursos de interface que facilitem a sua utilização, por causa disso, o estudo de conceitos de ergonomia e usabilidade também se fez presente o que possibilitou se chegar a uma solução de fácil manuseio e que não cause grandes transtornos ao utilizador.

**Palavras-chaves:** Ambiente de aprendizagem, Interface humano computador, Ferramentas de Interação

## **ABSTRACT**

The area of Artificial Intelligence applied to education aim to use of Artificial Intelligence techniques for solving educational problems in order to provide tools that help students and teachers in knowledge construction. Among the areas in which research in Artificial Intelligence applied to education involves the study undertaken for intelligent learning environments. This paper presents a study on virtual learning environments and an analysis of some existing environments in order to lift requirements and capabilities for the development of an interface to a virtual learning environment in computer programming. And thus, develop a tool that can minimize the difficulties in learning this discipline. In carrying out this project followed a methodology based on four phases with the first study and analysis of the problems of learning algorithms through virtual environments, the second being the modeling of interface and functionality, the third stage contemplating the construction phase and contemplating the final testing. The interface will integrate a system of analysis of algorithms to provide the teacher a diagnosis of the weaknesses of the student regarding the content in question. Because it is an environment for knowledge construction, the environment sought to add interface features that facilitate its use, because of this, the study of concepts of ergonomics and usability was also present which made it possible to reach a solution easy to use and does not cause major disruptions to the user.

**Keywords:** Environment of learning, Human computer interface, Interaction tools

## LISTA DE ILUSTRAÇÕES

Ilustração 1: Relação dos Componentes de um STI.....	20
Ilustração 2: Interface do sistema Aplusix.....	24
Ilustração 3: Interface do sistema AS-MCOE.....	25
Ilustração 4: Interface do Projeto AMBAP.....	26
Ilustração 5: Interface do Visual Automata Simulator no modo de tratamento de Autômatos.....	27
Ilustração 6: Interface do Visual Automata Simulator no modo de tratamento de Máquina de Turing.....	28
Ilustração 7: Interface do sistema ELP.....	29
Ilustração 8: Interface do AMPLIA mostrando informações para a construção das redes.....	31
Ilustração 9: Interface do AMPLIA no modo de construção da rede.....	32
Ilustração 10: Interface do Ambiente Moodle.....	33
Ilustração 11: Interface do Ambiente Moodle - Editor de texto.....	34
Ilustração 12: Interface do ambiente SOLAR.....	35
Ilustração 13: Interface do Ambiente SOLAR - Demonstração de Conteúdo de aula.....	35
Ilustração 14: Interface do Ambiente TelEduc.....	36
Ilustração 15: Interface do Ambiente TelEduc - Fórum.....	37
Ilustração 16: Interface do Ambiente Amadeus - Pagina Principal.....	38
Ilustração 17: Interface do Ambiente Amadeus - Gestão de Conteúdo.....	39
Ilustração 18: Caso de Uso Referente aos Usuários.....	46
Ilustração 19: Caso de Uso referente ao Aluno.....	47
Ilustração 20: Caso de Uso referente ao professor.....	50
Ilustração 21: Caso de Uso Referente ao Administrador.....	54
Ilustração 22: Interface Autenticação.....	56
Ilustração 23: Exemplo de Interface para o papel Aluno.....	57
Ilustração 24: Exemplo de Interface para o papel Professor.....	58
Ilustração 25: Exemplo de Interface para o papel Administrador.....	59
Ilustração 26: Interface do Mural de Recados.....	60
Ilustração 27: Interface do Janela de Conversa.....	61
Ilustração 28: Manutenção do Conteúdo Disciplinar.....	62
Ilustração 29: Interface para o desenvolver o exercício.....	63
Ilustração 30: Interface de Acompanhamento de Aprendizado.....	64
Ilustração 31: Protótipo de Interface - Perfil de Usuário.....	64
Ilustração 32: Protótipo de Interface - Modelo de Mensagem de Erro (Vermelho).....	65
Ilustração 33: Protótipo de Interface - Modelo de Mensagem de Atenção (Laranja).....	65
Ilustração 34: Protótipo de Interface - Modelo de Mensagem de Informação (Azul).....	66
Ilustração 35: Modelo de Estrutural do Sistema.....	68
Ilustração 36: Representação do Padrão MVC.....	71
Ilustração 37: Modelo Estrutura da Interface.....	73
Ilustração 38: Modelo de Comunicação.....	75
Ilustração 39: Diagrama de Classes do Ambiente.....	77
Ilustração 40: Tela de Cadastro de Turma x Professor.....	79
Ilustração 41: Tela de Adição de Conteúdo.....	80
Ilustração 42: Tela de Cadastro de Exercícios.....	81

Ilustração 43: Tela de Cadastro de Aluno.....	82
Ilustração 44: Tela de Criação de Mensagem.....	83
Ilustração 45: Tela de Resolução de Exercício.....	84
Ilustração 46: Tela de Acompanhamento de Aprendizado.....	85
Ilustração 47: Tela de Atribuição de Parecer.....	85
Ilustração 48: Respostas do Questionário Referentes a Classificação.....	87
Ilustração 49: Respostas das Questões referentes à Aspectos Pedagógicos.....	87
Ilustração 50: Respostas das Questões referentes à Aspectos Técnicos.....	88
Ilustração 51: Respostas das Questões referentes à Aspectos Técnicos.....	89
Ilustração 52: Respostas referente ao espaço para Comentários e Sugestões.....	89

## LISTA DE TABELAS

Tabela 1: CAI x STI Fonte : GIRAFFA (1997, p. 19).....	19
Tabela 2: Descrição da Funcionalidade Autenticar.....	46
Tabela 3: Descrição da Funcionalidade Solicitar Ajuda.....	47
Tabela 4: Descrição da Funcionalidade Resolver Exercícios.....	48
Tabela 5: Descrição da Funcionalidade Solicitar Análise.....	49
Tabela 6: Descrição da Funcionalidade Acessar Material.....	49
Tabela 7: Descrição da Funcionalidade Elaborar Material Didático.....	51
Tabela 8: Descrição da Funcionalidade Auxiliar Aluno.....	51
Tabela 9: Descrição da Funcionalidade Acompanhar Aprendizado.....	52
Tabela 10: Descrição da Funcionalidade Manter Exercícios.....	53
Tabela 11: Relação Utilizada pelo SIA-TP que relaciona Área de Conhecimento com o Conteúdo de aprendizagem.....	96
Tabela 12: Classificação dos sinais de perigo em cada uma das três áreas de aprendizagem da disciplina de algoritmo de acordo com a área de conhecimento que o aluno se encontra.....	96

## LISTA DE ABREVIATURAS E SIGLAS

<b>Sigla</b>	<b>Significado em Português</b>	<b>Significado em Inglês</b>
IA	Inteligência Artificial	
AVA	Ambiente Virtual de Aprendizagem	
STI	Sistemas Tutores Inteligentes	
CAI	Instrução/tutoria assistida ou apoiada por computador	Computer assisted instruction
XML	Linguagem Extensível de Formatação	Extensible Markup Language
MXML	Linguagem de Marcação Baseada em XML	Magic Extensible Markup Language
PHP		Hypertext Preprocessor
J2EE		Java Enterprise Edition
RPC		Remote procedure call
CORBA		Common Object Request Broker Architecture
AS3		ActionScript 3
RTMP		Real Time Messaging Protocol

## SUMÁRIO

1	Introdução.....	14
1.1	Objetivos.....	16
1.2	Organização do Documento.....	17
2	Ambientes de Aprendizagem.....	18
2.1	Inteligência Artificial Aplicada à Educação.....	18
2.2	Processo de Aprendizagem .....	21
2.3	Resolução de Problemas e a Programação de Computadores.....	21
2.4	Exemplos de Ambientes de Aprendizagem.....	23
2.4.1	Ambiente da Matemática:.....	23
2.4.2	Ambiente Infante-Juvenil.....	24
2.4.3	Ambientes em Programação de Computadores.....	25
2.4.4	Ambiente Médico.....	30
2.4.5	Ambientes de Gerência de Cursos:.....	32
2.4.6	Conclusões sobre os Ambientes Analisados.....	39
2.5	Dificuldades nos ambientes.....	41
2.6	Ergonomia e Usabilidade.....	42
3	Ambiente DE Aprendizagem DE Programação de Computadores.....	45
3.1	Requisitos do Ambiente.....	45
3.2	Interface e Funcionalidades.....	54
3.2.1	Componente de Autenticação.....	55
3.2.2	Componente Ambiente Caderno de Exercícios.....	56
3.2.3	Perfil.....	64
3.2.4	Componentes Auxiliares.....	65
3.3	Padrões de Arquitetura.....	66
3.3.1	Plataforma Adobe Flex.....	66
3.3.2	Plataforma JAVA.....	67
3.3.3	Protocolo RTMP.....	67
3.3.4	Servidor de Aplicação JBOSS.....	67
3.3.5	Aplicação J2EE.....	68
3.3.6	Empacotamento.....	69
3.3.7	Arquitetura.....	69
3.3.8	Requisitos de Sistema.....	69
3.3.9	Ferramentas de Desenvolvimento.....	70
3.3.10	Padrões de Projeto.....	70
3.3.11	Modelagem do Projeto.....	71
3.3.12	Estrutura do StudentBookWeb.....	72
3.3.13	Estrutura do STUDENTBOOKCLIENT.....	73

3.3.14ESTRUTURA DO STUDENTBOOKEJB.....	74
3.3.15ESTRUTURA DO STUDENTBOOKENTITY.....	75
3.3.16StudentBookEAR.....	77
4 Cenário de Utilização do Ambiente.....	79
5 Testes e Avaliação.....	86
6 Conclusão.....	91
6.1 Síntese.....	91
6.2 Contribuições do Trabalho.....	91
6.3 Perspectiva de Trabalhos Futuros.....	92
7 Referências.....	93
8 Anexos.....	96
8.1 Anexo A.....	96
8.2 Anexo B.....	97



# 1 INTRODUÇÃO

A área da Inteligência Artificial aplicada à Educação busca a utilização das técnicas de Inteligência Artificial para a resolução de problemas educacionais a fim de desenvolver ferramentas que auxiliem os alunos e professores na construção de conhecimento. Segundo ROSATELLI (2000), sistemas oriundos da área Inteligência Artificial Aplicada a Educação são sistemas destinados aos processos de ensino e aprendizagem que tem algum grau de tomada de decisão autônoma em relação às suas interações com os usuários. Esse processo de decisão é necessariamente feito durante as interações do sistema com os usuários e, geralmente, o sistema precisa acessar vários tipos de conhecimento e processos de raciocínio para que tais decisões sejam tomadas.

Dentre os paradigmas da Inteligência Artificial aplicada à Educação, merece destaque o estudo dos ambientes de aprendizagem inteligentes. Esses ambientes buscam criar situações de aprendizagem através de sucessivas interações com o usuário. Estas interações estão, contudo, restritas à natureza do conteúdo a ser ensinado, ao contexto operacional, e às características individuais do usuário. Tais restrições influenciam na concepção de interfaces multi-modais que possam dinamicamente e em tempo real suportar o modo de interação melhor adaptado ao usuário (WEBBER,2008).

Trazer a tarefa de ensino e aprendizagem nas áreas complexas aos ambientes virtuais de aprendizagem tem sido um desafio para diversos trabalhos e pesquisas acadêmicas. Devido às inúmeras interpretações que podem ser geradas pelos aprendizes e soluções que podem ser produzidas torna-se difícil desenvolver um ambiente dinâmico que auxilie a aprendizagem do aluno. A programação de computadores encontra-se nessa classificação por ser uma disciplina onde o aprendizado depende da resolução de problemas.

Segundo MALTEMPI e VALENTE (2000), a maioria dos alunos que chegam às universidades apresenta deficiência relacionada com raciocínio lógico e criatividade. Isso porque os mesmos são provenientes de um sistema educacional que privilegia e exige a memorização e reprodução de informações. Tais características praticamente inviabilizam o pensamento crítico e criativo. A matéria de Programação de Computadores faz parte do currículo dos cursos da área de Computação e Informática, e de vários cursos de Engenharia. Programar é uma atividade que tem como fundamento o desenvolvimento de algoritmos, que é a parte criativa do processo de programação, ficando a execução para ser realizada pelo computador. A tarefa de construir

algoritmos inibe a memorização e reprodução de informações e requer raciocínio lógico, reflexão, pesquisa e envolvimento do aluno com o problema a ser modelado.

Ensinar programação não é apenas ensinar uma linguagem de programação. Este ensino envolve entender problemas e descrever formas de resolução, de maneira imparcial, para que então sejam codificadas em uma linguagem. Ou seja, somente após o aprendizado dos conceitos de algoritmo e fundamentos de lógica, o estudante pode travar contato com uma linguagem de programação.

O aprendizado da programação de computadores tem sido feito a partir do uso dos conceitos ensinados em sala de aula e o auxílio dos compiladores / interpretadores. O aluno aplica os conceitos aprendidos, normalmente conceituais e teóricos, num ambiente desenvolvido com o propósito de desenvolvimento de software (compiladores). De certa forma, esses ambientes dificultam a aprendizagem e baixam a auto-estima do aluno que passa a ter de conhecer a linguagem que o compilador / interpretador utiliza e aprender na tentativa e erro. Esse tipo de método já se provou não ser eficiente visto que diversos alunos não compreenderam a lógica necessária para chegar na solução apenas obtiveram o resultado desejado perante a experimentos (método de tentativa-e-erro).

Diversos projetos de pesquisa e trabalhos relacionados foram concebidos a fim de atender a essas necessidades cita-se alguns como Projeto Modal, Projeto IAdapt, Projeto ModelGroup e SIA-TP, porém em nenhum deles a interface com o usuário teve grande ênfase de forma que seja agradável, intuitiva e que de fato auxilie o aluno na construção do conhecimento.

Dentre os projetos citados anteriormente merece destaque o projeto SIA-TP. Este sistema que surgiu inicialmente com (RIZZOTTO, 2005) e posteriormente com (VIGANO, 2008) busca através do uso de sistemas imunológicos artificiais diagnosticar algoritmos afim de auxiliar alunos iniciantes na disciplina de algoritmos, nesse período era conhecido apenas pelo nome de SIA. Mais tarde, com (POSSAMAI, 2008) esse mesmo sistema recebeu a visão da Teoria do Perigo, dessa forma, o diagnostico passou a detectar possíveis pontos de perigo. A partir desse momento passou a ser chamado de sistema SIA-TP. Esse sistema desenvolvido na plataforma *desktop* possibilita analisar os erros no algoritmo do aluno sinalizando as possíveis zonas de perigo. Cada zona corresponde a uma das áreas que a disciplina de algoritmo se divide. Uma situação de perigo ocorre toda vez que um aluno cometer uma quantidade pré estabelecida de erros em uma determinada área do conhecimento. Quando esta quantidade, que será tratada como limiar, for atingida, será identificado um sinal de perigo pelo sistema. A área de conhecimento de aprendizagem em que o aluno se encontra é que determina se o sinal de perigo vai ser forte ou fraco. Quando houver um sinal de perigo fora da área de conhecimento em que o aluno está inserido no momento da

avaliação, este sinal será fraco. Quando houver erros no algoritmo referentes à área de conhecimento atual do aluno, porém não atingindo o limiar, então será caracterizado um sinal fraco de perigo. Entretanto, se esta quantidade de erros for igual ou superior ao limiar, então o sinal de perigo será forte.

Assim sendo, para se dar continuidade aos projetos de pesquisa que encontram-se em desenvolvimento e por ser um foco com grande ênfase na interação humano - computador, este trabalho vem propor uma interface para um ambiente de aprendizagem que interaja com o aprendiz e que ofereça recursos e auxílios para que ele possa trilhar o seu caminho na descoberta do conhecimento tornando o ato de aprender programação uma tarefa mais agradável e prazerosa.

O presente trabalho busca desenvolver um ambiente web o qual utilize o projeto SIA-TP. Para que assim possa ser utilizado como ferramenta de apoio ao ensino da disciplina de algoritmos possibilitando ao professor uma melhor compreensão das dificuldades de cada aluno.

## 1.1 OBJETIVOS

Este trabalho tem como objetivo principal:

- Desenvolver um ambiente virtual de aprendizagem que dê suporte a aprendizagem da programação de computadores.

E, os seguintes objetivos específicos:

- Estudar conceitos da área de IA na Educação e ambientes virtuais de aprendizagem;
- Estudar os modelos de interface em ambientes de aprendizagem;
- Compreender a problemática da concepção de ambientes de aprendizagem de programação de computadores;
- Estudar o processo de aprendizagem baseado na resolução de problemas;
- Estudar os recursos de interface que beneficiam a aprendizagem;
- Levantar as dificuldades de cunho pedagógico existentes em ambientes virtuais de aprendizagem;
- Implementar componentes de interface fazendo uso das tecnologias Java e Flex que atendam as necessidades identificadas (editor de algoritmos, biblioteca de exercícios, ajuda, entre outros);

## **1.2 ORGANIZAÇÃO DO DOCUMENTO**

O capítulo 2 apresenta as características e a complexidade dos ambientes virtuais de aprendizagem a programação, as dificuldades na aprendizagem e um estudo de alguns ambiente virtuais disponíveis afim de se verificar os design das interfaces e os recursos de maior importância. Em seguida, no capítulo 3 apresenta-se o modelo de componentes de interface que atende as necessidade identificadas. O capítulo 4 apresenta um cenário contendo as funcionalidade apresentadas anteriormente. E, por fim o capítulo 5 apresentando os resultados de testes sobre ambiente construído.

As considerações finais apresentam o fechamento dessa etapa de pesquisa do projeto, revisitando o problema e reportando os resultados obtidos.

## **2 AMBIENTES DE APRENDIZAGEM**

A participação dos computadores nas diversas áreas do conhecimento já é uma realidade. Na Educação não é diferente, a utilização de novas tecnologias abre interessantes perspectivas para a educação. Uma bastante utilizada são as dos ambientes de ensino e aprendizagem que buscam despertar o interesse do aluno além de propiciar que o mesmo aprenda a aprender (SILVEIRA,2005).

Este capítulo busca apresentar a Inteligência Artificial e a sua inserção nos ambientes de virtuais de aprendizagem. Mostrar como se dá o processo de aprendizagem nesses ambientes e, as dificuldades de se ensinar a disciplina de programação de computadores. Analisar alguns ambientes existentes em instituições de ensino levantando aspectos de interface as quais se destacam perante a interação com o usuário. Desta forma, se terá uma melhor ambientação para a construção do ambiente proposto por este trabalho.

### **2.1 INTELIGÊNCIA ARTIFICIAL APLICADA À EDUCAÇÃO**

A área de Inteligência Artificial (IA) possui diversas definições. Nessa seção são apresentadas algumas delas:

Segundo RICH e KNIGHT (1994) a IA é a área da ciência da computação orientada ao entendimento, construção e validação de sistemas inteligentes, isto é, que exibem, de alguma forma, características associadas ao que chamamos inteligência.

Complementando essa idéia, CHARNIAK e MCDERMOTT (1985) definem que a IA é o estudo das faculdades mentais através do uso de modelos computacionais. E, MCCARTHY (1969) argumenta que uma máquina é inteligente se ela é capaz de solucionar uma classe de problemas que requerem inteligência para serem solucionados por seres humanos.

Basicamente, a IA fundamenta-se na idéia de que é possível modelar o funcionamento da mente humana através do computador. Esse tipo de pesquisa compreende uma série de sub-áreas tais como os Sistemas Especialistas, Robótica, Redes Neurais, Sistemas Multiagentes, Algoritmos Genéticos, entre outras.

Na educação, o computador tem sido utilizado como ferramenta de auxílio aos processos de ensino e aprendizagem. Por mais de 20 anos desenvolveram-se diversos sistemas afim de melhorar a aprendizagem e a qualidade de ensino. Os sistemas chamados “Treinamento baseado em

computador” (Computer Based Training, CBT) e os de Instrução/tutoria assistida ou apoiada por computador (Computer assisted instruction - CAI) foram os primeiros sistemas desenvolvidos e implantados afim de ensinar utilizando computadores. No entanto, eles não levavam em consideração as características individuais de cada aluno. Por esta razão, não ofereciam uma atenção individual ao aluno como a recebida na relação professor aluno.

A área de Educação envolve estudos de outras áreas como Psicologia, Ciências Cognitivas, Pedagogia. As demandas em termos de software na área da Educação são portanto para sistemas mais elaborados, que possam tomar decisões a partir dessas percepções. Por esta razão é que a IA passou a ser aplicada para o desenvolvimento de sistemas educacionais.

Com a aplicação da IA na educação surgiram os sistemas conhecidos como tutores inteligentes (Sistemas Tutores Inteligentes, STI) que apresentaram uma maior flexibilidade na apresentação do material educacional resultando num aumento considerável na performance e motivação dos alunos.

Para salientar as diferenças e pontos em comum entre os sistemas CAI e STI, apresenta-se o seguinte quadro comparativo:

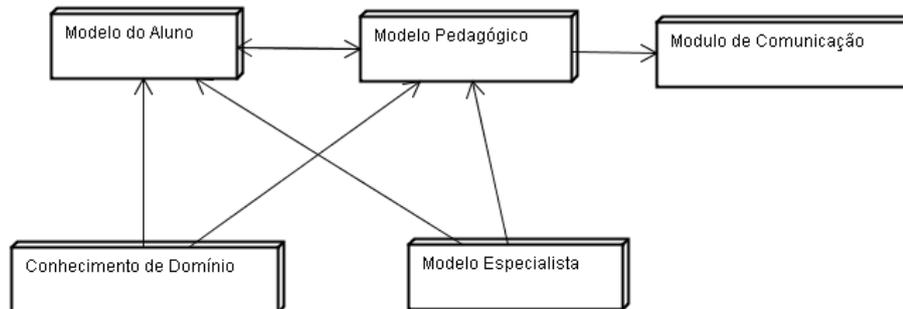
**Tabela 1: CAI x STI Fonte : GIRAFFA (1997, p. 19).**

<b>Aspecto</b>	<b>CAI</b>	<b>STI</b>
<b>Origem</b>	Educação	Ciência da Computação
<b>Bases Teóricas</b>	Skinner(Behaviorista)	Psicologia Cognitivista.
<b>Estruturação e Funções</b>	Uma única estrutura algorítmicamente pré-definida, onde o aluno não influi na seqüenciação.	Estrutura subdividida em módulos, cuja seqüenciação se dá em função das resposta do aluno.
<b>Estruturação do Conhecimento</b>	Algorítmica	Heurística
<b>Modelagem do Aluno</b>	Avaliam a última resposta	Tentam avaliar todas as respostas do aluno durante a interação
<b>Modalidades</b>	Tutorial, exercício e prática, simulação e jogos educativos.	Socrático, ambiente interativo diálogo bidirecional e guia.

Os STI evoluíram como o passar do tempo de acordo com o surgimento de novas tecnologias e a maior interação entre os pesquisadores da IA e os especialistas em Educação (CASAS, 1999). A contribuição dos STI no ensino fez com que novas teorias e estratégias instrucionais se desenvolvessem afim de aprimorar os STI ou ate mesmo melhorar a sua utilização

no meio acadêmico (COSTA,1997).

Segundo WOOLF (1992) um STI possui quatro grandes componentes que se relacionam da seguinte forma:



**Ilustração 1: Relação dos Componentes de um STI**

A Ilustração 1 destaca os seguintes componentes:

**Modelo do Aluno** : Armazena informações específicas de cada aprendiz, informações as quais servirão de base para o módulo pedagógico.

**Modelo Pedagógico** : Fornece um modelo de processo de ensino. Com base no modelo de estudante realiza decisões pedagógicas individuais.

**Conhecimento de Domínio** : Contém as informações a serem ensinadas ao aluno. Um dos desafios é como representar o conhecimento.

**Módulo de Comunicação** : Interações, layouts e os diálogos entre alunos ou entre professor e aluno são controlados por esse componente.

**Modelo Especialista** : Responsável por apontar onde estão localizadas as dificuldades do aluno.

As abordagens apresentadas nesta seção foram desenvolvidas a partir de estudos de equipes compostas por pesquisadores da área da Educação e IA. Para se desenvolver ambientes de aprendizagem é necessário compreender os processos de aprendizagem e como eles acontecerão através das interações com um software.

## **2.2 PROCESSO DE APRENDIZAGEM**

Segundo RINALDI (1997), a aprendizagem é um fenômeno extremamente complexo que envolve aspectos cognitivos, emocionais, orgânicos, psicossociais e culturais. É resultante do desenvolvimento de aptidões e de conhecimentos, bem como da transferência destes para novas situações. O processo de aprendizagem é desencadeado a partir da motivação. Esse processo se dá no interior do sujeito, estando, entretanto, intimamente ligado às relações de troca que o mesmo estabelece com o meio, principalmente, seus professores e colegas. Nas situações escolares, o interesse é indispensável para que o aluno tenha motivos de ação no sentido de apropriar-se do conhecimento.

Os ambientes de aprendizagem proporcionam um meio de aprendizagem diferente dos modelos tradicionais de ensino no qual o aluno se depara com o professor à sua frente explicando o conteúdo e dando suporte em caso de possíveis dúvidas. Nos espaços virtuais o aluno se torna responsável por buscar e formar o seu conhecimento. O professor passa a estar presente de uma forma indireta e isso faz com que o mesmo busque informação através cooperação entre os demais participantes e do conteúdo didático disponível. Os professores passam a ter a função de orientadores, articuladores, problematizadores, pesquisadores e especialistas na comunidade de aprendizagem. Isto implica em participar, instigar a discussão, acompanhar e analisar a construção do conhecimento através da participação individualizada nos espaços de interação.

Cada aluno possui um potencial perceptivo próprio, habilidades diferenciadas. SANTAROSA (2007) afirma que todos podem aprender mas não da mesma maneira é importante fornecer a possibilidade de cada aluno trabalhar o conteúdo conforme o seu ritmo de forma individualizada. E, cabe ao professor, auxiliado por recursos disponíveis no ambiente, perceber os pontos fortes e fracos do aluno e auxiliá-lo na evolução do conhecimento.

Uma das abordagens de ensino muito utilizada nos ambientes que buscam instigar o aluno na busca de conhecimento é a aprendizagem baseada na resolução de problemas.

## **2.3 RESOLUÇÃO DE PROBLEMAS E A PROGRAMAÇÃO DE COMPUTADORES**

A palavra “problema” muitas vezes é empregada de forma equivocada, referindo-se a uma série de exercícios que necessitam da aplicação de um procedimento já conhecido. Segundo CYRINO e TORALLES PERREIRA (2004), um problema nada mais é do que uma situação nova para o sujeito que vai resolvê-lo enquanto que um exercício é o uso de alguma habilidade ou

conhecimento já obtido pelo sujeito. Em outras palavras, o exercício envolve apenas a aplicação de procedimentos a um problema enquanto que resolver problemas envolve a criação de procedimentos para chegar a uma solução satisfatória.

Resolver problemas depende da experiência previa do indivíduo pois para se chegar a uma solução o indivíduo precisa buscar em seus esquemas mentais relacionados com situações análogas, já vivenciadas, para ter condições de criar a estratégia que o leve a solução do problema (FUKS,2000).

Programar é uma atividade que tem como fundamento o desenvolvimento de algoritmos, que é a parte criativa do processo de programação, ficando a execução (reprodução de passos) para ser realizada pelo computador. Construir algoritmos requer raciocínio lógico, reflexão, pesquisa e envolvimento por parte do aluno para com o problema a ser modelado. É também uma atividade de resolução de problemas que requer o domínio de uma linguagem de programação, o conhecimento do conteúdo que está sendo tratado, e criatividade (MALTEMPI, 2000).

Um algoritmo consiste em uma seqüência de instruções bem definidas que se seguidas seqüencialmente chegam a um resultado o qual pode ou não ser a solução para o problema. A atividade de ensino de algoritmos possui desafios bastante específicos. O maior deles é fazer com que o aluno consiga compreender a resposta que está construindo tendo em mente os resultados esperados (SAJANIEMI e KUITTINEN, 2005; KOLIKANT e BEN-ARI, 2008).

O estudo de algoritmos abrange diversas competências tanto abstratas quanto concretas (NARDI, 1993). Na aprendizagem de algoritmos, o aprendiz precisa compreender o que a máquina é capaz de conceituar e a lógica por traz da linguagem de programação (KOMIS, 2001). Além disso, ele precisa desenvolver a habilidade de transformar a solução abstrata do problema em solução concreta compreendida pela máquina, respeitando suas restrições e limitações.

A construção de um algoritmo requer do aluno a habilidade de identificar aspectos essenciais e estruturar uma solução adequada ao problema. Para isso o aluno pode dispor de diversas notações para auxiliar na representação tais como o fluxograma e o pseudocódigo. O fluxograma é a representação esquemática de um algoritmo ou processo, geralmente desenvolvidos de forma gráfica com intuito de ilustrar. O pseudocódigo consiste numa linguagem simplificada e informal com baixa complexidade, se comparada a linguagens de programação propriamente ditas, e que se utiliza de palavras nativas da linguagem do indivíduo (BRUSILOVSKY et al., 1997).

Conforme estudo realizado por SIOZOU, TSELIOS, e KOMIS (2008) que consistia em avaliar o efeito de diferentes representações no ensino de noções básicas de algoritmos, os autores concluíram que não existe diferenças significativas no aprendizado dos alunos que são ensinados via a metodologia fluxograma em relação aqueles ensinados através do pseudocódigo. Entretanto,

vale ressaltar que ambas as técnicas se complementam e que se utilizadas em conjunto contribuem ainda mais no aprendizado.

A habilidade de programação só pode ser desenvolvida através da prática. Porém, ao tentar escrever os primeiros programas os alunos iniciantes passam por uma série de dificuldades como utilizar o editor de texto, instalar a linguagem e o compilador, aprender a interpretar as mensagens de erro geradas pelo compilador, entre outras (TRUONG, 2005).

A falta de sucesso principalmente no estágio inicial de aprendizado levam a queda da auto-estima do aluno e dá a ele, uma má impressão sobre programação. Sem contar que em muitos casos, existe o número elevado de alunos numa turma e recursos limitados, fator que dificulta o aprendizado individual e o auxílio por parte do professor.

Uma vez que há sempre inúmeras maneiras de se resolver um problema por meio da programação diferentes soluções podem ser obtidas. Essa característica torna a atividade de programação especialmente interessante para ser empregada em um ambiente de ensino-aprendizagem (MALTEMPI, 2000).

## 2.4 EXEMPLOS DE AMBIENTES DE APRENDIZAGEM

Nessa sessão serão apresentados alguns ambientes virtuais de aprendizagem que foram desenvolvidos para auxiliar na formação de nível universitário. Também foram analisados ambientes utilizados na educação fundamental e média. Nesse estudo buscou-se levantar as funcionalidades existentes em cada um deles e os recursos de interface utilizados. E, com base nisso extrair as funcionalidades a serem desenvolvidas e modelar a interface para o ambiente proposto nesse trabalho.

Os ambientes analisados foram agrupados da seguinte forma: Ambiente da Matemática, Ambientes Infante-Juvenil, Ambientes de Gerência de Cursos, Ambientes em Programação de Computadores e Ambientes Médicos. Optou-se por ambientes de diversas áreas devido a variedade de funcionalidades e de recursos de interface.

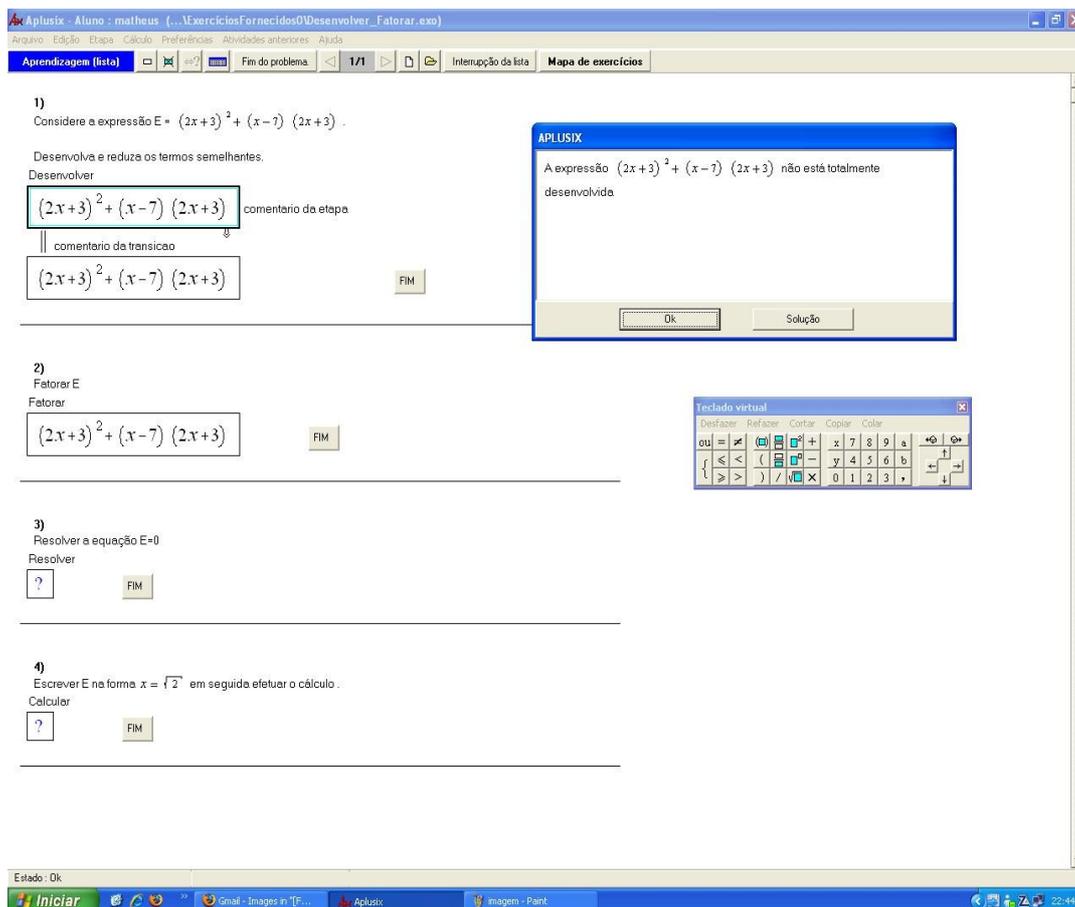
### 2.4.1 AMBIENTE DA MATEMÁTICA:

- **Aplusix**

O Aplusix consiste num software *desktop* livre direcionado a aprendizagem da álgebra. Desenvolvido por pesquisadores da DidaTIC, do laboratório Leibniz localizado em Grenoble,

França (APLUSIX, 2009). O sistema permite que o aluno resolva exercícios matemáticos, permite retroação assim o aluno pode corrigir possíveis erros. Possui um editor intuitivo semelhante a um ambiente papel e lápis, possibilitando o aluno desenvolver os cálculos passo a passo.

Dentre os itens de interface que o compõe destacam-se o editor, as ações usuais como seleção, copiar, colar, arrastar-soltar, comandos prontos, personalização de recursos configuráveis pelo professor, estatísticas de erros.



**Ilustração 2: Interface do sistema Aplusix**

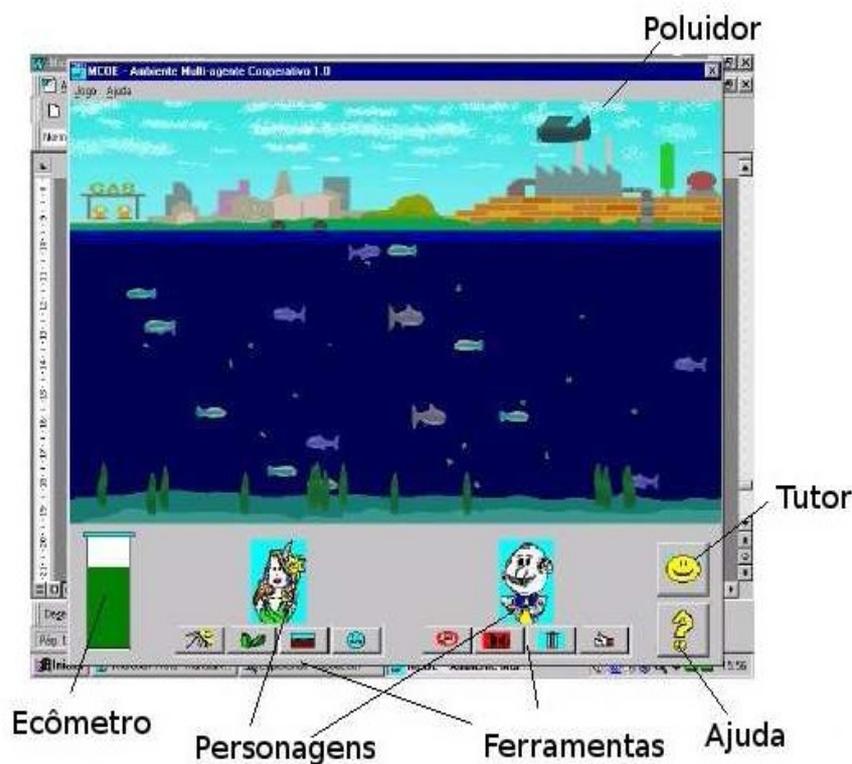
## 2.4.2 AMBIENTE INFANTO-JUVENIL

- **AS-MCOE**

O Projeto AS-MCOE (Tutor inteligente modelado em AgentSpeak(L) ) desenvolvido por Alexandre de Oliveira Zamberlam , Claudia Perez e Rodrigo Rafael Villarreal Goulart consiste num jogo educacional com a ecologia sendo a temática principal. O aluno comanda um personagem

que interage com uma cadeia alimentar de um lago. O objetivo consiste em manter o equilíbrio da cadeia, as dificuldades relacionadas ao conteúdo são avaliadas pelos agentes tutores que enviam dicas e explicações ao aluno. O aluno aprende com seus erros e passa a buscar informações relacionadas a determinados tipos de poluição atribuídas ao ambiente para que assim possa compreender quais as medidas necessárias para que se possa equilibrar o ambiente (ZAMBERLAM e GOULART, 2006) .

A interface se destaca por possuir um ambiente agradável ao usuário e animado. Botões simples e ferramentas auto-explicativas.



**Ilustração 3: Interface do sistema AS-MCOE**

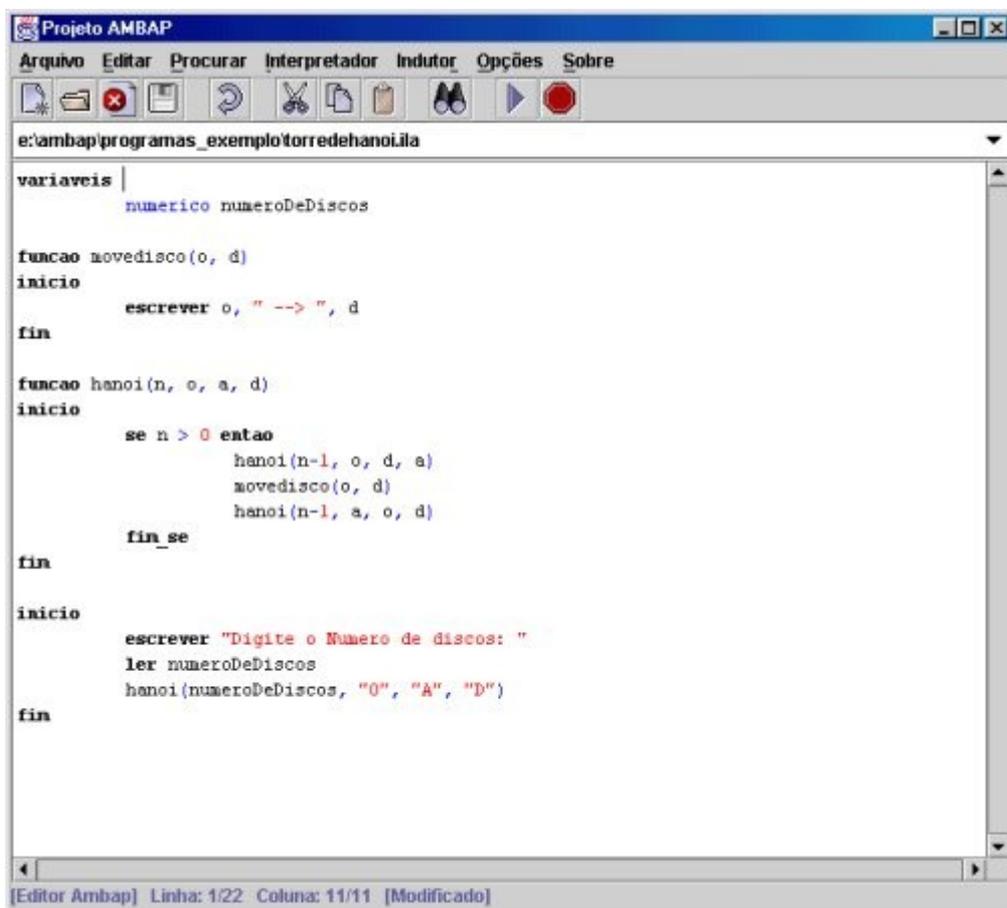
### 2.4.3 AMBIENTES EM PROGRAMAÇÃO DE COMPUTADORES

- **AMBAP**

O AMBAP é um ambiente desktop para o aprendizado de programação desenvolvido pela

UFAL – Universidade Federal de Alagoas. Sendo, André Atanasio Maranhão Almeida, Klebson dos Santos Silva e Rodrigo de Barros Paes os principais desenvolvedores desse projeto. A linguagem utilizada é o ILA (Interpretador de Linguagem Algorítmica) . Permite facilitar o aprendizado da lógica de programação através de um processo de simulação (AMBAP, 2009).

A interface é composta por uma barra de menu, uma barra de ferramentas, um editor. O editor se utiliza de cores para facilitar a identificação das palavras reservadas referentes a ILA. Possibilita salvar o algoritmo localmente e carregá-lo quando desejado. Possui recursos de recortar, colar, copiar, localizar , substituir e desfazer partes de texto. Permite a execução normal do algoritmo e uma execução passo-a-passo para fins de depuração. Um recurso de ajuda que permite ao aluno sanar dúvidas referentes à linguagem. Permite trabalhar com diversos programas ao mesmo tempo.



```
Projeto AMBAP
Arquivo Editar Procurar Interpretador Indutor Opções Sobre
e:\ambap\programas_exemplo\torredehanoi.ila
variaveis |
    numerico numeroDeDiscos
funcao movedisco(o, d)
inicio
    escrever o, " --> ", d
fin
funcao hanoi(n, o, a, d)
inicio
    se n > 0 entao
        hanoi(n-1, o, d, a)
        movedisco(o, d)
        hanoi(n-1, a, o, d)
    fin_se
fin
inicio
    escrever "Digite o Numero de discos: "
    ler numeroDeDiscos
    hanoi(numeroDeDiscos, "O", "A", "D")
fin
[Editor Ambap] Linha: 1/22 Coluna: 11/11 [Modificado]
```

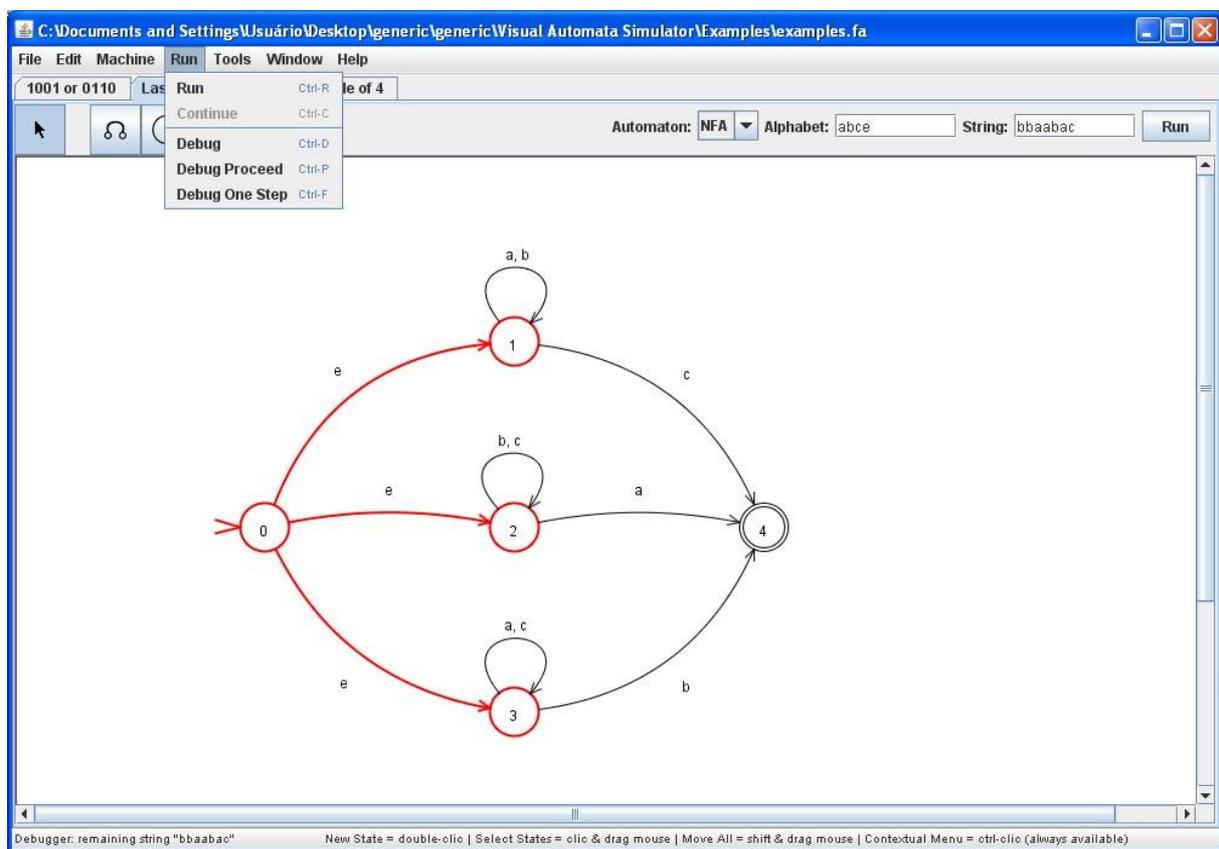
**Ilustração 4: Interface do Projeto AMBAP**

- **Visual Automata Simulator**

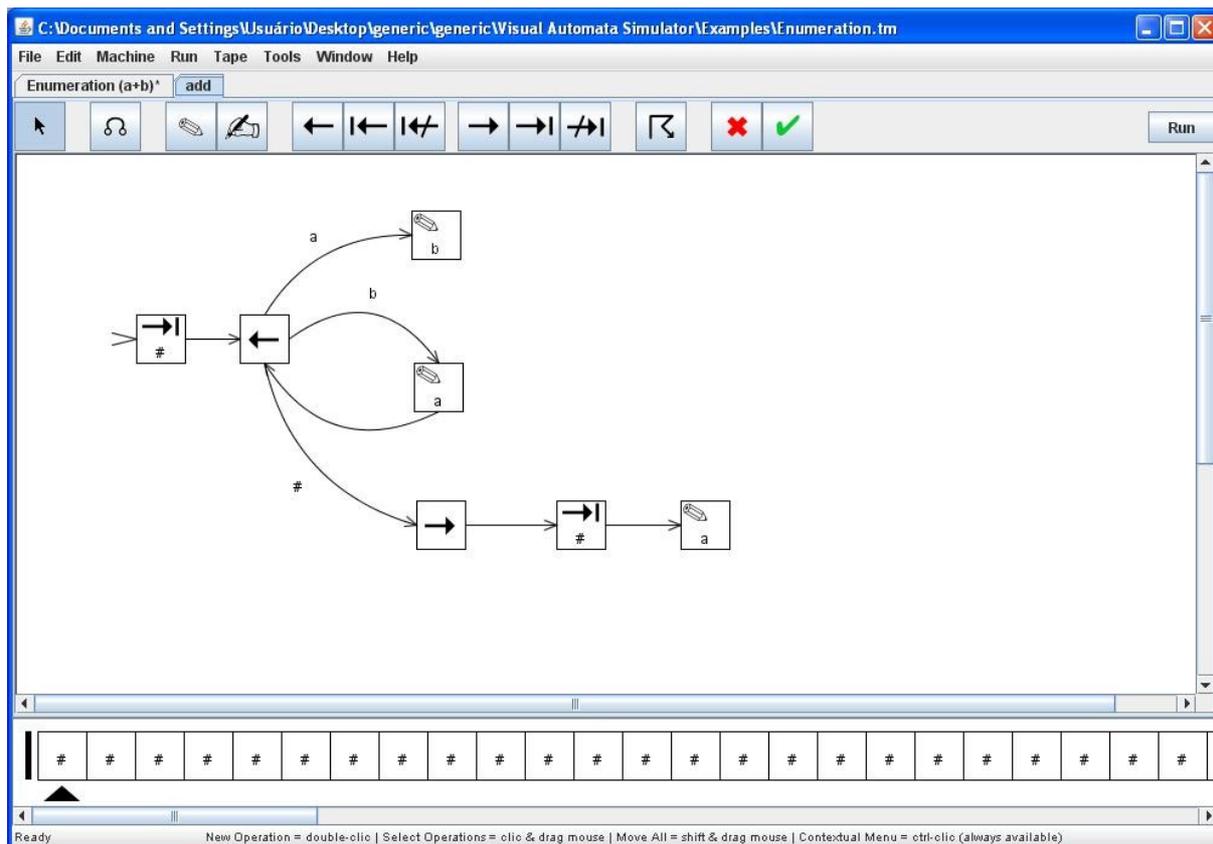
Ambiente *desktop*, desenvolvido por Jean Bovet na Universidade de San Francisco em

2004, que permite criar e simular autômatos finitos determinísticos e não determinísticos e, máquinas de *turing*. No momento de criação de um documento o usuário especifica se estará lidando com autômatos ou máquina de *turing*. Dependendo da opção escolhida a interface se adapta adicionando os componentes específicos para tratar a opção escolhida. Possui um modo de *debug* onde é possível depurar o programa passo a passo (BOVET, 2009).

A interface permite abrir múltiplos documentos. Todos os itens na área central podem ser movidos utilizando o mouse. Existem teclas de atalho de teclado e mouse afim de agilizar o manuseio. É composta por uma barra de menu, uma barra de abas, uma barra de ferramentas e uma barra de status.



**Ilustração 5: Interface do *Visual Automata Simulator* no modo de tratamento de Autômatos**

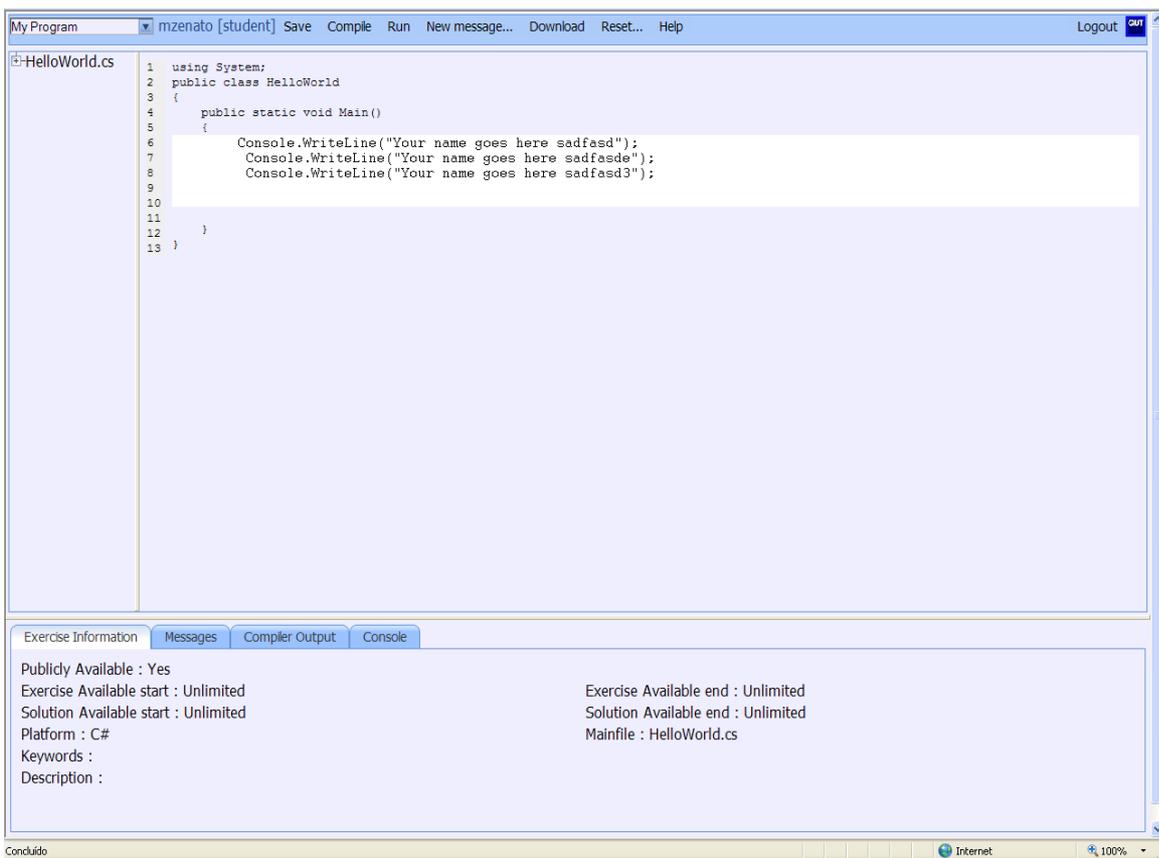


**Ilustração 6: Interface do *Visual Automata Simulator* no modo de tratamento de Máquina de Turing**

- **Environment for Learning to Program (ELP)**

ELP é um ambiente interativo baseado na *web* para o ensino de programação de computadores aos alunos de Tecnologia da Informação da Universidade de Tecnologia de Queensland desenvolvido por Paul Roe em parceria com Peter Bancroft, John Hynd entre outros. O Sistema é composto por diversos exercícios, cada exercício é composto por um programa parcial para resolver um determinado problema, o aluno por sua vez deve terminar a construção desse algoritmo a fim de resolver o exercício proposto. O sistema compila o programa do aluno devolvendo o resultado e, caso não consiga retorna mensagens referentes aos erros encontrados. O aluno pode deixar comentários no exercícios os quais são lidos pelo professor e dessa forma fornecer um feedback ao aluno (ENVIRONMENT FOR LEARNING TO PROGRAM, 2009).

A interface é composta por uma barra de menu, uma área de navegação, um editor e uma barra de status na qual o aluno recebe os feedbacks e posta as suas duvidas.



**Ilustração 7: Interface do sistema ELP**

#### 2.4.4 AMBIENTE MÉDICO

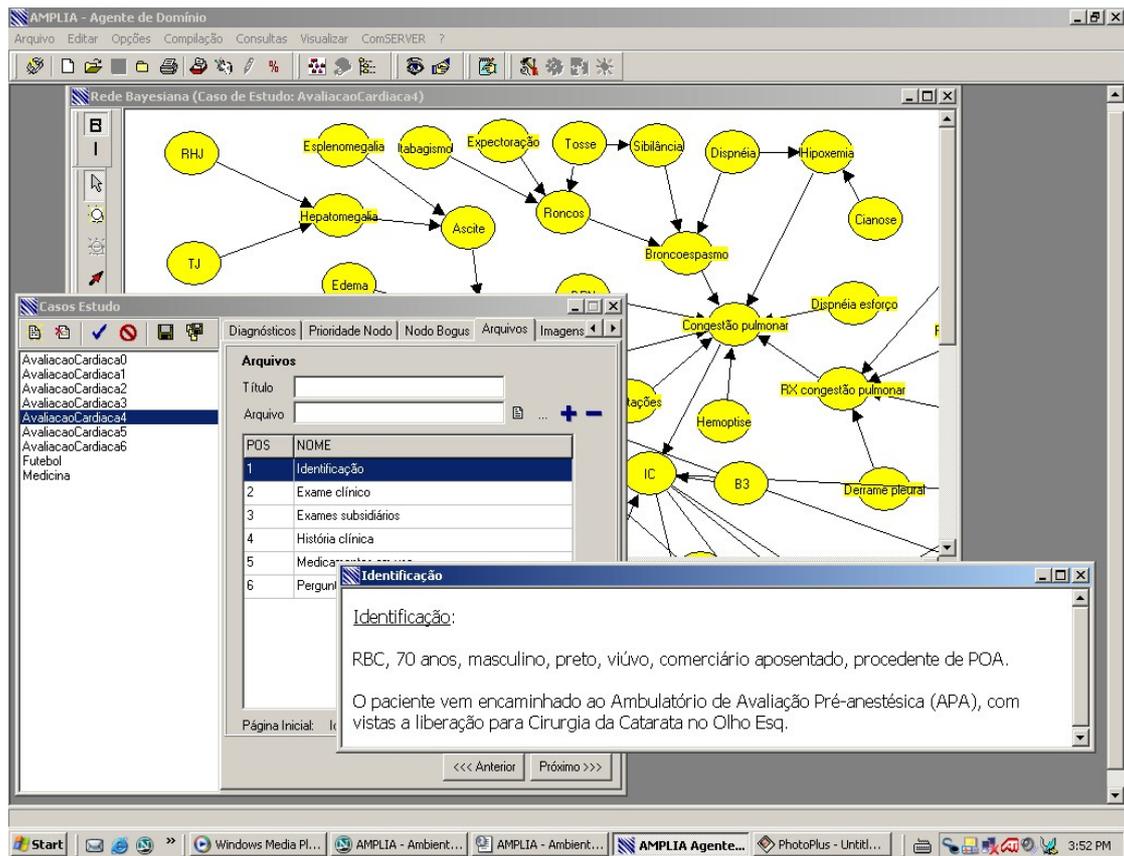
- **AMPLIA**

O AMPLIA (Ambiente Multiagente Probabilístico Inteligente de Aprendizagem) é produto de inúmeros projetos de pesquisa desenvolvidos pelo Grupo de Inteligência Artificial do Instituto de Informática da Universidade Federal do Rio Grande do Sul (II/UFRGS), em parceria com a Faculdade de Medicina da Universidade Federal do Rio Grande do Sul (FAMED/UFRGS), o Departamento de Ciência da Computação da Universidade de Brasília (CIC/UnB) e o Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa (DI/UL) (AMPLIA,2009).

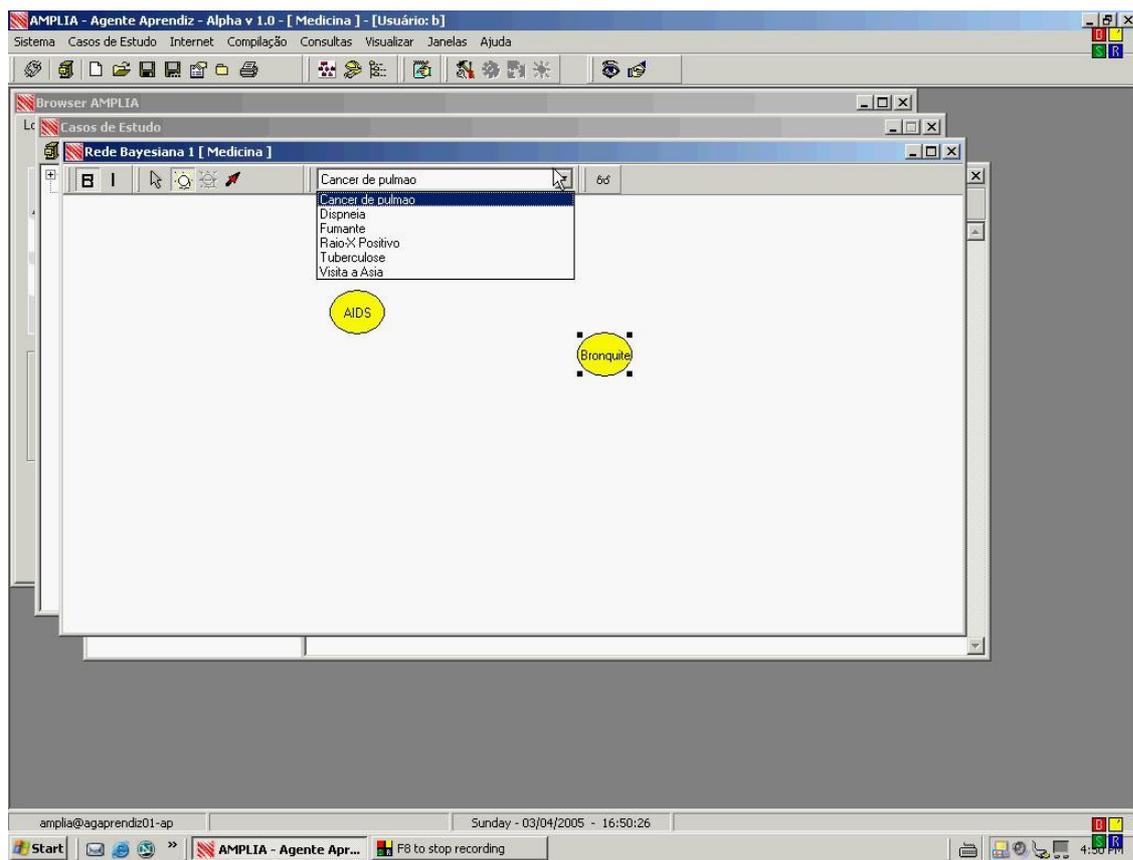
O sistema foi desenvolvido com o objetivo de auxiliar na educação médica, no apoio ao desenvolvimento do raciocínio diagnóstico e na modelagem das hipóteses diagnósticas. É constituído por um sistema multiagente e utiliza redes bayesianas, que têm sido amplamente utilizadas para modelar domínios incertos, como é o caso na medicina.

A proposta do AMPLIA, é oferecer um ambiente aberto para que um aluno construa um modelo gráfico da representação de sua hipótese diagnóstica para um dado caso clínico, utilizando redes bayesianas. A rede do aluno é comparada à rede de um especialista do domínio, que se encontra armazenada no ambiente, e as diferenças entre elas são tratadas por um agente inteligente que utiliza estratégias pedagógicas baseadas na teoria construtivista.

A interface possui uma barra de menu, uma barra de ferramentas e uma estrutura de janelas internas. Nas mensagens de erro ao usuário existem itens de ajuda referentes ao erro provocado com explicação e exemplo de como se deve proceder. A interface permite movimentar os nodos da rede livremente. Uma característica de destaque é que na construção da rede probabilística o usuário seleciona um elemento por vez, o utiliza, e após deve selecionar outro elemento na barra de ferramentas para poder mover ou adicionar outro item. Característica semelhante a dos softwares de edição de imagem existentes no mercado.



**Ilustração 8: Interface do AMPLIA mostrando informações para a construção das redes**



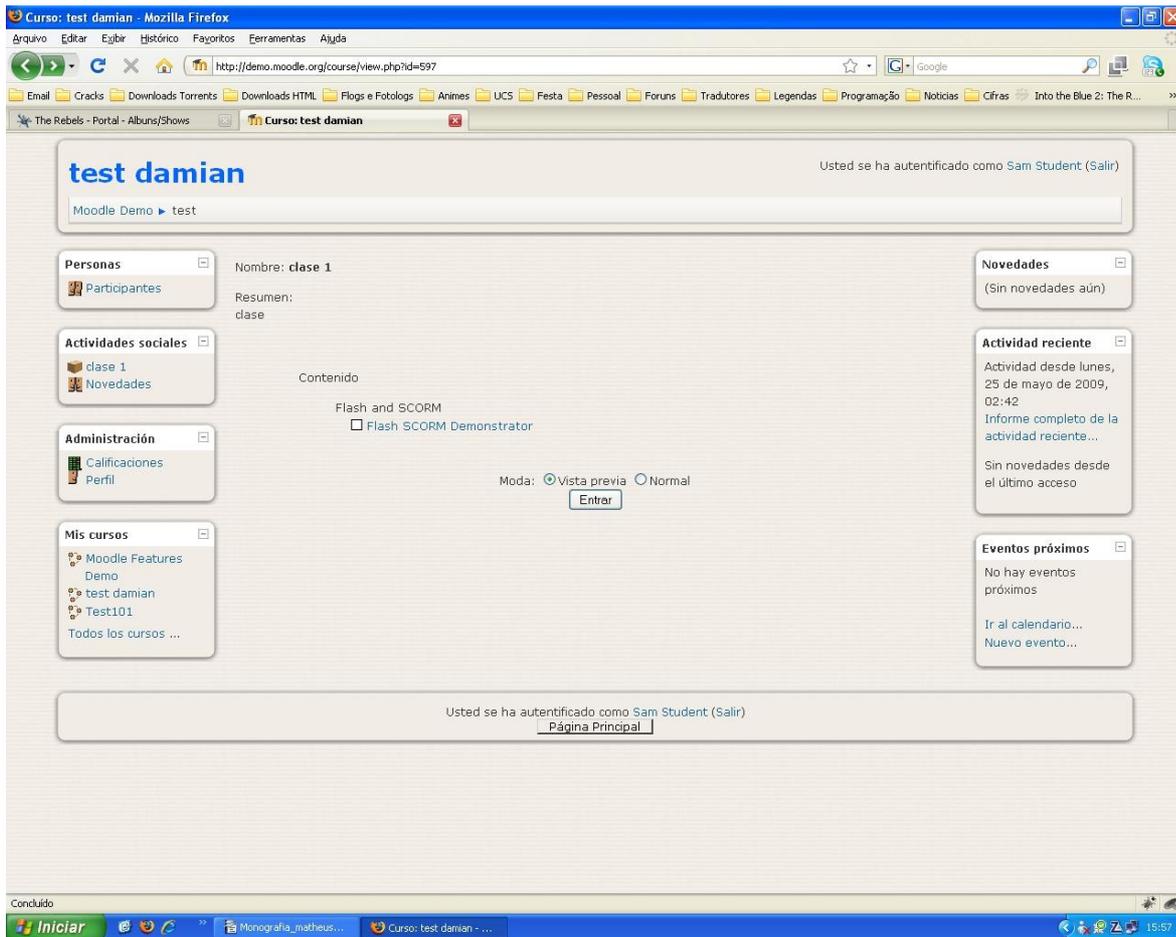
**Ilustração 9: Interface do AMPLIA no modo de construção da rede**

#### 2.4.5 AMBIENTES DE GERÊNCIA DE CURSOS:

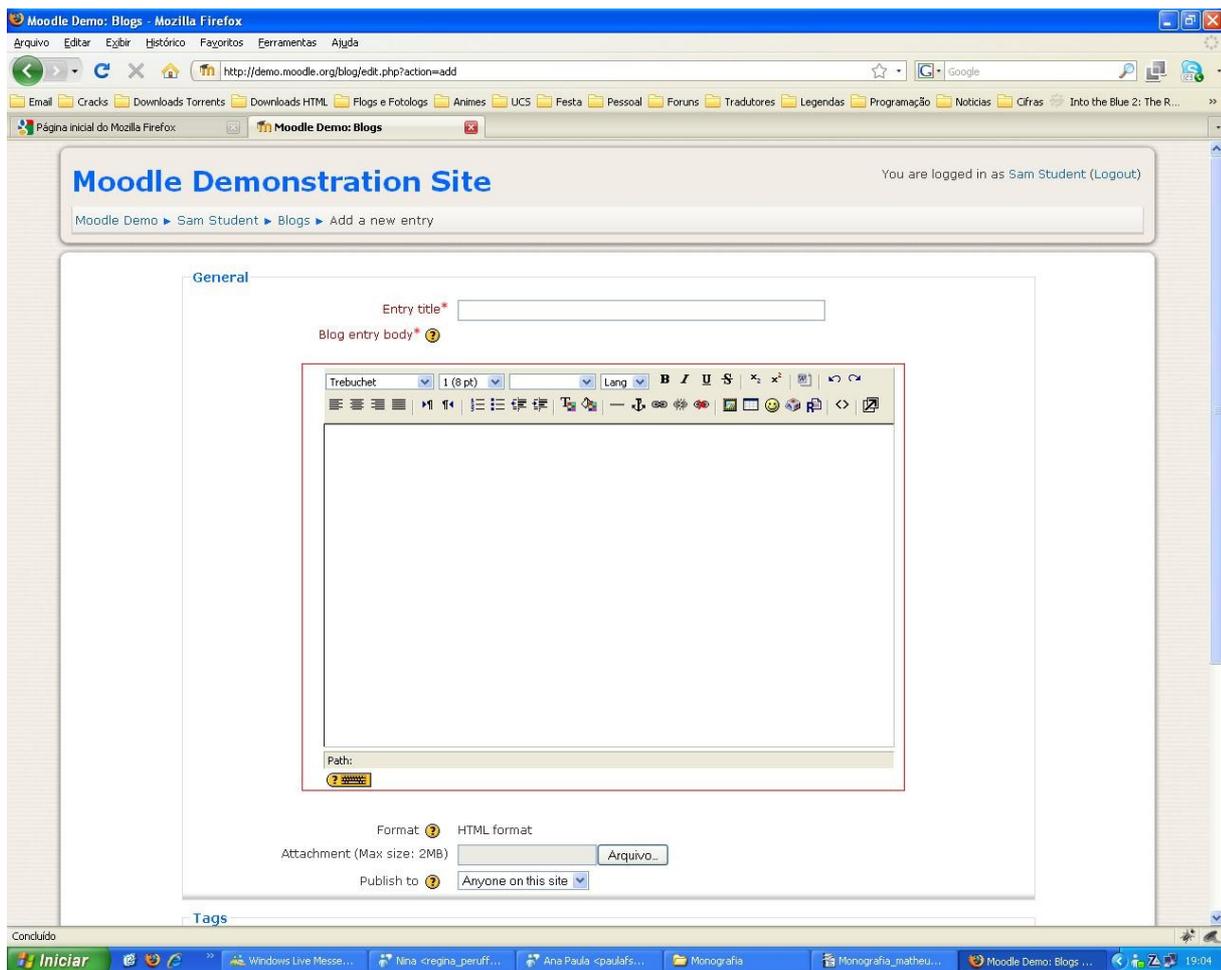
- **Moodle**

Moodle é um software livre, de apoio à aprendizagem. Criado em 2001 pelo educador e cientista computacional Martin Dougiamas. Constitui-se em um sistema de administração de atividades educacionais destinado à criação de comunidades on-line em ambientes virtuais voltados para a aprendizagem colaborativa. Permite, de maneira simplificada, a um estudante ou a um professor integrar-se, estudando ou lecionando, num curso on-line à sua escolha. Alguns dos recursos disponíveis são: Materiais, Avaliação do Curso, *Chat*, Diálogo, Diário, Fórum, Glossário, Lição, Pesquisa de Opinião, Questionário, SCORM, Tarefa, Trabalho com Revisão, *Wiki* (MOODLE, 2009).

A interface do ambiente é composta basicamente por uma barra de navegação, conjuntos de funcionalidades nas laterais, uma área principal ao qual exhibe o conteúdo focal e uma barra de status. Faz uso de um editor de texto on-line bem semelhante aos editores de texto *desktop*.



**Ilustração 10: Interface do Ambiente Moodle**



**Ilustração 11: Interface do Ambiente Moodle - Editor de texto**

- **Solar**

O Solar é um ambiente virtual de aprendizagem desenvolvido pelo Instituto UFC Virtual, da Universidade Federal do Ceará. Projeto de pesquisa universitária, coordenado pelo Prof. Dr. Mauro Cavalcante Pequeno, tendo como equipe inicial: Prof. Msc. Henrique Pequeno, Patrícia de Sousa Paula, Prof. Msc. Wellington W. F. Sarmiento, Prof. Msc. Robson Loureiro e Profa. Msc. Cátia Oliveira (SOLAR, 2009).

O sistema foi desenvolvido baseado na abordagem educacional construtivista, com auxílio a estruturação de conteúdo. Agrega um grande número de ferramentas e recursos flexíveis e configuráveis.

Possui recursos como: Agenda, Perfil dos participantes, Chat, Fórum, Correio eletrônico, Material de apoio, Portfólio, Estatísticas de acesso, Funções administrativas e de configuração

A interface desse ambiente é composta basicamente de um menu principal localizado ao lado esquerdo, e toda a área central fica destinada ao conteúdo. Possui uma barra de navegação para que o aluno possa se situar no ambiente. O aluno pode visualizar o conteúdo das aulas de forma

seqüencial através de um recurso de paginação.

Aluno: **Isabela** :: Turma: **Turma A** **SOLAR**

Aluno > Tutoria em EAD > Aulas

Educação a Distância		
Aula	Descrição	Tema
Ambientação	Neste espaço	Tutoria em EAD ✓
Ambientação	Neste espaço	Tutoria em EAD ✓
Educação a Distância	Educação a Distância	Tutoria em EAD ✓
Autonomia em EAD	Autonomia em EAD	Tutoria em EAD ✓
Autonomia em EAD	Autonomia em EAD, continuação.	Tutoria em EAD ✓
Perfil de competências do professor tutor	Perfil de competências do professor tutor	Tutoria em EAD ✓
Perfil do professor tutor	Perfil do professor tutor	Tutoria em EAD ✓
Desenvolvimento da ação tutorial	Desenvolvimento da ação tutorial	Tutoria em EAD ✓
Modalidades de tutoria	Modalidades de tutoria	Tutoria em EAD ✓
Tutoria a Distância	Tutoria a Distância	Tutoria em EAD ✓
Prática tutorial	Prática tutorial	Tutoria em EAD ✓
Programação da ação tutorial	Programação da ação tutorial	Tutoria em EAD ✓
Estrutura da rede tutorial	Estrutura da rede tutorial	Tutoria em EAD ✓

✓ Indica que a aula já foi visualizada.  
Para visualizar uma aula, clique sobre o link no nome da mesma. Se não houver link, a aula não teve nenhum arquivo indicado para abertura (Informe o responsável).

**Ilustração 12: Interface do ambiente SOLAR**

Curso > Módulo > Aula Aluno: **claudya**

Tutoria em EAD > Educação a Distância > Educação a Distância

1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 : 9 : 10 > Aula 1 de 12

## Formação de Tutores em EAD

### Aula 2 - Educação a Distância

A concepção da Educação a Distância em qualquer parte do mundo e, por isso, no Brasil é inevitavelmente influenciada por acontecimentos globais, como por exemplo na Europa e nos Estados Unidos no que se refere a sua aplicabilidade em grandes grupos. Essas influências e propostas pedagógicas vão trazer para a EAD uma perspectiva individualista, modular e descontextualizada, cuja proposta não estimula a autonomia que é uma das características necessárias para a prática da Educação a Distância.

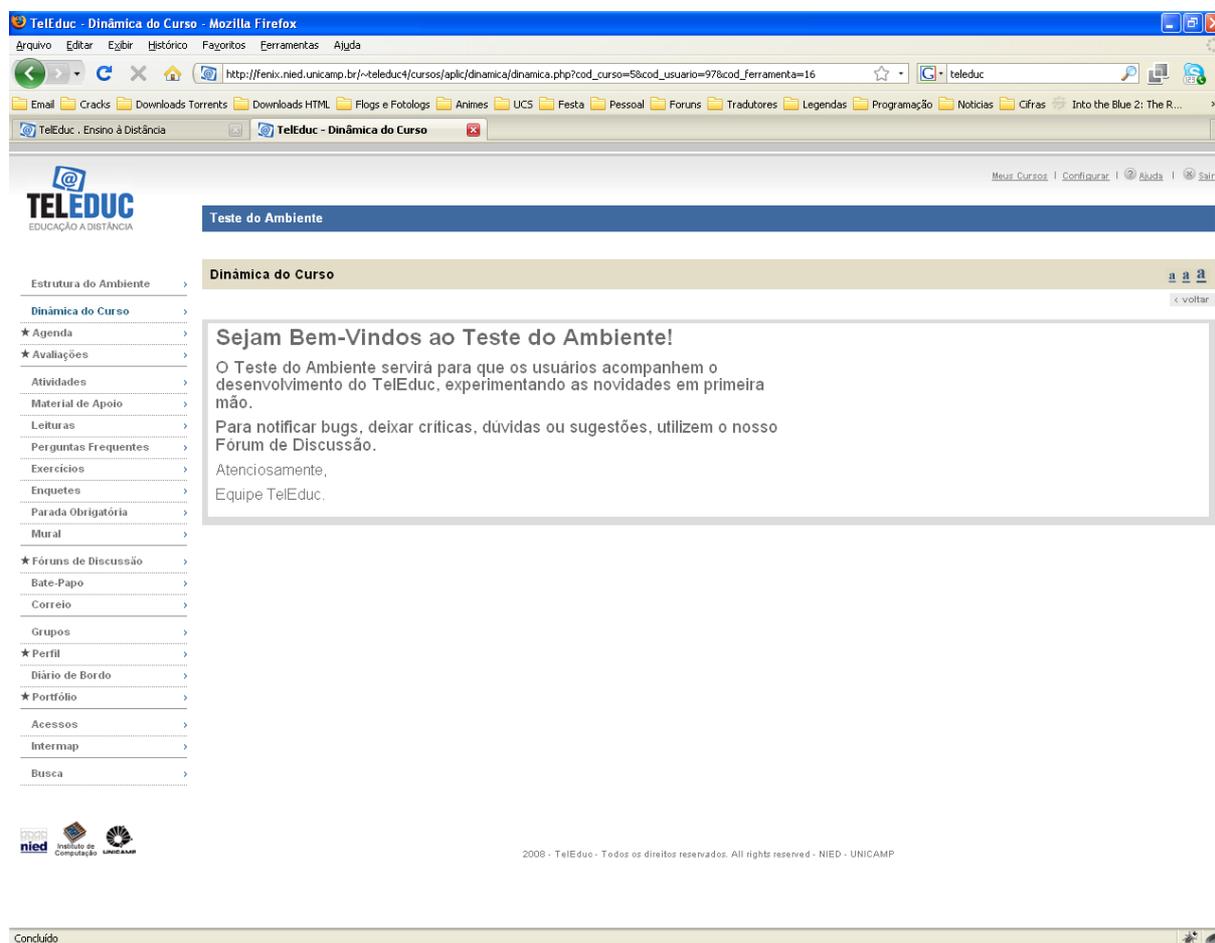
Por outro lado, o fenômeno da globalização, a necessária qualificação crescente dos trabalhadores impõem a formação constante e a busca de geração de conhecimentos, habilidades e relações que possibilite um crescimento intelectual e um ganho de conhecimento.

**Ilustração 13: Interface do Ambiente SOLAR - Demonstração de Conteúdo de aula**

- **TelEduc**

O TelEduc é um ambiente desenvolvido por pesquisadores do Núcleo de Informática Aplicada à Educação da Unicamp. Direcionado ao gerenciamento de cursos na Web. Desenvolvido a partir das necessidades levantadas junto a usuários. Contempla diversas funcionalidades como: Correio Eletrônico, Grupos de Discussão, Mural, Portfólio, Diário de Bordo, Bate-Papo, Material de Apoio, Leituras.

Sua interface é simples basicamente composta por uma barra de menu lateral, e uma área central que apresenta o conteúdo correspondente ao item de menu selecionado. Possui recursos para controle de tamanho de fonte.



**Ilustração 14: Interface do Ambiente TelEduc**

TELEEDUC - FÓRUMS DE DISCUSSÃO - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://fenix.nied.unicamp.br/~teleduc4/cursos/aplic/forum/ver\_forum.php?cod\_forum=3&cod\_curso=5&status=A

TELEEDUC - Ensino à Distância

Meu Curso | Configurar | Ajuda | Sair

**TELEEDUC**  
EDUCAÇÃO A DISTÂNCIA

Teste do Ambiente

Fóruns de Discussão - Ver fórum - Zona de Testes

Retornar à lista de fóruns Compor nova mensagem

Mensagens (1 a 5 de 5) Ordenar por: **Árvore**

#	Título	Autor	Relevância	Data
1.	Respondendo	Márcia Cristina Pereira Fortuna	Relevância Não Avaliada	24/03/2009
2.	Eco	Sandra Fonseca	Relevância Não Avaliada	19/03/2009
3.	Re: Eco	Moema Gomes Moraes	Relevância Não Avaliada	20/03/2009
4.	Re: Eco	Helio Ferrari	Relevância Não Avaliada	23/03/2009
5.	Re: Eco	Denise Faria	Relevância Não Avaliada	06/04/2009

Exibir todos

Concluído

2008 - Teleduc - Todos os direitos reservados. All rights reserved - NIED - UNICAMP

**Ilustração 15: Interface do Ambiente Teleduc - Fórum**

- **Amadeus**

O Amadeus é um ambiente de administração de atividades educacionais destinado a aprendizagem colaborativa. Desenvolvido pelo grupo de pesquisa em tecnologia educacional CCET, do Centro de Informática da UFPE. O ambiente conta com diversos recursos como: Materiais de Estudo, Avaliação, Chat, Fórum, Gestão de Conteúdo, Glossários entre outros.

Se destaca pela a utilização de recursos como webcasting, vídeo sob demanda, jogos multiusuários, acessibilidade por dispositivos móveis como celular.

Sua interface é composta por uma barra de menus na parte esquerda, e uma área central referente ao conteúdo. Na parte superior possui um recurso de navegação que permite que o usuário se localize.



**Ilustração 16: Interface do Ambiente Amadeus - Pagina Principal**

Olá Professor.  
[Sair](#)

projetoAmadeus  Alpha

O Projeto Casos de Sucesso Grupo CCTE Fale Conosco Licença

**Gestão de Conteúdo**

[Página Inicial](#) [Design da Interação](#) Módulos

[Dados do Curso](#)  
[Listar Participantes](#)  
[Editar Dados do Curso](#)

<b>Design da Interação</b>		
Professores: Professor.		
Monitores: Alex Sandro Gomes.		
Data de Início: 04/05/2007		
Data de Término: 04/08/2007		

<b>Conceito</b>	<b>Visível</b>	<b>1</b>
Descrição do Módulo: <i>Definir o conceito de interação</i>		
<b>Materiais</b>	<b>Atividades</b>	
<ul style="list-style-type: none"> <li>• <a href="#">Texto</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Importância do design de interação</a> <a href="#">Ver Resultados</a></li> </ul>	
<a href="#">Editar</a>		

<b>Nome do Módulo</b>	<b>Visível</b>	<b>2</b>
Descrição do Módulo: <i>Lorem ipsum sit amet abajour rolis strange text just for fun girls have more than this</i>		
<b>Materiais</b>	<b>Atividades</b>	
	<ul style="list-style-type: none"> <li>• <a href="#">primeira avaliação</a></li> </ul>	
<a href="#">Editar</a>		

[Criar novo modulo](#)

**Ilustração 17: Interface do Ambiente Amadeus - Gestão de Conteúdo**

## 2.4.6 CONCLUSÕES SOBRE OS AMBIENTES ANALISADOS

O estudo realizado sobre os ambientes anteriormente citados não teve o intuito de compará-los mas sim levantar aspectos de interfaces e funcionalidades comuns. De uma forma geral, constatou-se que as interfaces tendem a optar pela simplicidade.

Percebeu-se que a cor é um aspecto muito importante pois é através dela que o ambiente transmite ao usuário um ar de seriedade e, quando necessário pode ser utilizada para chamar

atenção do usuário.

Na maioria dos casos os menus contendo o acesso as principais funcionalidade ficam posicionados na parte superior da tela ou na parte esquerda. A utilização de imagens com metáfora apropriada também é empregada na tentativa de remover os rótulos dos botões e conseguir um aspecto visual mais agradável. O uso de abas para separar conteúdos também é utilizado tanto em aplicações web quando na desktop. Utilizar padrões para botões ou links é uma boa prática visto que o usuário consegue se situar mais facilmente no ambiente sem ter que precisar procurar o acesso as funcionalidades em meio ao conteúdo disponível.

Constatou-se também que os ambientes de gerência de cursos possuem funcionalidades bem semelhantes enquanto que os ambientes especialistas ficam restritos a funcionalidades relacionadas à área a qual atua. Em nenhum dos casos encontrou-se um ambiente que possua as funcionalidades de um ambiente de gerência de cursos agregadas a funcionalidades especialistas e que possam ser utilizadas paralelamente.

## 2.5 DIFICULDADES NOS AMBIENTES

Os ambientes virtuais de aprendizagem são ferramentas muito úteis para o auxílio à aprendizagem e para a educação a distância. Porém, para que eles se tornem mais eficazes é fundamental que alunos e professores avaliem o seu processo de ensino-aprendizagem. O professor por sua vez deve desenvolver atividades junto ao ambiente que permitam desenvolver no aluno habilidades como a de auto-estudo, interpretação de textos e reflexão. O aluno por sua vez deve ser cada vez mais participante da sua construção de conhecimento.

Segundo um estudo realizado pelo Laboratório de Ambientes Virtuais de Aprendizagem (Lavia) da Universidade de Caxias do Sul, que consistia em entender como ocorre a aprendizagem nos ambientes identificando dificuldades e obstáculos referentes ao uso das ferramentas, as concepções prévias dos alunos e outras variáveis que interferem no processo de aprendizagem, se faz necessário desenvolver, previamente, atividades que permitam o desenvolvimento de habilidades de auto-estudo, interpretação de texto e reflexão para os alunos. Com base nesse estudo, os aspectos pedagógicos relacionados aos ambientes e que merecem destaque são:

- Alunos que não lêem e nem pensam no que estão fazendo: muitas vezes simplesmente sugerindo que relacionem o que lêem com o problema a ser resolvido, a dificuldade é sanada;
- Concepção de aprendizagem do aluno: em geral acredita que aprende assistindo ao que o professor faz;
- Dificuldades dos alunos em expressar, por escrito, dúvidas ou problemas encontrados, elaborando de forma organizada, suas percepções, dificuldades e concepções;
- Falta de habilidade dos alunos para lerem e interpretarem os hipertextos disponíveis nos ambientes, e a partir daí deduzirem maneiras de resolverem problemas;
- Dificuldades dos alunos em relacionar o que está sendo estudado com outros conceitos já estudados (ancoragem);
- Dificuldades dos alunos em avaliar se um resultado obtido pelo processamento de um software está adequado ou correto;

- Dificuldades dos alunos em interagir com os colegas, examinando e analisando os resultados obtidos, com base na teoria estudada;
- Falta de organização, por parte dos alunos, de agenda de estudo;
- Dificuldades dos alunos em lidar com a Internet e seus recursos, como formulários e correio eletrônico;
- Flexibilidade para programar horários de estudo, fator relevante, na nossa realidade de alunos que trabalham;
- Ferramentas disponíveis: não foram suficientes ou adequadas para propiciar a interação e colaboração on-line, entre os alunos;
- Realização de tarefas de aprendizagem propostas: foi constatado que os alunos se reuniam em grupos, presencialmente, para estudos e realização dessas tarefas.

De um modo geral, os ambiente têm oferecido bons meios que promovam o aprendizado. Porém, não é uma solução para os problemas de aprendizagem existentes hoje. Para que os ambientes consigam atingir resultados melhores deve ocorrer uma mudança de paradigma no que se refere a questão educacional com intuito de fornecer um conteúdo bem estruturado e formas de avaliações diferenciadas.

## **2.6 ERGONOMIA E USABILIDADE**

Inicialmente, os softwares eram desenvolvidos pelos seus próprios usuários. Com o passar do tempo, o público alvo passou a ser um pequeno grupo o qual era treinado para utilizar o sistema. A partir do momento em que se passou a ver o software como um produto e serviço onde o público passou a ser mais amplo e menos treinado se começou a perceber a importância das interfaces. Interfaces mal projetadas dificultavam mais do que ajudava no trabalho dos usuários, em alguns casos, causando um forte impacto na maneira como os mesmos desenvolvessem as estratégias que irão utilizar em suas atividades (NIELSEN, 2003).

Experiências negativas entre o usuário e o software pode gerar uma baixa auto-estima, aborrecimentos, frustrações, quadros de ansiedade, estresse, irritação, até mesmo crises de pânico.

As causas para isso resumem-se no desconhecimento das atividades, desconhecimento do cognitivo humano e o desinteresse pela lógica de utilização (CYBIS,2003).

O desenvolvimento de sistemas com boa usabilidade irá impactar a tarefa no sentido da eficiência, eficácia, produtividade da interação. O usuário irá atingir plenamente seus objetivos com menos esforço e mais satisfação (CYBIS,2003).

A usabilidade segundo a norma ISO 9241 é a capacidade que um sistema oferece ao seu usuário para a realização de tarefas de uma maneira eficaz, eficiente e agradável.

A usabilidade compreende sete componentes de qualidade (PREECE, 2002 , NIELSEN, 2003). Os sete componentes de qualidade inerentes ao termo usabilidade são:

- Facilidade de aprendizagem: refere-se a facilidade com que se aprende a utilizar a interface. Os usuários não gostam de despende de muito tempo para aprender a usar algo.
- Facilidade de memorização: refere-se a quão facilmente o usuário lembra do modo de utilização da interface depois de ter compreendido seu funcionamento. Se as operações a aprender são ilógicas, obscuras ou com seqüências pobres a tendência é esquecer como utilizar a interface nos casos de pouco uso.
- Eficiência: mede o nível de produtividade.
- Segurança: envolve a proteção do utilizador de condições perigosas e situações indesejáveis. Há que prevenir que o utilizador cometa erros graves reduzindo o risco de ativação de botões ou teclas erradas e dar aos utilizadores vários meios de recuperação caso algum erro seja cometido. O sistema não pode induzir o usuário ao erro.
- Satisfação: mede o quanto os utilizadores gostaram da interface.
- Eficácia: mede a adequação da interface.
- Utilidade: refere-se à capacidade da interface apresentar as funcionalidades certas para que o utilizador atinja o seu objetivo.

No aspecto de design PREECE (2002) introduz alguns princípios de design. São eles:

- Visibilidade: quanto mais visíveis estiverem as funções mais facilmente o utilizador saberá o que fazer de seguida.
- Feedback (ação/reação): utilizar feedback na forma e quantidade certa pode sortir a visibilidade necessária para a interação com o utilizador
- Restrições: diz respeito à determinação de maneiras de restringir a interação num determinado momento. Em interfaces gráficas é usual desativar certas opções do menu restringindo assim o utilizador às ações permitidas nesse estágio da atividade. Uma das

vantagens destas restrições é impedir o utilizador de seleccionar uma opção incorreta e assim reduzir o risco de erro.

- **Consistência:** operações semelhantes devem ter objetivos idênticos. Se numa janela um botão azul abre um documento e um vermelho o grava, na janela seguinte o botão azul pode mostrar um gráfico e o vermelho gravá-lo, por exemplo. Uma interface consistente é mais facilmente aprendida e utilizada.
- **Atribuição Correta:** refere-se aos atributos de um objecto que permitem a uma pessoa saber como o utilizar. Um campo em branco sugere a ação “escrever aqui”, um botão com relevo traduz “clique aqui”, etc.

Conhecer os componentes de qualidade de interface e alguns princípios de design contribuem para a construção de interfaces melhores e que evitem o desgaste do utilizador elevando a sua auto-estima e permitindo que o sistema contribua de forma eficiente nas tarefas desejadas pelo usuário.

### **3 AMBIENTE DE APRENDIZAGEM DE PROGRAMAÇÃO DE COMPUTADORES**

O estudo sobre os ambientes de aprendizagem e as idéias obtidas durante a manipulação das interfaces resultou com que esse projeto não se restrinja a apenas uma interface para o sistema SIA-TP e sim, um ambiente de aprendizagem que poderá abrigar diversas futuras funcionalidades.

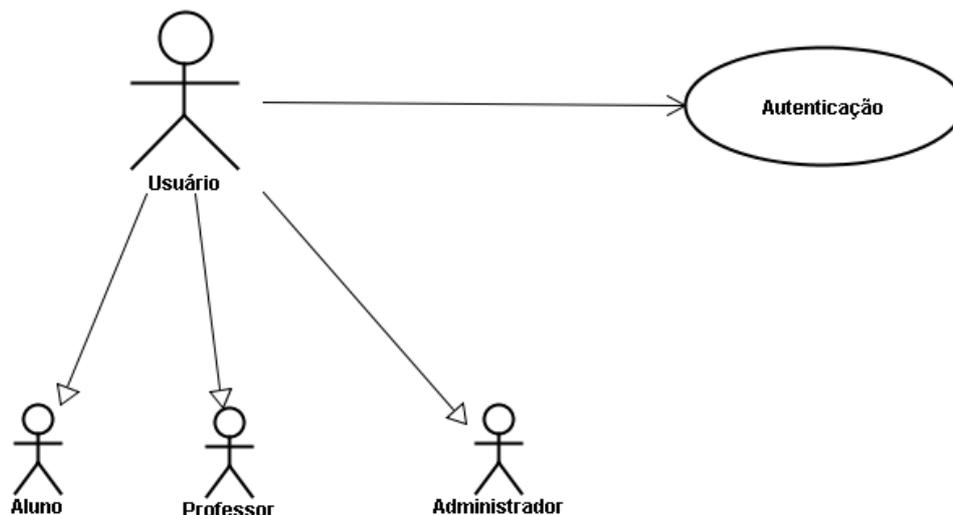
O objetivo desse projeto não é desenvolver um compilador, muito menos que seja uma ferramenta que induza o aluno ao acerto. A meta é que o aluno aprenda com os seus erros e através do auxílio do professor ou dos demais participantes construa seu conhecimento.

Com base nos ambientes anteriormente citados levantou-se os seguintes requisitos para a construção do ambiente desenvolvido nesse trabalho.

#### **3.1 REQUISITOS DO AMBIENTE**

Os papéis envolvidos nesse ambiente são o Aluno, Professor e o Administrador. Entende-se por usuário qualquer utilizador do sistema, desta forma, contempla Aluno, Professor e Administrador. O papel Aluno representa o estudante que utiliza o sistema. O papel Professor representa o tutor da disciplina responsável por montar o conteúdo e orientar o aluno. O Administrador possui o papel de criar as turmas e cadastrar os professores.

A Ilustração 18 apresenta o caso de uso referente ao usuário. A Tabela 2 descreve a funcionalidade apresentada no caso de uso.

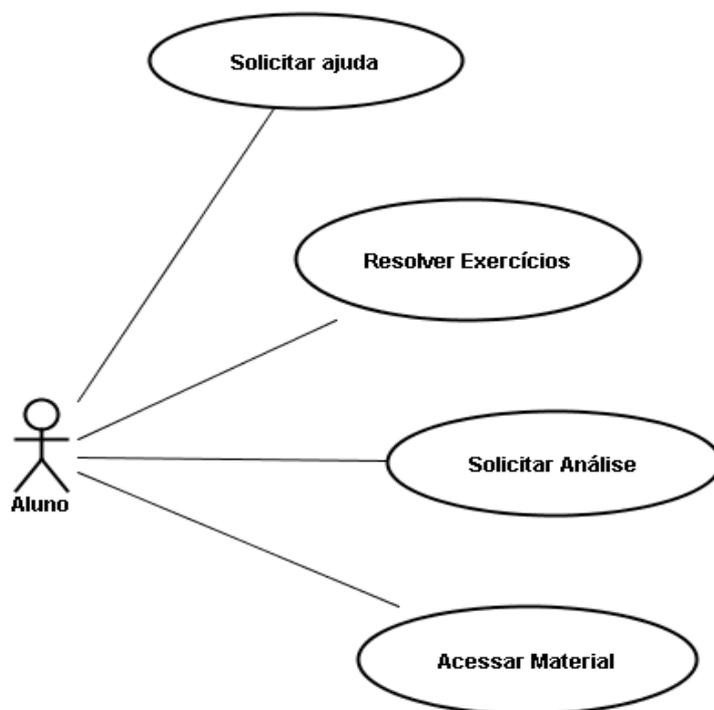


**Ilustração 18: Caso de Uso Referente aos Usuários**

**Tabela 2: Descrição da Funcionalidade Autenticar**

<b>Nome:</b>	Autenticar
<b>Atores:</b>	Aluno, Professor, Administrador
<b>Finalidade:</b>	Autenticar o usuário no sistema.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade efetuar a autenticação do usuário no sistema .
<b>Tipo:</b>	Essencial
<b>Ações:</b>	O aluno ou professor ou informa seu email e sua senha. O sistema valida os dados Caso os dados estejam corretos, o sistema permite o acesso e mostra a tela principal do sistema. Caso os dados não estejam corretos, o sistema avisa que os dados não estão corretos e se mantem na tela de autenticação.
<b>Tabelas Relacionadas:</b>	SystemUser
<b>Regras:</b>	Para a autenticação a interface recebe como entrada de dados o email e a senha do usuário. A mesma criptografa a senha utilizando o algoritmo MD5 e transfere as informações para a camada de negócio.  A camada de negócio por sua vez busca na tabela SystemUser por um usuário que possua um email e senha correspondentes as informadas. Caso encontre um usuário retornar um objeto do tipo SystemUser. Caso contrário, lançar uma exceção com a mensagem de “Usuário ou Senha inválidos”.

A Ilustração 19 apresenta um caso de uso de um aluno que interage com o sistema. As Tabelas 3, 4, 5 e 6 descrevem as funcionalidades apresentadas no caso de uso.



**Ilustração 19: Caso de Uso referente ao Aluno**

**Tabela 3: Descrição da Funcionalidade Solicitar Ajuda**

<b>Nome:</b>	Solicitar Ajuda
<b>Atores:</b>	Aluno
<b>Finalidade:</b>	Solicitar ajuda de algum aluno ou professor mediante a uma conversação.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer ao aluno uma ferramenta de conversação na qual o mesmo possa sanar suas duvidas diretamente com o professor ou com a ajuda de seus colegas
<b>Tipo:</b>	Essencial
<b>Ações:</b>	O aluno cria uma sala de mensagens para a determinada duvida. O aluno adiciona os alunos que deseja que participem da conversa. O sistema fornece um sistema de envio e recebimento de mensagens. O professor sempre será adicionado as conversações afim de saber o que está sendo discutido e fazer seus pareceres caso ache necessário.

<b>Observações:</b>	Somente será possível adicionar nas conversas os Alunos que possuem a flag “Aceito Ajudar” no seu perfil ligada.
<b>Tabelas Relacionadas:</b>	Message, GroupUsers, MessageGroup, SystemUser
<b>Regras:</b>	<p>Quando o usuário estiver criando uma nova conversação deve-se criar na tabela MessageGroup um registro contendo o título da mensagem. Após, criar registros na GroupUsers associado ao registro MessageGroup para cada participante adicionado na conversa. As mensagens trocadas entre os usuários nessa sala de conversa devem ser gravadas na tabela Message.</p> <p>Para os casos onde o usuário irá adicionar um participante ou responder alguma mensagem numa conversação já cadastrada deve-se apenas buscar a MessageGroup correspondente e criar um registro nas tabelas relacionadas associando a esse registro.</p> <p>A lista de mensagens deve conter apenas as mensagens cujo o usuário logado no sistema esteja participando. Assim, a busca para preencher essa lista deve ser feita verificando todas as MessagesGroup que contenham em seu GroupUsers relacionado um registro para com o usuário logado no sistema.</p> <p>Na tela de criação de mensagem / resposta, para preencher a lista com os participantes que poderão ser incluídos. Deve-se buscar todos os usuários do sistema e que já não estejam adicionados na conversação e que possuam o atributo isHelper = 1.</p>

**Tabela 4: Descrição da Funcionalidade Resolver Exercícios**

<b>Nome:</b>	Resolver Exercícios
<b>Atores:</b>	Aluno
<b>Finalidade:</b>	Permitir que o usuário possa resolver o exercício proposto.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer ao usuário o exercício a ser resolvido e, um ambiente para que o mesmo possa desenvolver o seu algoritmo.
<b>Tipo:</b>	Essencial
<b>Ações:</b>	O aluno seleciona um exercício proposto. O sistema abre uma aba contendo os dados do exercício, além de um editor onde o aluno irá desenvolver sua solução. O aluno poderá enviar seu algoritmo ao professor para que seja corrigido.
<b>Tabelas Relacionadas:</b>	Exercise, Solution
<b>Regras:</b>	Quando o usuário solicitar por salvar o exercício. Deve-se criar um registro na tabela Solution associado ao Exercício o qual o aluno está resolvendo. O atributo status deve possuir valor igual a 0. O atributo level deve possuir valor 0.

**Tabela 5: Descrição da Funcionalidade Solicitar Análise**

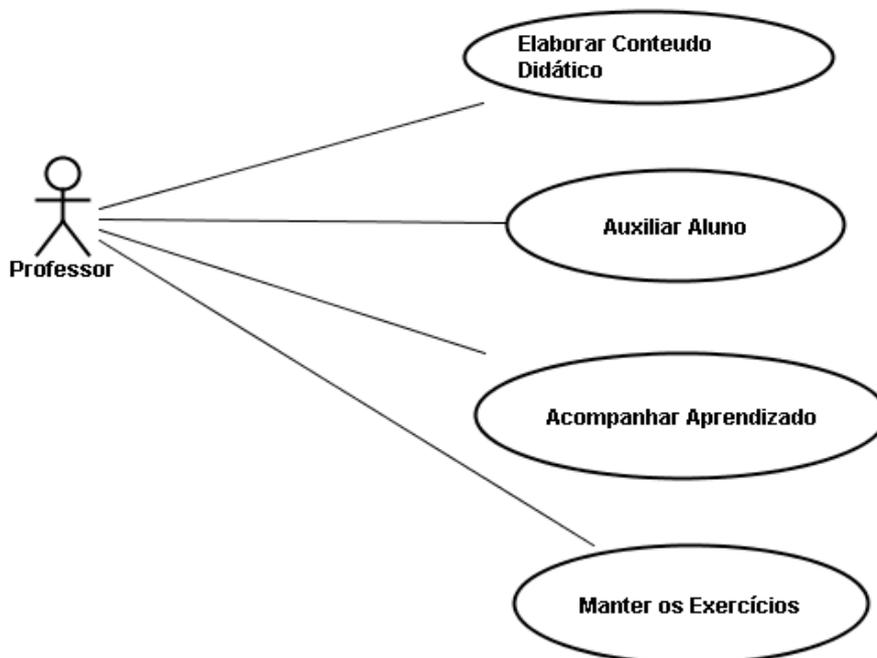
<b>Nome:</b>	Solicitar Análise
<b>Atores:</b>	Aluno
<b>Finalidade:</b>	Enviar o algoritmo desenvolvido para análise do SIA-TP ou para o Professor.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade enviar o algoritmo para o sistema analisar e retornar seu resultado para o aluno.
<b>Tipo:</b>	Essencial
<b>Ações:</b>	Fluxo 1: O aluno conclui seu algoritmo. O aluno clica em Analisar. O sistema SIA-TP analisa o algoritmo desenvolvido e retorna o resultado da análise para o aluno.  Fluxo 2: O aluno conclui seu algoritmo. O aluno clica em enviar para o professor. O aluno informar o grau de dificuldade para o exercício. O sistema bloqueia a interação do usuário para com aquele algoritmo e envia para o professor.
<b>Tabelas Relacionadas:</b>	Solution, SIATPAnalysisHistory
<b>Regras:</b>	Fluxo 1: Chamar a classe analisadora do sistema SIA-TP. Com o resultado criar um registro na tabela SIATPAnalysisHistory armazenado o algoritmo analisado e o resultado da análise.  Fluxo 2: Modificar o valor do atributo status da tabela Solution para 1. Modificar o valor do atributo level da tabela Solution para o valor informado pelo usuário.

**Tabela 6: Descrição da Funcionalidade Acessar Material**

<b>Nome:</b>	Acessar Material
<b>Atores:</b>	Aluno
<b>Finalidade:</b>	Acessar o material contendo o conteúdo.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer o acesso ao conteúdo disciplinar disponibilizado pelo professor.
<b>Tipo:</b>	Essencial

<b>Ações:</b>	O aluno seleciona o tópico do conteúdo a ser visualizado. O sistema abre uma aba contendo o conteúdo disponibilizado pelo professor. O sistema possibilita o download do conteúdo.
<b>Tabelas Relacionadas:</b>	ContentFolder, Content
<b>Regras:</b>	Para preencher a lista com o conteúdo deve-se buscar todos os registros da tabela ContentFolder cujo team corresponda ao do usuário logado. Agregase nessa pesquisa a busca na tabela Content utilizando o atributo Team e/ou ContentFolder. A tabela Content representa um arquivo de conteúdo. A tabela ContentFolder representa uma pasta. Logo, a pesquisa deve buscar os arquivos que estiverem dentro de pastas assim como os que não possuem ligação com nenhuma pasta.  Se existir valor no campo PathFile da tabela Content então, disponibilizar um link para download utilizando o nome do arquivo salvo nesse campo.

A Ilustração 20 apresenta o caso de uso de um professor que interage com o ambiente. As Tabelas 7, 8, 9 e 10 descrevem as funcionalidades apresentadas no caso de uso.



**Ilustração 20: Caso de Uso referente ao professor**

**Tabela 7: Descrição da Funcionalidade Elaborar Material Didático**

<b>Nome:</b>	Elaborar Material Didático
<b>Atores:</b>	Professor
<b>Finalidade:</b>	Fornecer o conteúdo didático ao aluno
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer ao professor as ferramentas necessárias para que ele possa disponibilizar o conteúdo didático aos alunos
<b>Tipo:</b>	Baixo
<b>Ações:</b>	<p>O professor visualiza a listagem de conteúdo.</p> <p>Para adicionar um novo conteúdo clica em “Adicionar”. O sistema apresenta a tela de cadastro de conteúdo. Após a confirmação o conteúdo será criado.</p> <p>Para modificar um conteúdo. Seleciona-se um conteúdo disponível na listagem e clica em “Modificar”. O sistema apresenta a tela que permite a realização da modificação. Após a confirmação o conteúdo será alterado.</p> <p>Para excluir um exercício. Seleciona-se um conteúdo disponível na listagem e clica em “Remover”. O sistema solicita a confirmação de exclusão. Em caso positivo remove o conteúdo.</p>
<b>Tabelas Relacionadas:</b>	Content, ContentFolder
<b>Regras:</b>	<p>Se for solicitado a criação de uma pasta deve-se criar um registro na tabela ContentFolder com o título informado e associar a turma referente a do usuário logado no sistema.</p> <p>Caso seja uma modificação, acessar o registro correspondente a ContentFolder e alterar o título.</p> <p>Caso seja solicitado a remoção do registro, acessar o registro correspondente e o remover.</p> <p>Quando for solicitado uma inclusão de arquivo, deve-se criar um registro na tabela Content. Caso o usuário realize um upload de arquivo então deve-se armazenar o arquivo dentro da pasta StudentBookAttachment contida dentro do EAR da aplicação e salvar o nome do arquivo no campo PathFile da tabela Content.</p> <p>Se solicitado uma alteração no arquivo, deve-se acessar o registro correspondente e o modificar.</p> <p>Se solicitado a remoção do arquivo, deve-se acessar o registro correspondente e remove-lo da base de dados</p>

**Tabela 8: Descrição da Funcionalidade Auxiliar Aluno**

<b>Nome:</b>	Auxiliar Aluno
--------------	----------------

<b>Atores:</b>	Professor
<b>Finalidade:</b>	Responder as duvidas do aluno mediante a mensagens
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer ao professor as ferramentas necessárias para que ele possa conversar com o seu aluno afim de sanar as duvidas do mesmo
<b>Tipo:</b>	Essencial
<b>Ações:</b>	O professor visualiza a mensagem recebida do aluno. O professor responde
<b>Tabelas Relacionadas:</b>	Message, GroupUsers, MessageGroup, SystemUser
<b>Regras:</b>	<p>Quando o usuário estiver criando uma nova conversa deve-se criar na tabela MessageGroup um registro contendo o título da mensagem. Após, criar registros na GroupUsers associado ao registro MessageGroup para cada participante adicionado na conversa. As mensagens trocadas entre os usuários nessa sala de conversa devem ser gravadas na tabela Message.</p> <p>Para os casos onde o usuário irá adicionar um participante ou responder alguma mensagem numa conversa já cadastrada deve-se apenas buscar a MessageGroup correspondente e criar um registro nas tabelas relacionadas associando a esse registro.</p> <p>A lista de mensagens deve conter apenas as mensagens cujo o usuário logado no sistema esteja participando. Assim, a busca para preencher essa lista deve ser feita verificando todas as MessagesGroup que contenham em seu GroupUsers relacionado um registro para com o usuário logado no sistema.</p> <p>Na tela de criação de mensagem / resposta, para preencher a lista com os participantes que poderão ser incluídos. Deve-se buscar todos os usuários do sistema e que já não estejam adicionados na conversa e que possuam o atributo isHelper = 1.</p>

**Tabela 9: Descrição da Funcionalidade Acompanhar Aprendizado**

<b>Nome:</b>	Acompanhar Aprendizado
<b>Atores:</b>	Professor
<b>Finalidade:</b>	Permitir que o professor visualize as análises sobre o algoritmo desenvolvido pelo aluno.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer ao professor as ferramentas necessárias para que ele visualize os resultados das análises geradas sobre o algoritmo desenvolvido pelo aluno
<b>Tipo:</b>	Essencial

<b>Ações:</b>	<p>O professor seleciona o aluno o qual deseja acompanhar os resultados. O ambiente traz o histórico de análises do aluno e seus resultados.</p> <p>O professor visualiza os algoritmos os quais foram solicitados pelo aluno que sejam corrigidos. O professor realiza a correção informando se o algoritmo está correto ou não e informa o seu parecer.</p>
<b>Tabelas Relacionadas:</b>	Solution, SIATPAnalysisHistory, TeacherAnalysisHistory, Exercise, Team, SystemUser
<b>Regras:</b>	<p>Para preencher a lista dos alunos os quais serão auxiliados pelo professor buscar os alunos na tabela SystemUser que pertençam a mesma turma (Team) do professor.</p> <p>Para preencher a lista correspondente com os Exercise que possuam associação a Team referente ao professor.</p> <p>Quando selecionado um exercício na listagem, buscar todos os registros da tabela Solution que correspondam ao Exercise selecionado e que possua o campo status = 2.</p> <p>Quando selecionado a solução desejada. Buscar os registros da tabela SIATPAnalysisHistory, TeacherAnalysisHistory para preencher as listagens Analise do SIATP e Analise do Professor.</p>

**Tabela 10: Descrição da Funcionalidade Manter Exercícios**

<b>Nome:</b>	Manter Exercícios
<b>Atores:</b>	Professor
<b>Finalidade:</b>	Manter o cadastro e configuração dos exercícios.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer ao professor as ferramentas necessárias para que ele possa cadastrar, editar e excluir os exercícios no ambiente. Esse cadastro deve permitir a configuração das áreas de conhecimento para que o SIA-TP possa realizar a análise.
<b>Tipo:</b>	Essencial
<b>Ações:</b>	<p>O professor visualiza a listagem de exercícios.</p> <p>Para adicionar um novo exercício clica em “Adicionar”. O sistema apresenta a tela de cadastro de exercícios. Após a confirmação o exercício será criado.</p> <p>Para modificar um exercício. Seleciona-se um exercício disponível na listagem e clica em “Modificar”. O sistema apresenta a tela que permite a realização da modificação. Após a confirmação o exercício será alterado.</p> <p>Para excluir um exercício. Seleciona-se um exercício disponível na listagem e clica em “Remover”. O sistema solicita a confirmação de exclusão. Em caso positivo remove o exercício.</p>
<b>Tabelas Relacionadas:</b>	Exercise, Team

<b>Regras:</b>	<p>Quando solicitado a criação de um novo exercício criar um registro na tabela Exercise associado ao Team (turma) correspondente ao do Professor logado. No cadastro de exercícios o professor pode configurar o os limiars e a área do conhecimento correspondente ao exercício.</p> <p>Quando solicitado alteração, acessar o registro correspondente ao Exercise selecionado e modificar as suas informações.</p> <p>Quando solicitado a exclusão do registro deve-se acessar o registro correspondente e remover-lo da base de dados.</p>
----------------	--

A Ilustração 21 apresenta o caso de uso de um administrador que interage com o ambiente. A tabela descreve as funcionalidades apresentadas no caso de uso.



**Ilustração 21: Caso de Uso Referente ao Administrador**

<b>Nome:</b>	Criar Turma / Professor
<b>Atores:</b>	Administrador
<b>Finalidade:</b>	Cadastrar a Turma e o Professor.
<b>Visão Geral:</b>	Este caso de uso tem por finalidade fornecer ao administrador as ferramentas necessárias para que ele possa cadastrar as turmas e os professores.
<b>Tipo:</b>	Essencial
<b>Ações:</b>	O administrador visualiza a tela de cadastro. Para cadastrar preenche-se o código da turma e os dados do professor. Pressiona-se salvar. O sistema consiste os dados e cria a nova turma associando a ela o professor recém criado.

### 3.2 INTERFACE E FUNCIONALIDADES

A interface apresentada nesse trabalho, foi modelada seguindo uma estrutura de abas. Essa forma possibilita que o usuário possa realizar diversas atividades simultaneamente. A primeira aba

apresentada após a autenticação é a única a qual o usuário não poderá fechar pois é nela que o mesmo acessará todos os recursos disponíveis. Essa aba prove apenas as informações pertinentes ao usuário logado, possibilitando assim um acesso rápido às suas atividades. As funcionalidades sempre estarão disponíveis mediante a abertura de novas abas ou através de janelas *popups*.

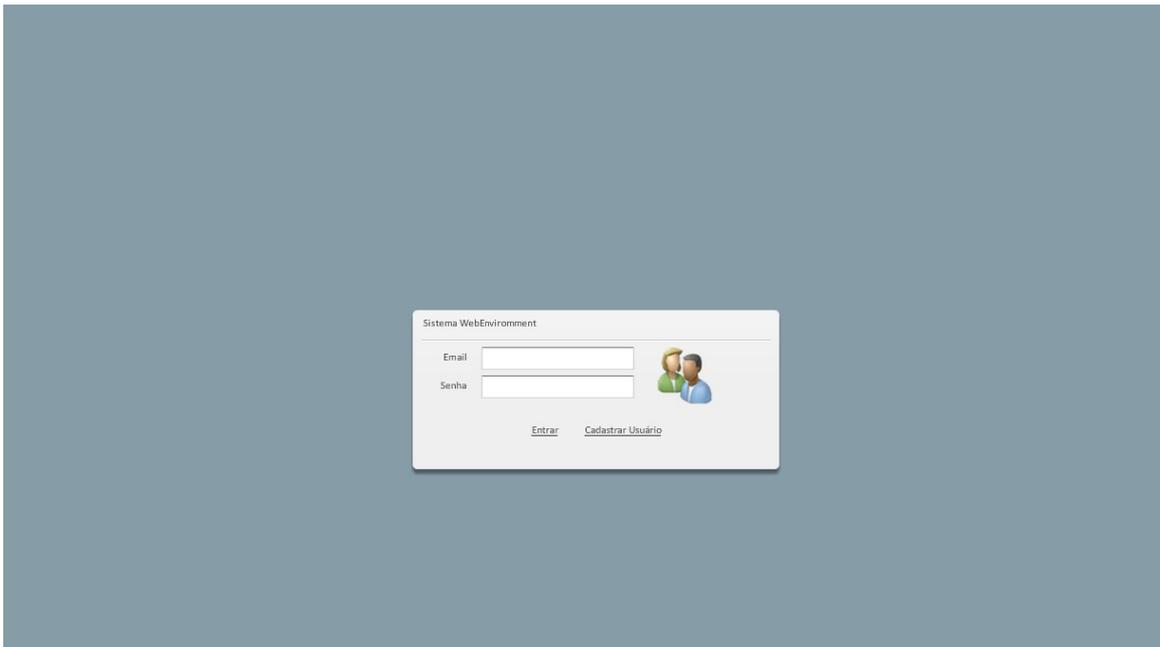
Com o intuito de atender aos requisitos levantados anteriormente, a interface foi projetada numa estrutura de componentes. Cada componente busca atender a funções específicas e interagir com os demais componentes. Isso possibilitará maior facilidade na construção do sistema propriamente dito.

Os componentes que compõem a interface são: Autenticação e o Ambiente Caderno de Exercício.

### **3.2.1 COMPONENTE DE AUTENTICAÇÃO**

Neste componente, a interface contém campos nos quais o usuário irá informar seus dados de email e sua senha.

A tela deve solicitar que sejam informados o email e a senha do usuário. Caso estes campos não sejam informados, o sistema deve destacar o campo com outra cor e informar ao usuário que o campo é obrigatório. A Ilustração 22 apresenta a interface gerada a partir das necessidades levantadas.



**Ilustração 22: Interface Autenticação**

Para a autenticação o sistema recebe como entrada de dados o usuário e a senha definidos em seu cadastro. Quando o usuário solicita o acesso ao sistema, o sistema codifica a senha utilizando a criptografia MD5 e envia essas informações para a camada de negócio. A camada de negócio valida as informações com a base de dados. Caso as informações estejam corretas o sistema descarrega o módulo de autenticação e carrega o módulo do ambiente possibilitando ao usuário acesso ao sistema. Caso contrário a tela mostra ao usuário o resultado da exceção à regra.

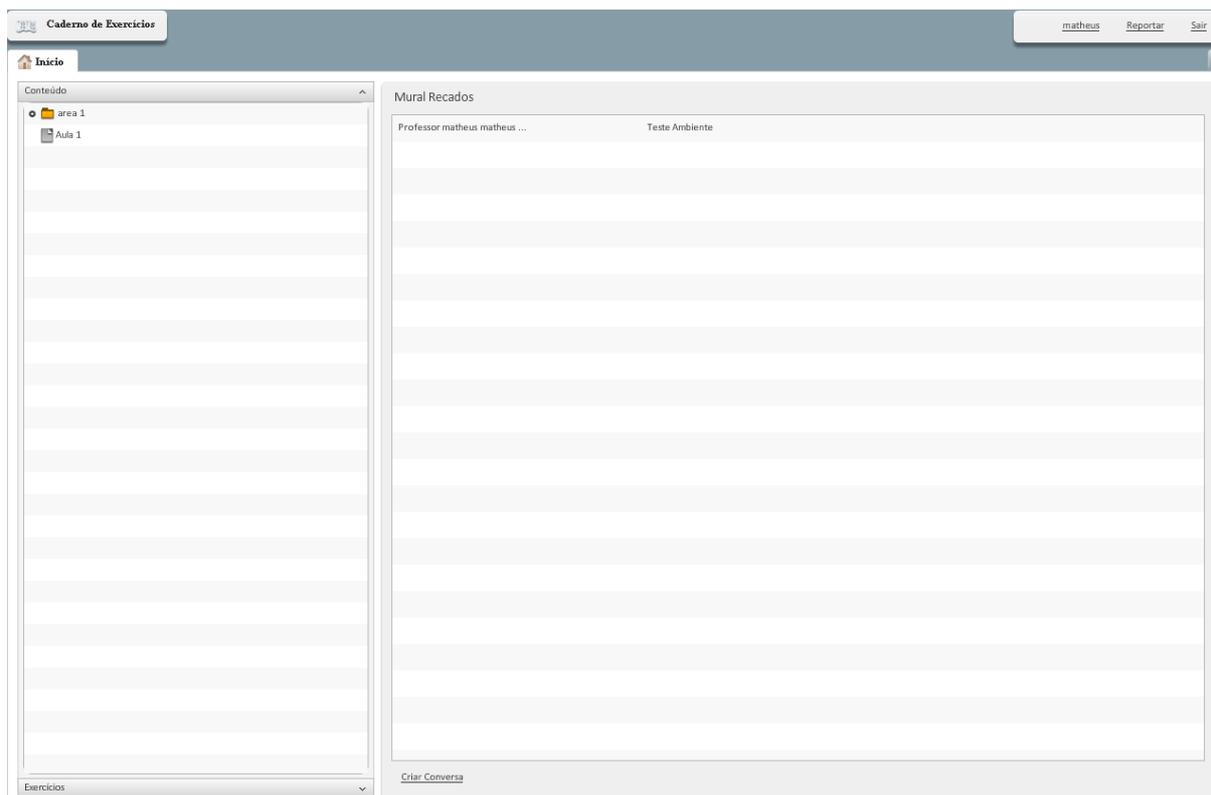
### **3.2.2 COMPONENTE AMBIENTE CADERNO DE EXERCÍCIOS**

Somente a partir da autenticação na aplicação o usuário poderá ter acesso ao ambiente. Esse componente engloba diversos subcomponentes os quais contemplam funcionalidades distintas. São eles: Mural de Recados, Conteúdo Disciplinar, Exercícios, Acompanhar Aluno, Cadastro Turma e Professor.

Com base no tipo de usuário o ambiente se molda devido as características de cada papel.

- Para o caso de um aluno:

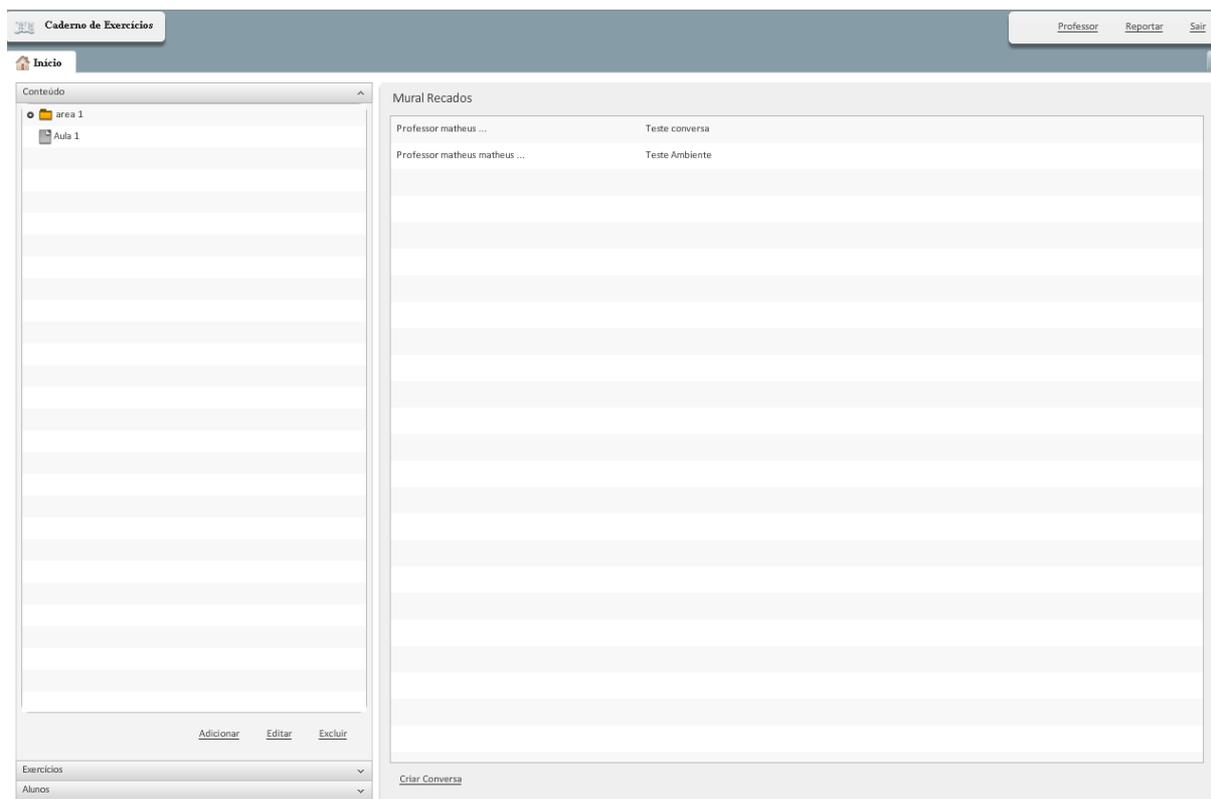
O ambiente apresenta a interface contendo o Mural de Recados, o Conteúdo Disciplinar e os Exercícios disponíveis. A Ilustração 23 demonstra a interface correspondente a um usuário cujo papel é de Aluno.



**Ilustração 23: Exemplo de Interface para o papel Aluno**

- Para o caso de um professor:

O papel de professor concede uma interface contendo o Mural de Recados, o Conteúdo Disciplinar, os Exercícios e o Acompanhar Aluno. A Ilustração 24 demonstra a interface de um usuário cujo papel é Professor.



**Ilustração 24: Exemplo de Interface para o papel Professor**

- Para o caso de um administrador:

O usuário que possui esse papel terá acesso a uma interface que contém o Cadastro Turma e Professor. A Ilustração 25 demonstra a interface cujo papel corresponde a administrador.

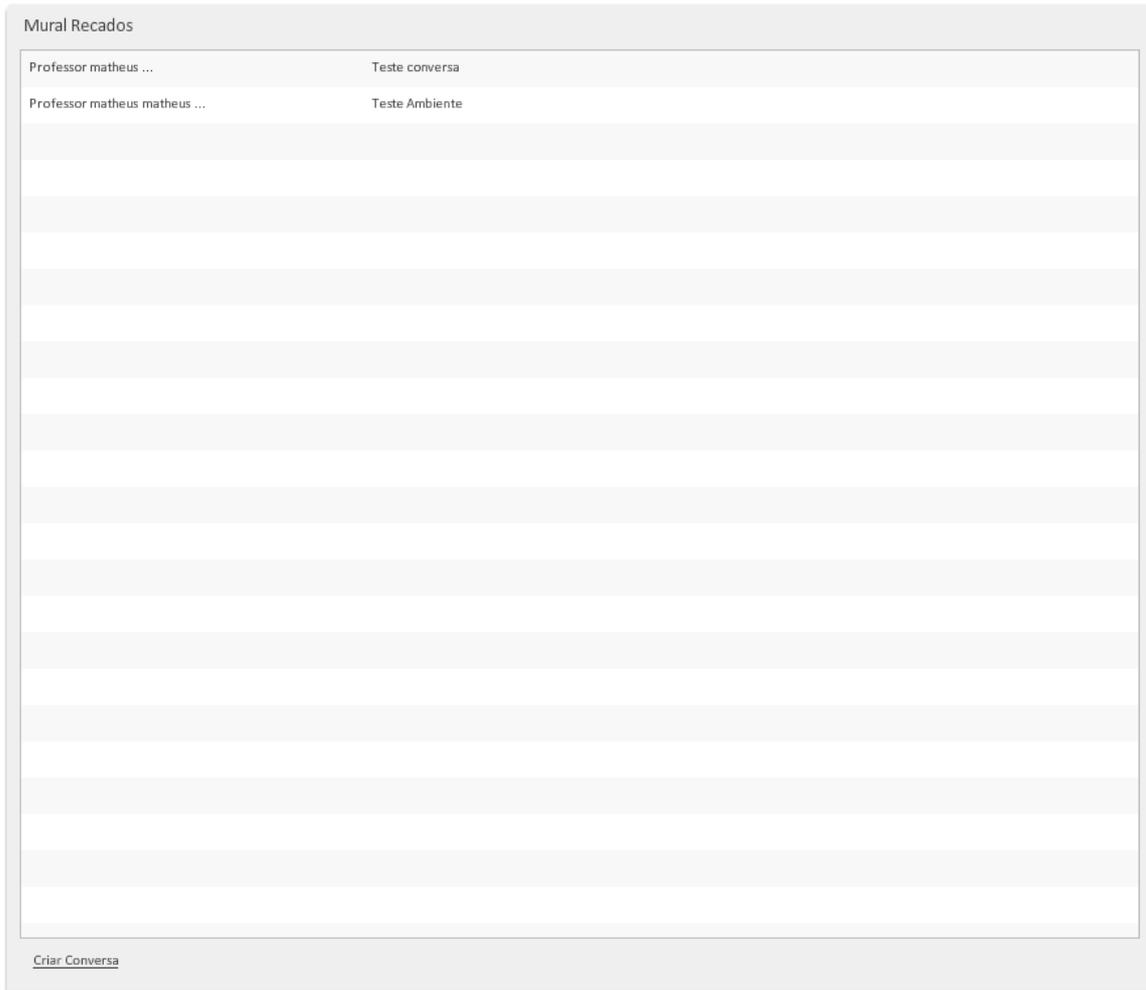


**Ilustração 25: Exemplo de Interface para o papel Administrador**

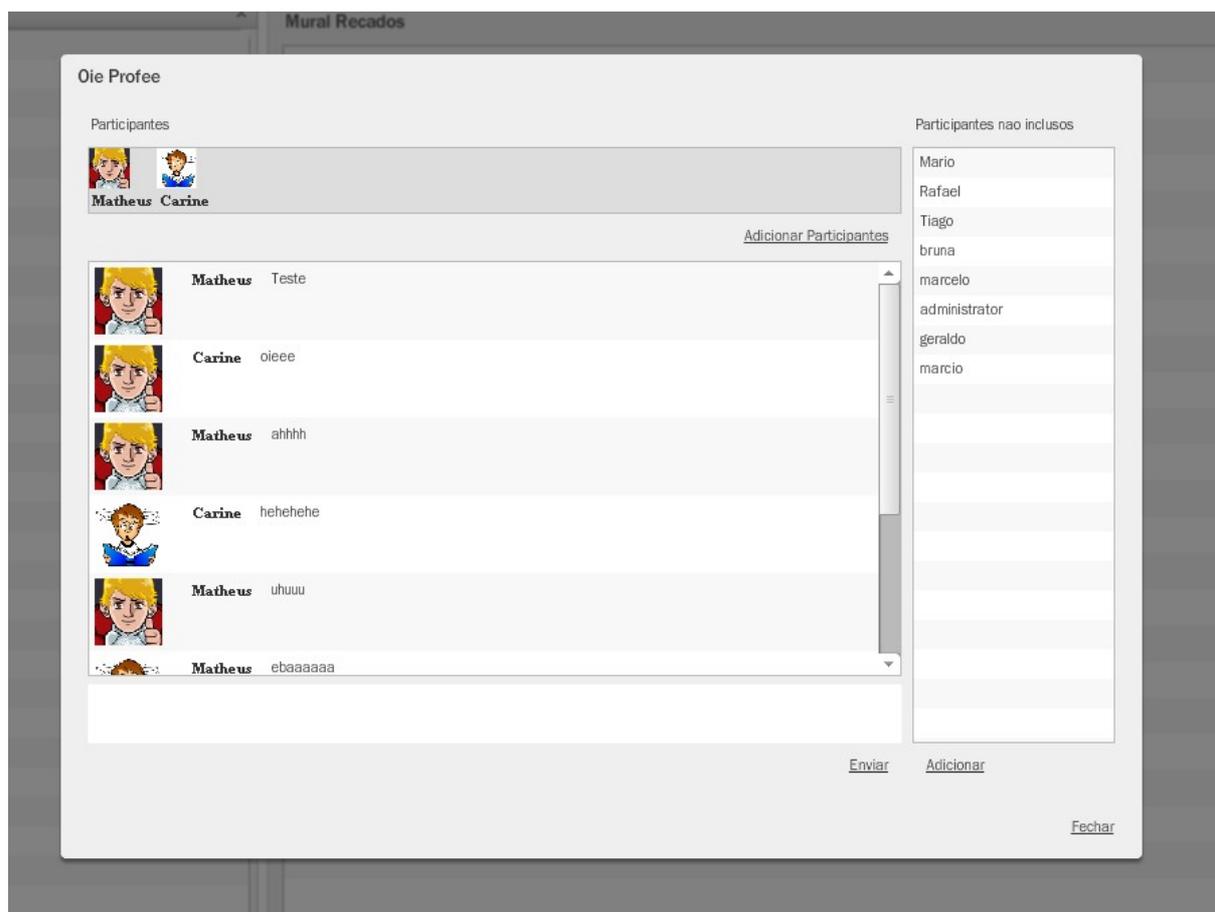
### Mural de Recados

O sistema de recados é o componente responsável pela comunicação do aluno com os demais alunos envolvidos e com o próprio professor. A interface permite que o usuário crie um recado e que possa informar quais usuários participarão dessa conversa. A qualquer momento novos usuários podem ser adicionados a discussão. A Ilustração 26 busca mostrar em forma de lista as mensagens as quais o usuário autenticado está participando. A Ilustração 27 demonstra uma janela de

conversação entre usuários.



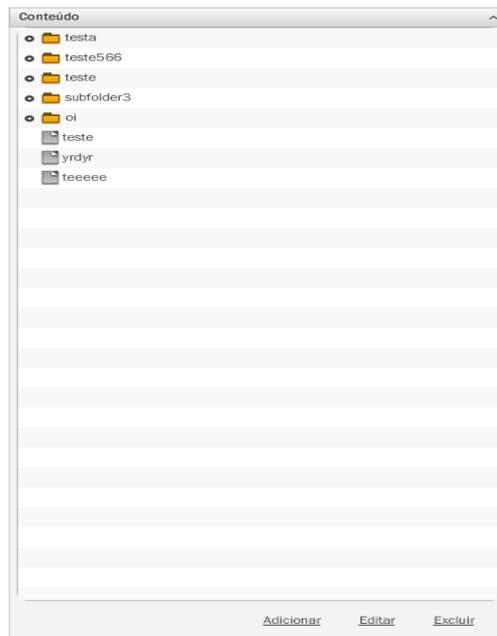
**Ilustração 26: Interface do Mural de Recados**



**Ilustração 27: Interface do Janela de Conversa**

### Conteúdo Disciplinar

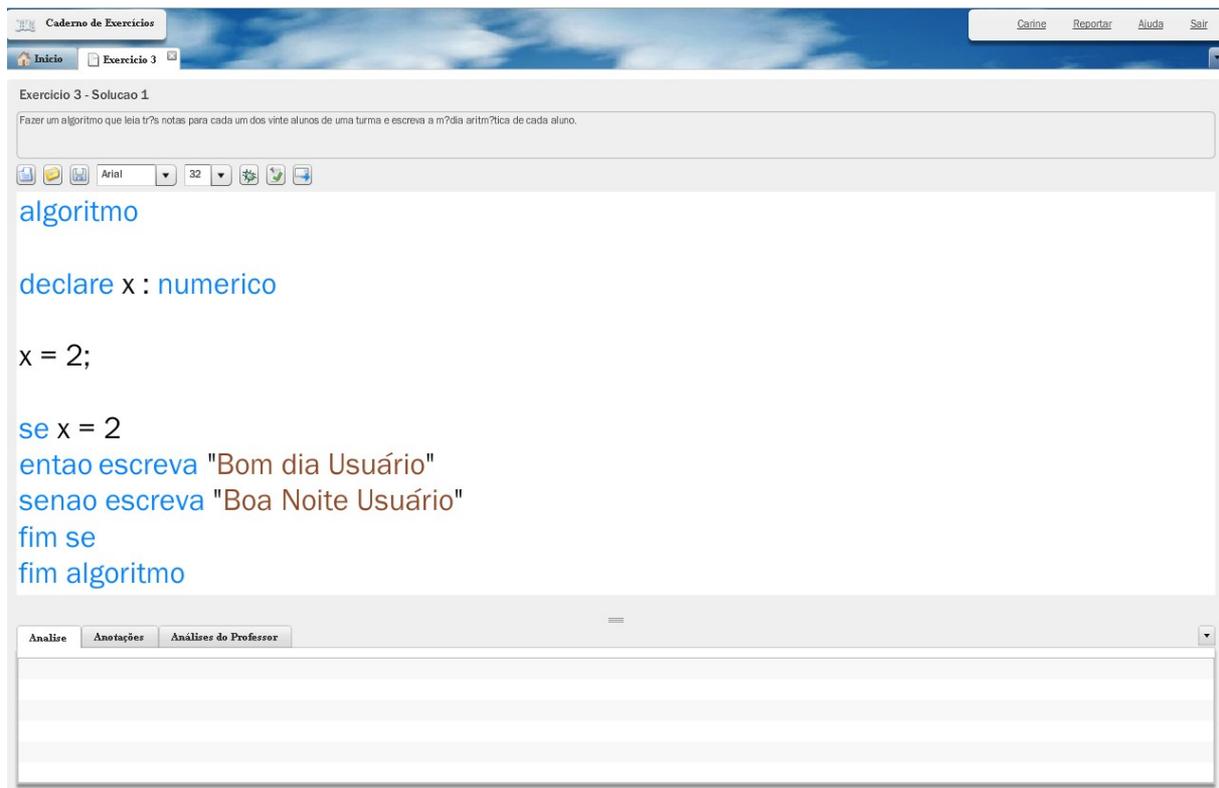
Esse componente permite que o professor adicione, modifique ou remova o conteúdo disciplinar referente a disciplina de algoritmos. O aluno, por sua vez, pode realizar o download desse material para fins de estudo. O sistema permite uma estrutura de pastas a fim de uma melhor organização. Cabe ao professor decidir a sua forma. O material deve estar obrigatoriamente no formato PDF. A Ilustração 28 demonstra a tela em questão.



**Ilustração 28: Manutenção do Conteúdo Disciplinar**

### Exercícios

O componente de Exercícios possibilita ao professor adicionar, editar ou remover exercícios. O exercício nada mais é do que uma tarefa para o aluno. Essa tarefa consiste na elaboração de um algoritmo. O aluno pode então, desenvolver sua solução na tela correspondente ao exercício que deseja resolver. O ambiente proporciona um editor de algoritmos, a possibilidade de salvar o seu trabalho, de realizar análise sobre o algoritmo e de envio para o professor. A Ilustração 29 demonstra a interface utilizada pelo aluno para desenvolver o exercício proposto.



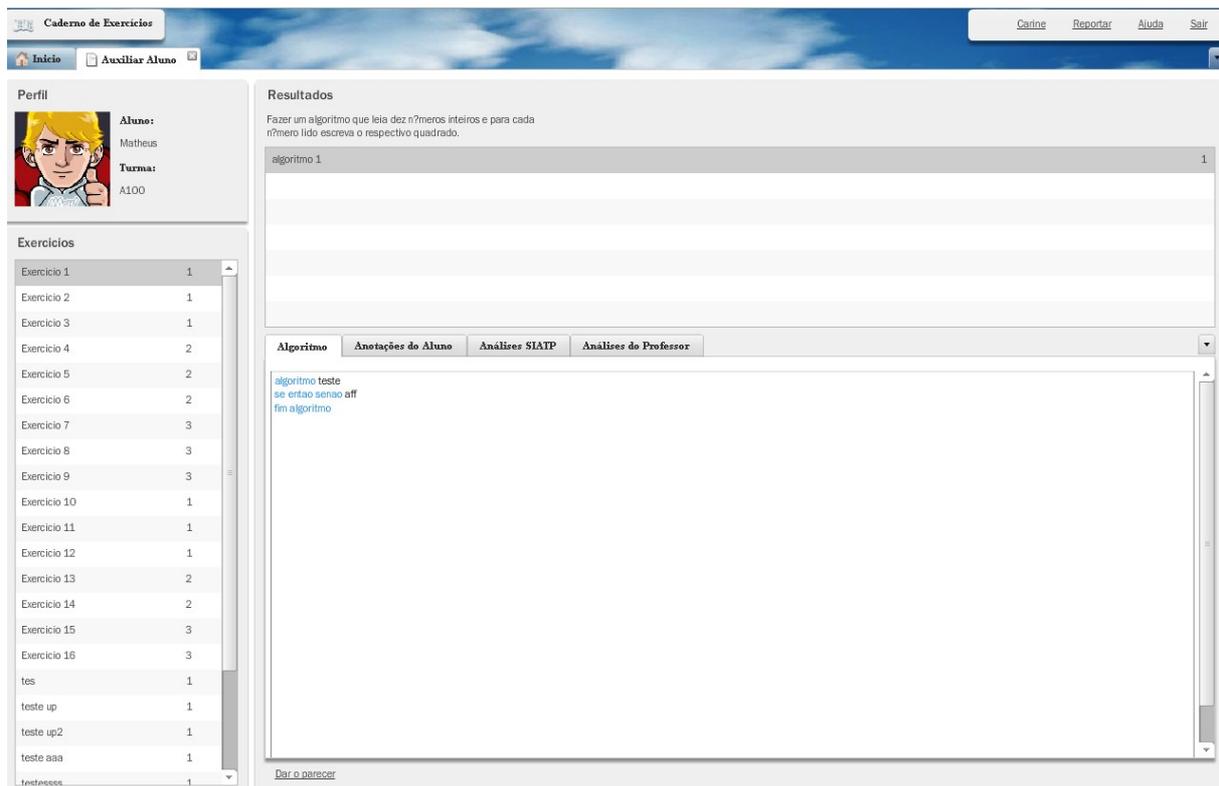
**Ilustração 29: Interface para o desenvolver o exercício**

### Acompanhar Aluno

O componente de acompanhamento do aluno é utilizado apenas pelo professor. Contém as seguintes características:

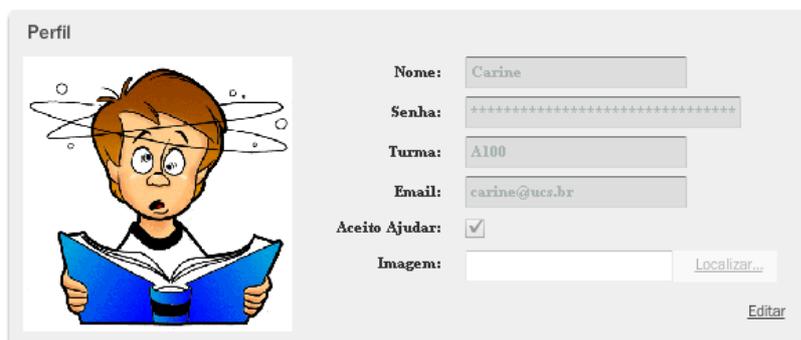
- Permite que o professor visualize as versões do algoritmo feito pelo aluno;
- Visualize os resultados dos diagnósticos SIATP;
- Envie comentários, sugestões;
- Corrija o algoritmo entregue e informe se o mesmo atende como solução ao problema proposto, em caso negativo, informe o motivo do algoritmo estar incorreto;

A Ilustração 30 ilustra o componente e suas funcionalidade descritas.



**Ilustração 30: Interface de Acompanhamento de Aprendizado**

### 3.2.3 PERFIL



**Ilustração 31: Protótipo de Interface - Perfil de Usuário**

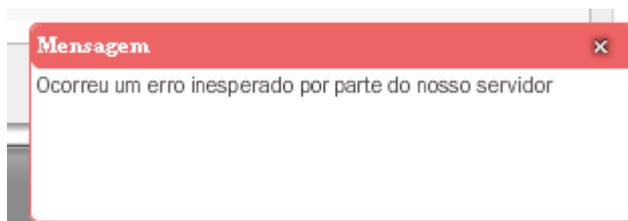
O componente de perfil de usuário deve possibilitar a mudança nos dados cadastrais e permitir o upload de foto do usuário. A Ilustração 31 demonstra as funcionalidades na forma de um cartão de identificação.

### 3.2.4 COMPONENTES AUXILIARES

Os componentes auxiliares contemplam o grupo de componentes que não adicionam funcionalidades referentes ao negócio do ambiente mas sim, interagem com os demais a fim de atender as seguintes funcionalidades :

- Informar ao usuário que sua tarefa está sendo processada;
- Informar ao usuário sobre o andamento do processo de carga dos componentes;
- Alertar ao usuário sobre possíveis erros, avisos que merecem atenção e avisos de sistema.;

Quanto aos tipos de avisos buscou-se utilizar de cores para identificação do tipo de mensagem. Utilizou-se a cor vermelha em mensagens de erro para despertar a atenção do usuário para o erro ocorrido. A Ilustração 32 ilustra um exemplo de mensagem desse tipo.



**Ilustração 32: Protótipo de Interface - Modelo de Mensagem de Erro (Vermelho)**

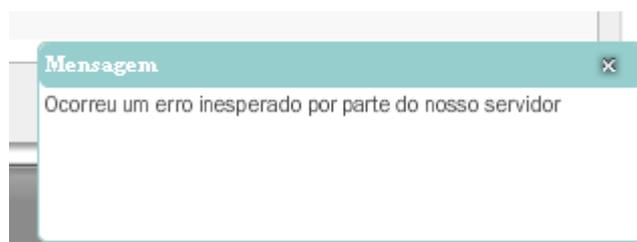
Nos casos onde exista a necessidade de apenas alertar o usuário de um problema leve, ou seja, que não compromete o funcionamento da aplicação e que pode ser ajustado em outro momento. Para representar visualmente esse tipo de ação se utilizou da cor laranja. A Ilustração 33 exemplifica essa situação.



**Ilustração 33: Protótipo de Interface - Modelo de Mensagem de Atenção (Laranja)**

Para os avisos globais, ou seja, aqueles utilizados em grande quantidade com o intuito de resposta a uma ação realizada por parte do utilizador. A cor azul foi utilizada para representar esse

tipo de mensagem. A Ilustração 34 demonstra uma mensagem desse tipo.



**Ilustração 34: Protótipo de Interface - Modelo de Mensagem de Informação (Azul)**

As cores utilizadas nesses componentes foram baseadas no recurso da usabilidade conhecido como uso da cores.

### **3.3 PADRÕES DE ARQUITETURA**

Para o desenvolvimento do ambiente proposto se faz necessário definir padrões referentes a arquitetura do sistema. A presente sessão apresenta as tecnologias utilizadas para o desenvolvimento assim como as ferramentas necessárias.

#### **3.3.1 PLATAFORMA ADOBE FLEX**

A tecnologia Flex é uma estrutura de código aberto para o desenvolvimento de aplicações ricas voltadas a internet. Sua estrutura é composta por duas linguagens MXML, uma linguagem declarativa baseada em XML, utilizada principalmente na construção do layout e modelar o comportamento de interface e, ActionScript 3, uma linguagem de programação avançada responsável por controlar os recursos de interface. É composto por uma biblioteca de componentes que auxiliam o desenvolvimento. As aplicações desenvolvidas utilizando essa tecnologia rodam na plataforma Adobe Flash Player isso permite que as mesmas rodem sobre qualquer navegador que suporte a plataforma.

Porém, essa tecnologia foi desenvolvida apenas para desenvolvimento de interfaces com usuário, por si só ela não consegue conectar a uma base de dados, ou realizar regras de negócio de alto nível. Para isso, a Adobe oferece um recurso *open-source* para o acesso remoto e *web messaging* permitindo assim, que o aplicativo Flex possa se integrar com diversas outras

linguagens como Java, PHP, .Net, entre outras.

Em um modelo Cliente-servidor ou multi-camada, aplicações Flex servem como a camada de aplicação. Ao contrário de aplicações HTML que se baseiam-se em páginas, o Flex mantém o estado do cliente onde mudanças significativas de visualização não necessariamente requerem uma carga de uma nova página.

### **3.3.2 PLATAFORMA JAVA**

Java é uma linguagem de programação desenvolvida pela empresa Sun Microsystems. Consiste numa linguagem totalmente orientada a objetos. Sua compilação resulta numa representação intermediária chamada *bytecode* o qual é interpretado pela máquina virtual Java. Isso permite que os sistemas desenvolvidos sobre esse plataforma possam rodar em qualquer ambiente que possua a JVM instalada.

### **3.3.3 PROTOCOLO RTMP**

O RTMP (*Real Time Messaging Protocol*) é um protocolo proprietário desenvolvido pela Adobe Systems que utiliza, por padrão, a porta 1935 e foi utilizado primeiramente no *Flash Communication Server*.

RTMP usa o protocolo TCP/IP para a transmissão de pacotes e não é usado para RCP (*Remote Procedure Calls*). O RTMP mantém uma conexão persistente com o servidor e permite a comunicação em tempo real de dados do tipo: áudio, vídeo e objeto. Para ser realizada a comunicação RTMP, é necessário, além do servidor, um arquivo swf. Este arquivo que deverá ser compartilhado pelos usuários finais, deve ser dotado de chamadas RTMP, normalmente codificadas em *Action Script*.

### **3.3.4 SERVIDOR DE APLICAÇÃO JBOSS**

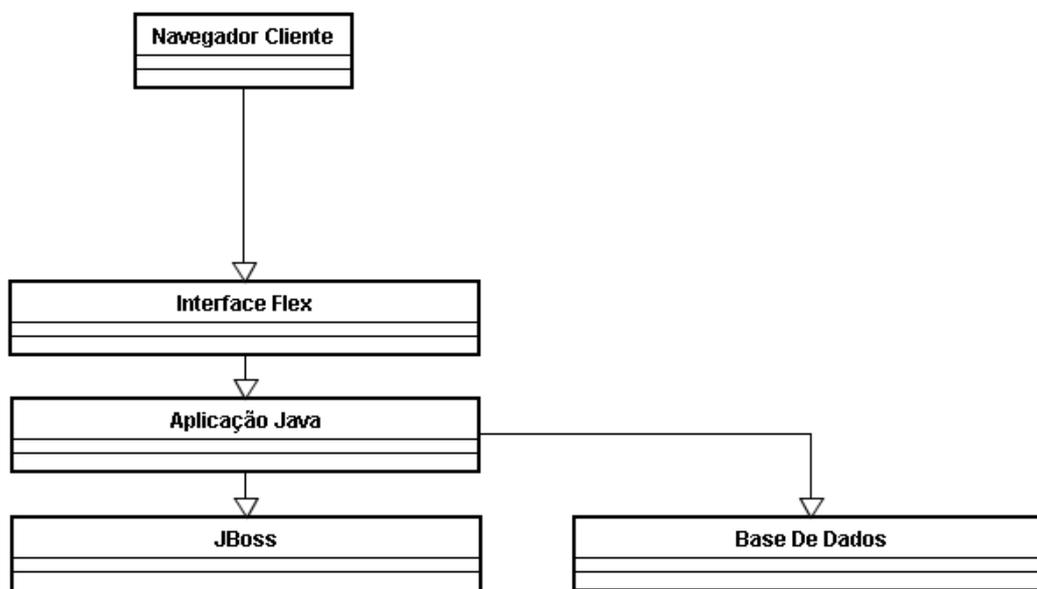
O servidor de aplicação Jboss consiste num software o qual oferece um ambiente para a instalação e execução de aplicações. Dá suporte aos padrões J2EE e possui código fonte aberto.

Desenvolvido em Java o que possibilita ser utilizado sobre qualquer sistema operacional com suporte a Java. Permite o desenvolvimento de aplicações com foco na solução de problemas de negócio.

### 3.3.5 APLICAÇÃO J2EE

A plataforma Java 2 Enterprise Edition, é considerado um padrão de desenvolvimento de aplicações Java multi-camadas para web. Essa plataforma define padrões para os componentes de negócio e de interação com a web. Por possuir diversos padrões e conseguir separar de forma organizada as camadas de negócio, camada de interface e o acesso aos dados é muito utilizado em aplicações corporativas.

A Ilustração 35 demonstra como as camadas do sistema se comunicam. Primeiramente se dá através de uma requisição feita pelo navegador cliente. O navegador cliente interage com a interface Flex e através das ações do usuário, a interface Flex se comunica com as funções referentes ao negócio contidas na camada de aplicação java. A camada de aplicação java processa as informações, realiza regras específicas, acessa a base de dados caso necessite e retorna o resultado para a interface. Toda essa estrutura é executada no ambiente Jboss o qual fornece diversas ferramentas que possibilitam a realização de toda essa estrutura.



**Ilustração 35: Modelo de Estrutural do Sistema**

### 3.3.6 EMPACOTAMENTO

O sistema todo será armazenado conforme a seguinte estrutura:

Pacote EAR: Esse pacote contempla a aplicação como um todo. Para a instalação do sistema apenas será necessário um servidor que implemente as especificações J2EE e colocar esse arquivo na pasta correspondente às aplicações do servidor desejado. Essa estrutura é composta por arquivos do tipo:

JAR- os quais são responsáveis por conter a estrutura com a lógica do sistema.

WAR – responsável por armazenar a camada web, ou seja, as interfaces com o usuário e os componentes de integração da mesma com os componentes da lógica de negócio.

### 3.3.7 ARQUITETURA

O ambiente em questão foi desenvolvido utilizando as linguagens Java e AS3 baseada no *framework* Flex. A tecnologia Flex será aplicada no desenvolvimento da interface enquanto que a tecnologia Java será responsável por manter a lógica de negócio do sistema e se integrar com a interface.

A escolha pelas linguagens se baseou nas seguintes vantagens:

Portabilidade: a aplicação desenvolvida pode ser implantada em diversas plataformas sem a necessidade de se modificar ou reestruturar o sistema.

Orientação a Objetos: esse paradigma de programação permite uma simplificação do desenvolvimento de sistemas complexos.

Flexibilidade: ambas linguagens já se encontram bem difundidas existindo diversas comunidades de desenvolvedores, ampla documentação e diversas bibliotecas prontas que podem ser utilizadas.

### 3.3.8 REQUISITOS DE SISTEMA

Para que seja possível a utilização do ambiente, existem requisitos que devem ser atendidos tanto por parte do cliente quando por parte do servidor. São eles:

Cliente:

A interface poderá ser executada em qualquer navegador de internet que possua o Adobe Flash Player 10 instalado.

O mínimo de memória necessária para utilização é 512Mb de RAM.

Recomenda-se preferencialmente o uso de monitores que suportem resolução igual ou superior a 1024 x 768.

Servidor:

Ter o Java JRE instalado.

Possuir o servidor de aplicação JBOSS 5.1.0.

Banco de dados Firebird.

O EAR correspondente ao ambiente

Ter no mínimo 1Gb de memória RAM

### **3.3.9 FERRAMENTAS DE DESENVOLVIMENTO**

Para o desenvolvimento e manutenção do ambiente foram utilizadas as seguintes ferramentas:

- Java Development Kit (JDK) 1.6
- IDE Eclipse 3.4 J2EE
- Adobe Flex Builder 3 ( Licenciado para estudante)
- JBoss 5.1.0 GA
- Repositório CVS

### **3.3.10 PADRÕES DE PROJETO**

MVC (Model-View-Controller) é um padrão de projeto que será utilizado para o desenvolvimento da interface que consiste em separar os componentes de tela em 3 partes distintas:

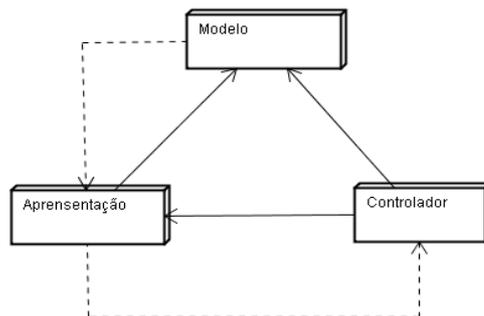
Model: O Model (Modelo) representa os dados do programa.

View: A View(Apresentação) é responsável por apresentar os dados contidos no Model ao

usuário e captar as ações realizadas pelo usuário na interface.

Controller: O Controller (Controlador) é o objeto que responde as ações executadas pelo usuário, é responsável pelas mudanças relacionadas aos dados do Model e por controlar as views a serem exibidas.

A separação da aplicação nessas camadas permite uma melhor separação das responsabilidades sobre os componentes de interface e ajuda na manutenção do sistema. A Ilustração 36 demonstra a comunicação entre os componentes definidos pelo padrão MVC.



**Ilustração 36: Representação do Padrão MVC**

### 3.3.11 MODELAGEM DO PROJETO

Para o desenvolvimento do projeto se optou por utilizar a língua inglesa como padrão de nomenclatura dessa forma, toda e qualquer classe, variável, método e tabela se utiliza de um nome representativo na língua inglesa. Assim como o próprio ambiente, que para o usuário será conhecido por Caderno de Exercícios, internamente será conhecido como *StudentBook*. Adotou-se esse padrão para evitar a utilização de acentos, cedilhas, tremas e hifens e, unificar a leitura visto que as linguagens de programação utilizam a língua inglesa.

Com o intuito de uma melhor organização e agilidade no desenvolvimento a estrutura dos fontes que geram o ambiente ficou dividida em diversos projetos onde cada um busca suprir uma funcionalidade específica. São eles:

- StudentBookAttachment: Projeto responsável por armazenar os anexos realizados via upload pelo professor e as imagens de cada usuário.
- StudentBookEAR: Projeto responsável por montar o pacote EAR contendo a aplicação. Cabe a esse projeto linkar os demais e configurá-los para que seja possível a execução

seguindo os padrões J2EE.

- StudentBookEJB: Projeto responsável por armazenar todas as classes responsáveis por manter as regras de negócio do ambiente.
- StudentBookEJBClient: Projeto responsável por fornecer as classes acessadas pela camada web (interface gráfica), responsável por invocar as classes de negócio.
- StudentBookSIATP: Projeto responsável por manter o sistema SIATP.
- StudentBookWeb: Projeto responsável pela interface gráfica do ambiente.

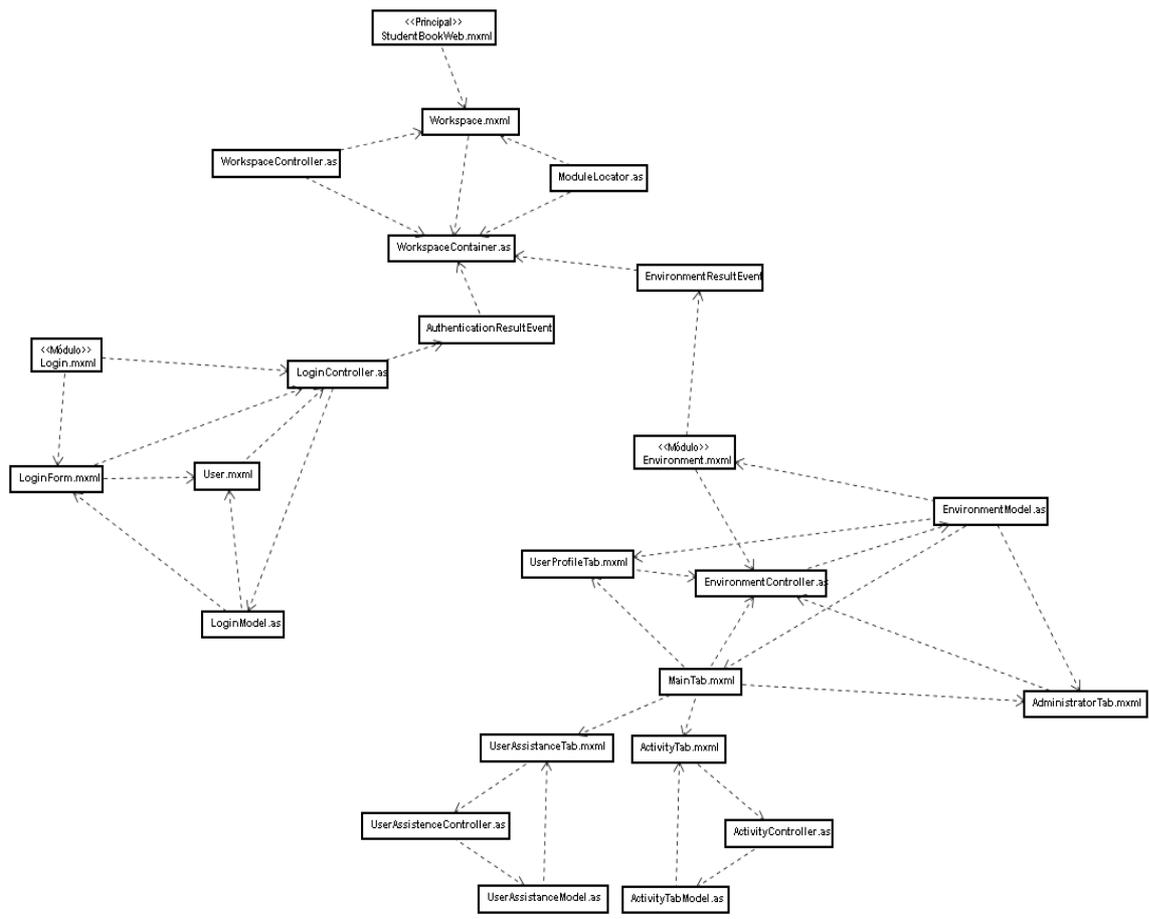
### **3.3.12 ESTRUTURA DO STUDENTBOOKWEB**

O projeto StudentBookWeb é responsável pela interface gráfica e por controlar todas as interações com o usuário. Para se chegar a interface proposta se fez necessária a construção de diversos artefatos. Os artefatos e as suas interligações estão representados na Ilustração 37.

Compõem esses artefatos as telas ou views, controladores ou controllers e os modelos de dados ou models, denominação oriunda do padrão MVC. Dentre todos os artefatos merecem destaque o WorkspaceContainer o qual é responsável por alternar entre a tela de login e a tela do ambiente.

As classes Login, LoginModel e LoginController destacam-se por controlarem a autenticação do usuário, ou seja a classe Login exibe a tela ao usuário, a classe LoginModel armazena as informações inseridas pelo usuário e a LoginController fica responsável por enviar esses dados para a camada de negócio.

As classes Environment, EnvironmentModel, EnvironmentController são responsáveis por controlar o ambiente propriamente dito. A classe Environment é responsável por agregar os subcomponentes na tela. A EnvironmentModel se encarrega de armazenar os dados utilizados pela Environment e seus subcomponentes enquanto que o EnvironmentController fica responsável por invocar os métodos da camada de negócio e por controlar as ações realizadas na interface.



**Ilustração 37: Modelo Estrutura da Interface**

**3.3.13 ESTRUTURA DO STUDENTBOOKCLIENT**

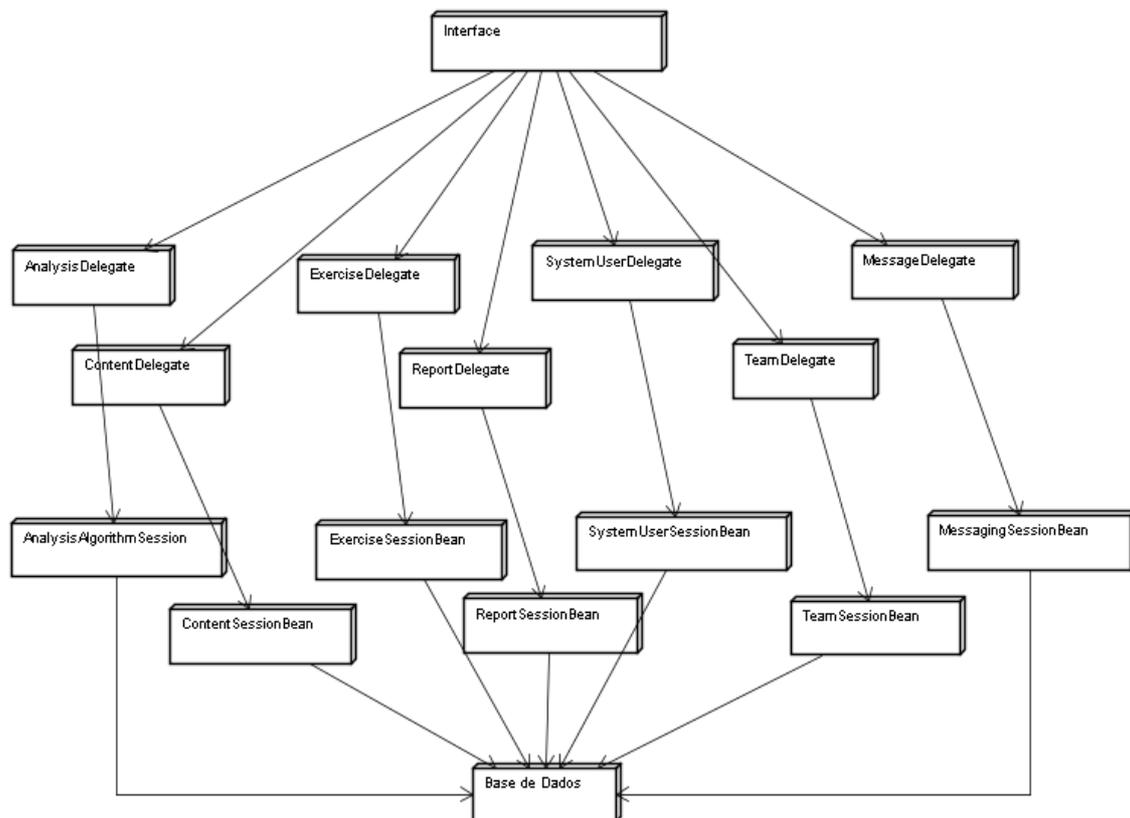
Esse projeto é responsável por receber as requisições realizadas pela interface e delegar a ação para o componente de negócio responsável por aquele tipo de ação. Em outras palavras, a interface se comunica com o *delegate* correspondente a ação que deseja realizar e o mesmo dispara essa solicitação para o componente de negócio correspondente. Tendo isso em mente construiu-se as seguintes classes delegadoras: AnalysisDelegate, ContentDelegate, ExerciseDelegate, MessageDelegate, ReportDelegate, SystemUserDelegate, TeamDelegate.

### 3.3.14 ESTRUTURA DO STUDENTBOOKEJB

Esse projeto é responsável por manter as lógicas de negócio do ambiente. Todas as classes construídas e que estão presentes nesse projeto realizam o acesso à base de dados, manipulam os dados da base de acordo com a necessidade e retorna o resultado. Cada uma dessas classes é responsável pelas funcionalidades referentes ao conceito que representa no ambiente.

As classes que compõem esse projeto são: *AnalysisAlgorithmSession*, *ContentSessionBean*, *ExerciseSessionBean*, *MessagingSessionBean*, *ReportSessionBean*, *SystemUserSessionBean*, *TeamSessionBean*. A classe *AnalysisAlgorithmSession* é responsável por manter a análise proporcionada pelo ambiente assim como as buscas nas tabelas correspondentes, cabe a ela também integrar o ambiente ao sistema SIATP. A classe *ContentSessionBean* têm a responsabilidade de controlar as buscas, adições, edições e remoções dos conteúdos disciplinares. *ExerciseSessionBean* é a classe responsável pela manutenção dos exercícios. A classe *MessagingSessionBean* controla as mensagens trocadas entre os participantes via mural de recados. A *ReportSessionBean* se encarrega de persistir os relatórios de sugestão/erro. As classes *SystemUserSessionBean*, *TeamSessionBean* possuem respectivamente a responsabilidade de autenticar, cadastrar, editar e buscar os usuários e, criar e buscar as turmas.

De uma forma geral, a interface se comunica com as classes *delegate* que por sua vez delegam a tarefa para essas classes de negócio. Utilizando esse tipo de organização pode-se perceber a existência de duas camadas. A camada de apresentação onde se enquadram os projetos *StudentBookWeb* e o *StudentBookClient*, o primeiro por construir a interface visual e o segundo por estar fortemente ligado a atender as solicitações oriundas da interface. E, a camada de negócio a qual contempla o projeto *StudentBookEJB*. A Ilustração 38 ilustra a comunicação entre as classes citadas.



**Ilustração 38: Modelo de Comunicação**

### 3.3.15 ESTRUTURA DO STUDENTBOOKENTITY

O projeto StudentBookEntity mantém as entidades modeladas para a aplicação. Estão contidos nesse projeto as classes utilizadas pelo ambiente que representam Usuário, Turma, Exercício, Pasta, Arquivo, Solução, Mensagem, Análise do Professor e a Análise do SIATP. Por se utilizar a língua inglesa como padrão para nomenclatura considera-se respectivamente as classes *SystemUser*, *Team*, *Exercise*, *ContentFolder*, *Content*, *Solution*, *Message*, *TeacherAnalysisHistory* e *SIATPAnalysisHistory* como sendo as classes correspondentes as anteriormente sitadas.

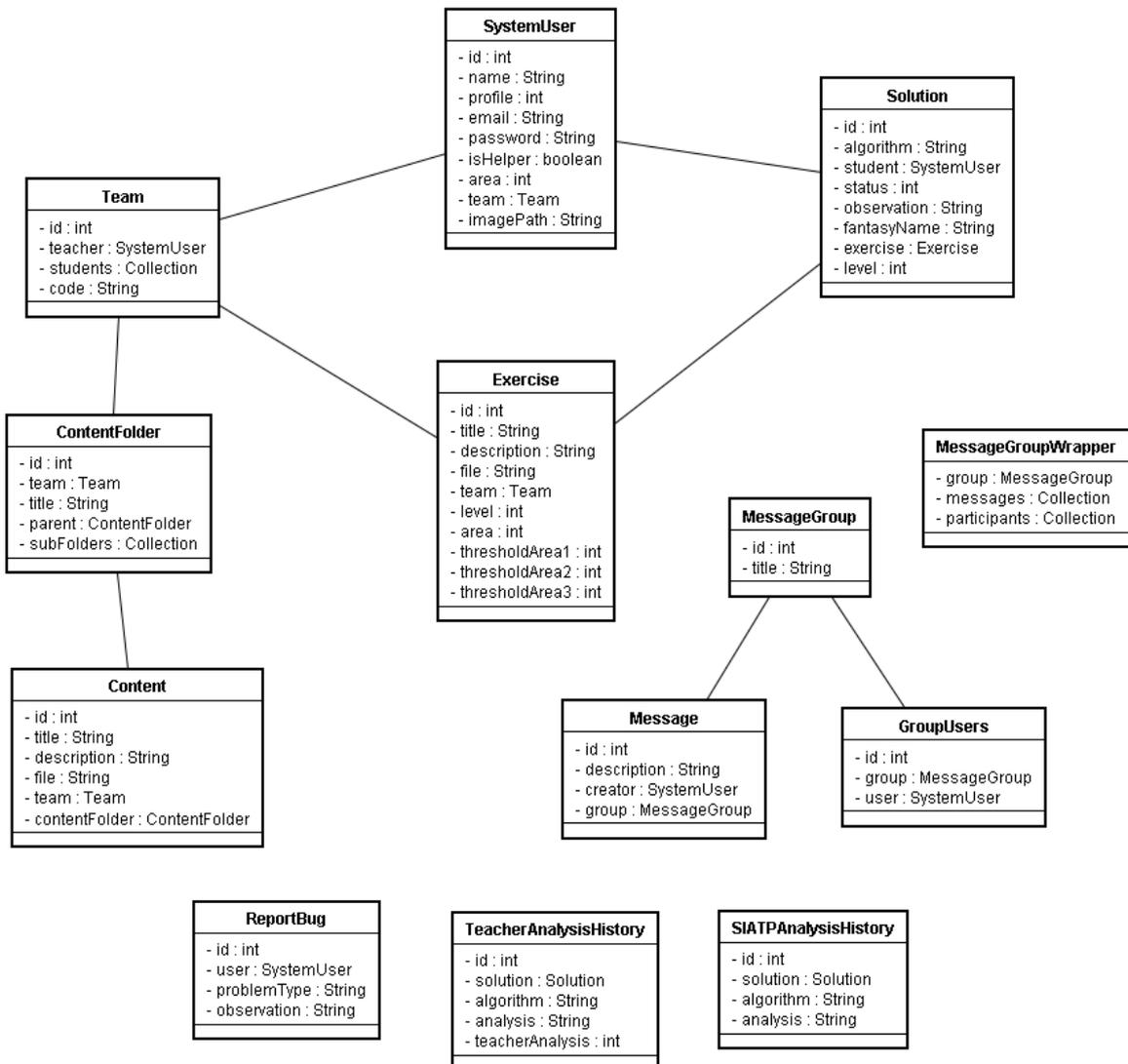
A classe *SystemUser* representa um usuário perante ao ambiente, o atributo *name* representa o nome do usuário, o atributo *profile* representa o tipo de papel que esse usuário representa. Considera-se *profiler* igual a 1 para representar alunos, profiler igual a 2 para professor e *profiler* igual a 3 para administrador. Os atributos *email*, *password* e *imagePath* correspondem respectivamente ao email, a senha e o nome da imagem usada pelo usuário. O atributo *isHelper* indica se o usuário aceita ajudar outros participantes. E, *area* informa a área do conhecimento que esse aluno se encontra.

A classe *Team* é utilizada para representar uma turma. Essa classe possui um atributo *code* que indica o código da turma, um atributo *teacher* que informa o professor responsável pela turma e, a lista de *students* que contém todos os alunos cadastrados nessa turma.

A *Exercise* é a classe que representa um exercício. Essa classe possui os seguintes atributos: *title*, *description*, *file*, *team*, *level*, *area*, *thresholdArea1*, *thresholdArea2*, *thresholdArea3*. O atributo *title* representa o título dado ao exercício. O atributo *description* representa a descrição do exercício. *File* é o atributo responsável por armazenar o nome do arquivo anexado no cadastro de exercício. O atributo *team* corresponde a turma o qual esse exercício faz parte. *Level* corresponde ao nível de dificuldade do exercício. O campo *area* representa a área do conhecimento o qual esse exercício está enquadrado. Os atributos *thresholdArea1*, *thresholdArea2*, *thresholdArea3* correspondem aos limiares de erro de cada uma das áreas do conhecimento.

A classe *Solution* representa uma solução. A solução para o ambiente nada mais é que uma possível resposta para o exercício. Os atributos existentes nessa classe são: *algorithm*, responsável por conter o algoritmo desenvolvido; *student*, corresponde ao usuário aluno criador da solução; *status*, esse campo pode assumir os valores zero(0) quando está sendo editado e utilizado pelo aluno e um (1) quando encontra-se em análise pelo professor; os campos *observation*, *fantasyName*, *level* e *exercise* representam respectivamente as anotações realizadas pelo aluno, o nome dado ao arquivo de solução pelo aluno, o grau de dificuldade atribuído pelo aluno ao exercício e o exercício o qual essa solução corresponde.

A Ilustração 39 demonstra o diagrama de classes das entidades apresentadas assim como os seus relacionamentos.



**Ilustração 39: Diagrama de Classes do Ambiente**

### 3.3.16 STUDENTBOOKEAR

O projeto StudentBookEAR se encarrega de montar a estrutura de arquivos e pastas necessários para o pacote EAR com base nas classes dos outros projetos. Nesse projeto encontram-se também os arquivos de configuração para que o ambiente possa ser instalado. Os arquivos de configuração contidos nesse projeto são:

- Properties-Services.xml, responsável por gerenciar as variáveis de configuração do ambiente. O ambiente se utiliza das seguintes variáveis para o seu funcionamento:

“studentbook.attachment.repository” e “studentbook.server.address”. A primeira é responsável por apontar no sistema de arquivos do sistema operacional utilizado o local onde se encontra a pasta StudentBookAttachment.war dentro do servidor de aplicação JBoss. Um exemplo de valor para essa variável seria “C:\\jboss-5.1.0.GA\\server\\default\\deploy\\StudentBookEAR.ear\\StudentBookAttachment.war\\”. A segunda, refere-se ao ip da máquina servidora na rede em que está inserida.

- Firebird-ds.xml, responsável por configurar a conexão com a base de dados. Nesse arquivo se informa o local onde se encontra a base de dados utilizada.

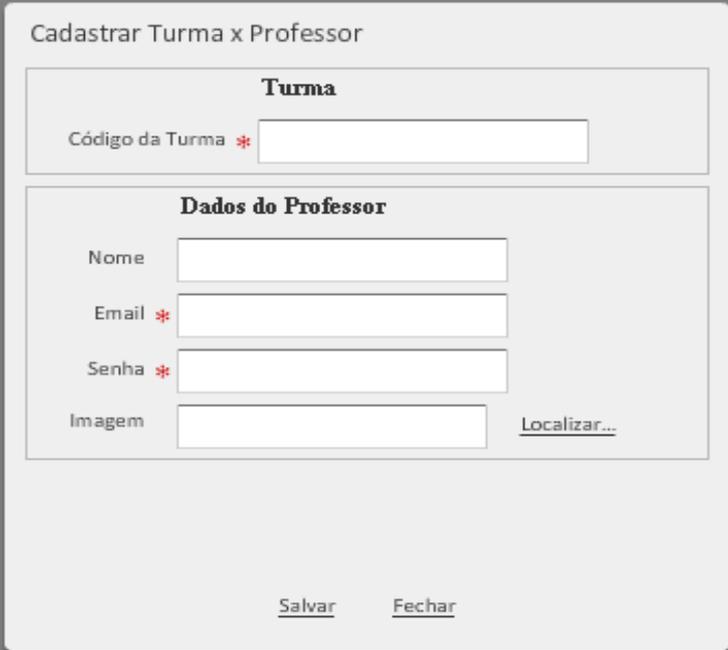
## 4 CENÁRIO DE UTILIZAÇÃO DO AMBIENTE

Nesse capítulo apresenta-se um cenário de utilização do Ambiente que descreve um conjunto ordenado de interações para exemplificar a sua utilização abrangendo todas as funcionalidades implementadas. Para esse cenário considerou-se a existência de um usuário do tipo administrador previamente cadastrado na base de dados.

### Cadastro de Turma e Professor

O ambiente só funcionará se existir ao menos uma turma e um professor assim sendo, se dá início com o cadastro de turma e professor. Apenas usuários com o papel administrador podem executar esse tipo de ação. Quando logado, o ambiente apresenta um menu com apenas a opção Cadastrar Turma disponível.

Acessando a opção do menu disponível Cadastrar Turma, o ambiente apresenta a seguinte tela para inclusão demonstrada pela Ilustração 40.



A tela de cadastro, intitulada "Cadastrar Turma x Professor", é dividida em duas seções principais. A primeira seção, "Turma", contém um campo de texto rotulado "Código da Turma" com um asterisco vermelho obrigatório. A segunda seção, "Dados do Professor", contém quatro campos de texto: "Nome", "Email" (com asterisco vermelho obrigatório), "Senha" (com asterisco vermelho obrigatório) e "Imagem". Ao lado do campo "Imagem" há um botão "Localizar...". Na base da tela, há dois botões: "Salvar" e "Fechar".

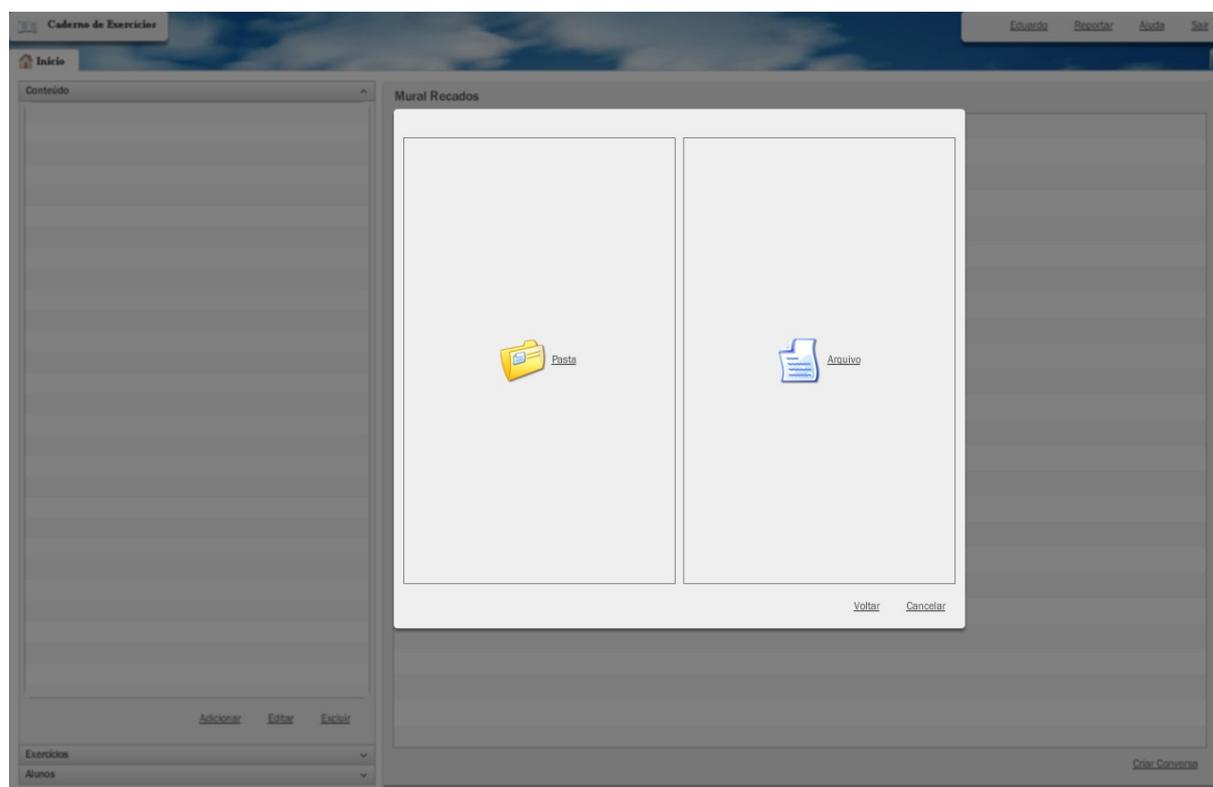
**Ilustração 40: Tela de Cadastro de Turma x Professor**

Para o ambiente, não pode existir uma turma sem um professor associado. O administrador informa o nome da turma e preenche o cadastro referente ao professor que será vinculado a essa turma. Ao teclar *salvar* o ambiente cadastra as informações e avisa o usuário.

A partir desse momento, tem-se uma turma e um novo professor disponível para o ambiente.

### **Conteúdo Didático**

Autenticando-se com o usuário professor criado anteriormente, tem-se livre acesso para adicionar conteúdo disciplinar, enviar mensagens para seus alunos, cadastrar os exercícios e acompanhar o aprendizado do aluno. Ao tentar adicionar um conteúdo disciplinar é solicitado o tipo de inclusão que o professor deseja realizar. Adicionar Pasta ou Arquivo são as duas opções disponíveis. A pasta nada mais é que uma forma do professor organizar o seu conteúdo separando como ele bem entender. O arquivo é o conteúdo o qual o professor deseja adicionar ao sistema, o mesmo pode ser em forma de arquivo PDF ou até mesmo texto puro.



**Ilustração 41: Tela de Adição de Conteúdo**

### **Exercícios**

Para que o ambiente tenha sentido no aprendizado se faz necessário que o professor cadastre alguns exercícios para que os alunos possam resolver e aperfeiçoar o seu aprendizado. Utilizando o menu da lateral esquerda o professor pode facilmente acessar o tópico de exercícios. Ao solicitar a inclusão de um novo exercício o ambiente lhe proporciona a seguinte interface ilustrada na

Ilustração 42 que segue.

The image shows a web form titled "Adicionar Exercício". It contains the following elements:

- Título:** A single-line text input field.
- Descrição:** A large multi-line text area.
- Anexo:** A text input field followed by a "Localizar..." button.
- Dificuldade:** A numeric input field with up/down arrows, currently showing "1".
- Área do Conhecimento:** A numeric input field with up/down arrows, currently showing "1".
- Limiar Área 1:** A numeric input field with up/down arrows, currently showing "0".
- Limiar Área 2:** A numeric input field with up/down arrows, currently showing "0".
- Limiar Área 3:** A numeric input field with up/down arrows, currently showing "0".
- Buttons:** "Salvar" and "Cancelar" buttons at the bottom center.

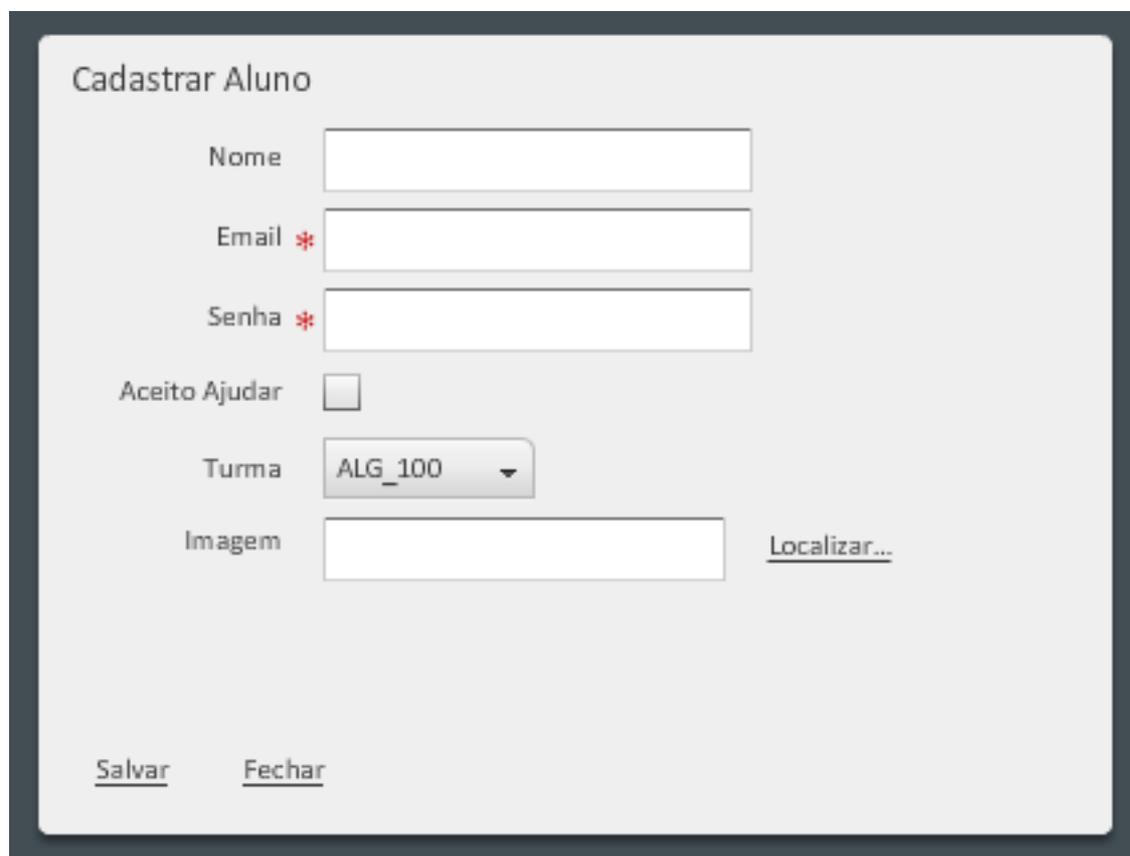
**Ilustração 42: Tela de Cadastro de Exercícios**

A inclusão de um exercícios se dá mediante a inserção de um título, uma descrição do mesmo contendo as informações necessárias para que o aluno consiga compreender o que deve ser feito. Pode-se ainda, para auxiliar na explicação ao aluno adicionar um anexo o qual entende-se por um pdf ou uma imagem. O campo “dificuldade” serve para informar o grau de dificuldade do exercício. O campo “área do conhecimento” identifica em qual das 3 áreas do estudo de algoritmos esse exercício se enquadra. Os campos “limiar área 1”, “limiar área 2”, “limiar área 3” servem para configurar o sistema SIA-TP. O limiar nada mais é do que o limite de erros que o aluno pode gerar naquela área específica quando estiver desenvolvendo sua solução.

### **Cadastro de Aluno**

Nesse momento, contendo uma turma e exercícios o ambiente está apto a ser utilizado pelos

alunos. O cadastro de aluno é livre. Qualquer pessoa que deseja ingressar ao ambiente como aprendiz pode se cadastrar através do botão presente na interface de autenticação. Quando solicitado o cadastro de um novo usuário o ambiente apresenta a seguinte tela de cadastro.



A screenshot of a web form titled "Cadastrar Aluno". The form contains the following fields and controls:

- Nome:** A text input field.
- Email \*:** A text input field with a red asterisk indicating it is required.
- Senha \*:** A text input field with a red asterisk indicating it is required.
- Aceito Ajudar:** A checkbox.
- Turma:** A dropdown menu with "ALG\_100" selected.
- Imagem:** A text input field with a "Localizar..." link to its right.
- Buttons:** "Salvar" and "Fechar" buttons at the bottom left.

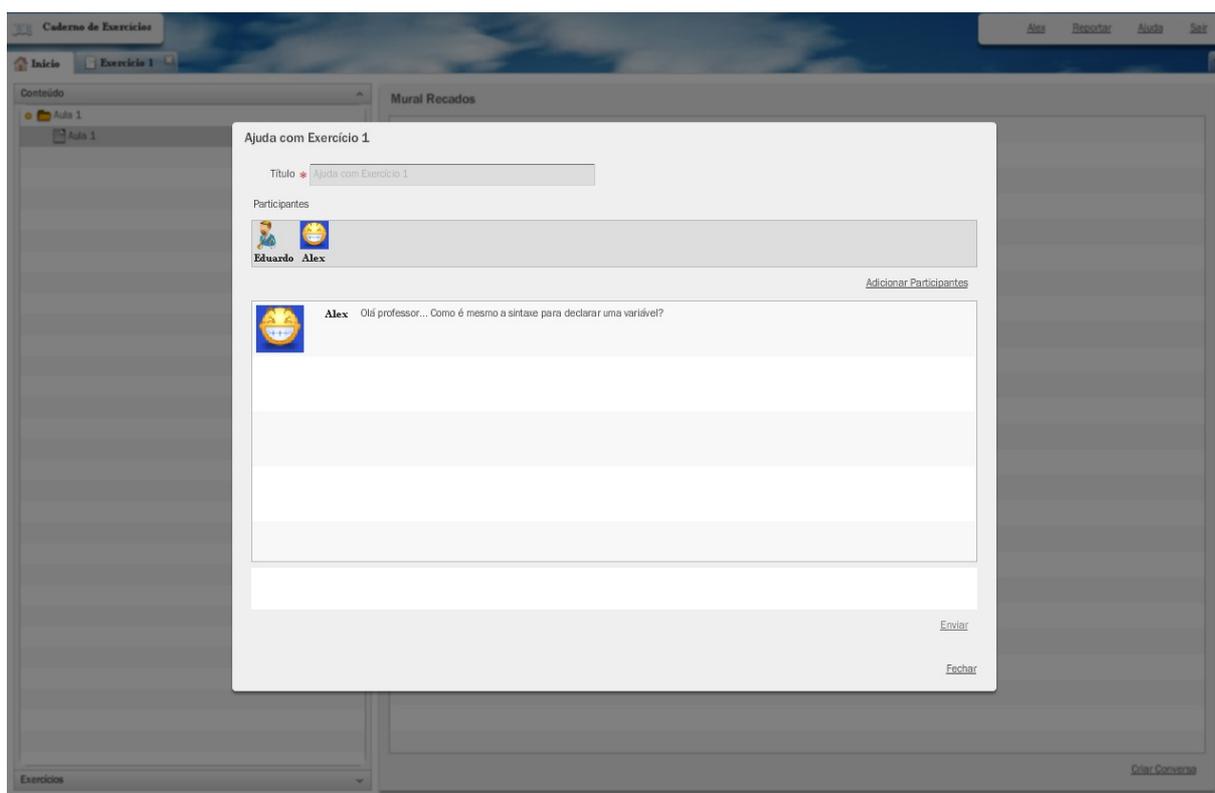
**Ilustração 43: Tela de Cadastro de Aluno**

Para o cadastro o aluno deve informar um email e uma senha, os quais serão utilizados como meio de autenticação, o seu nome, a turma a qual deseja ingressar e informar se ele aceita contribuir ajudando os demais alunos participantes, ou seja, se o aluno não desejar ajudar os demais o ambiente somente irá permitir que ele troque mensagens com o professor, caso contrário ele poderá se comunicar com o professor e os demais alunos.

### **Mensagens**

Em caso de dúvida ou necessidade de se comunicar com o professor ou até mesmo outros alunos participantes do ambiente o aluno pode enviar mensagens utilizando o sistema de recados. Desta forma, via botão "Criar Conversa" o aluno acessa a seguinte tela representada pela Ilustração 44. Informa obrigatoriamente um título para a mensagem, adiciona os participantes da conversa e envia a mensagem que deseja para o professor ou colegas. Todas as conversas as quais esse aluno

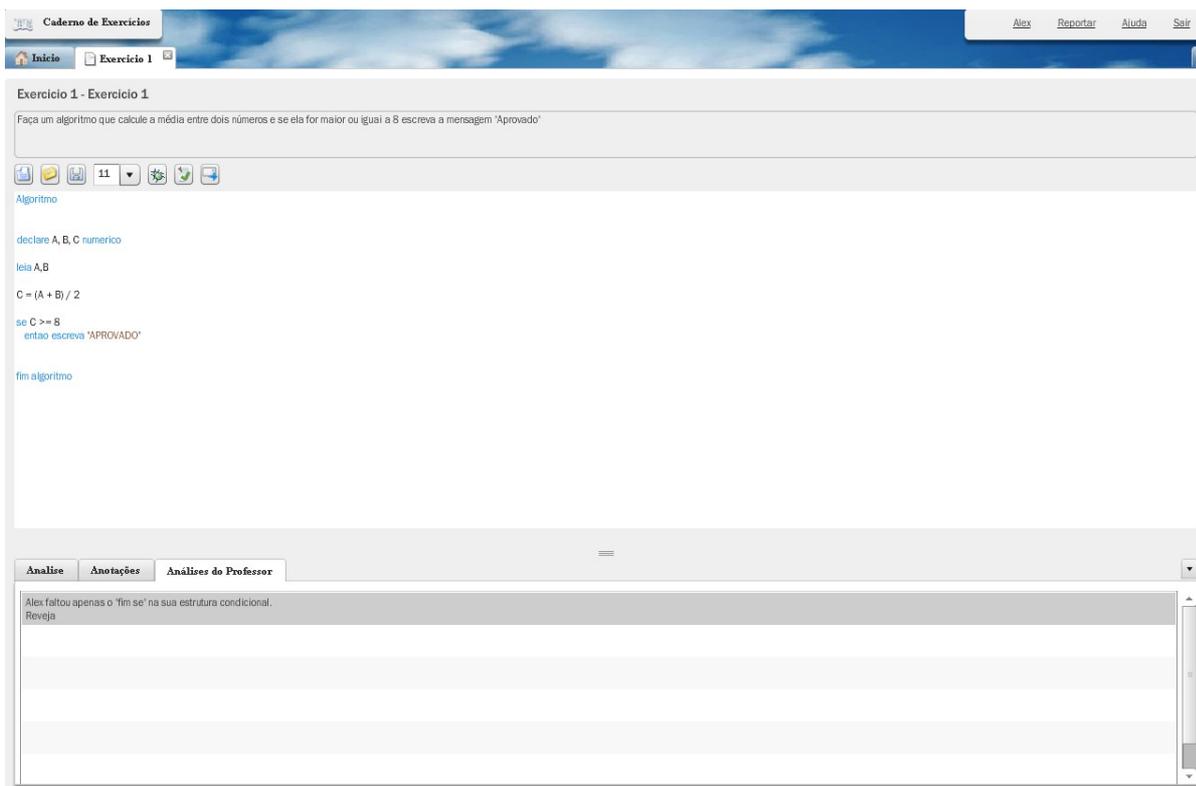
estiver participando serão exibidas na tela Mural de Recados.



**Ilustração 44: Tela de Criação de Mensagem**

### **Resolvendo Exercícios**

No submenu Exercícios contido na parte esquerda da tela o aluno pode visualizar os exercícios cadastrados pelo professor. Com um duplo clique do mouse o mesmo tem acesso a tela de resolução de exercícios. A interface apresenta a ordem do exercício, botões com as seguintes ações criar, abrir, salvar, enviar para análise, enviar para o professor e o editor para que se possa desenvolver sua solução. O aluno desenvolve a sua solução no editor e assim pode solicitar uma análise ao sistema SIATP. O resultado dessa análise é apresentado na aba Análises. A aba anotações pode ser utilizada pelo aluno para colocar anotações sobre o exercício que o auxilie na solução. A aba análise do professor contém o *feedback* do professor sobre o exercício quando enviado para que o mesmo analise. A Ilustração 45 demonstra a tela de resolução de exercício descrita.

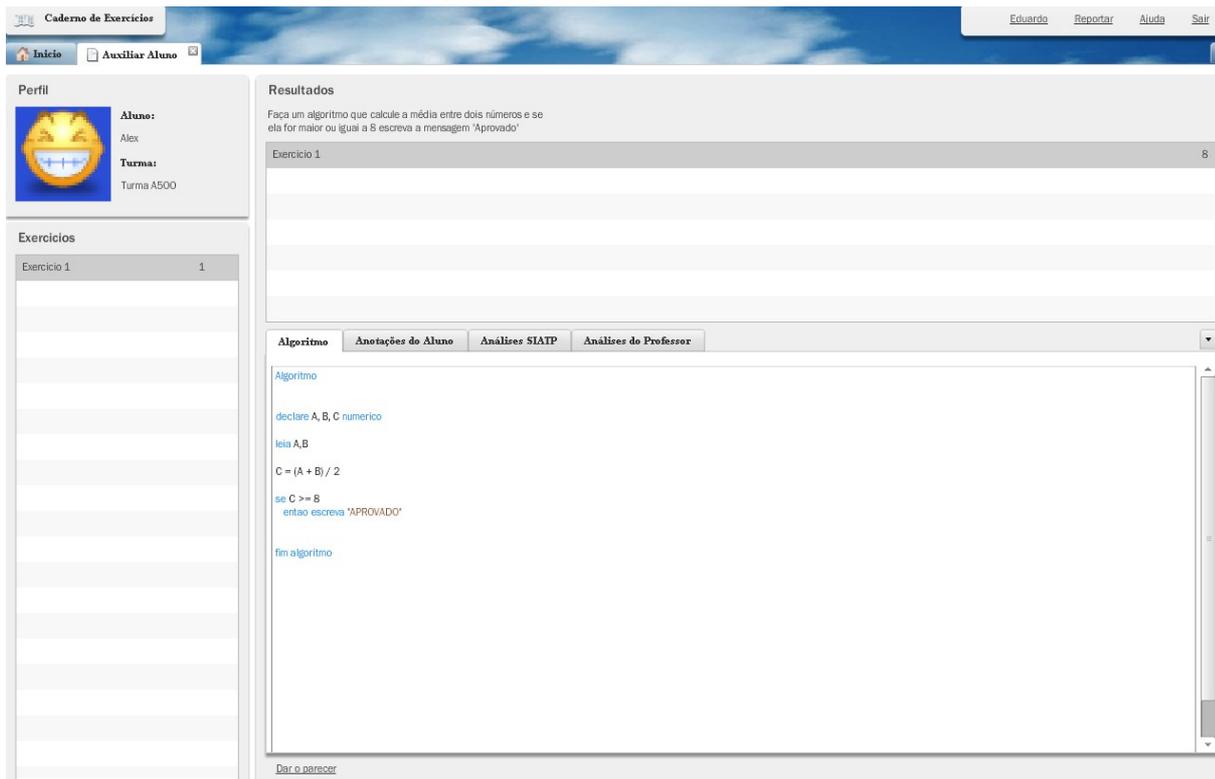


**Ilustração 45: Tela de Resolução de Exercício**

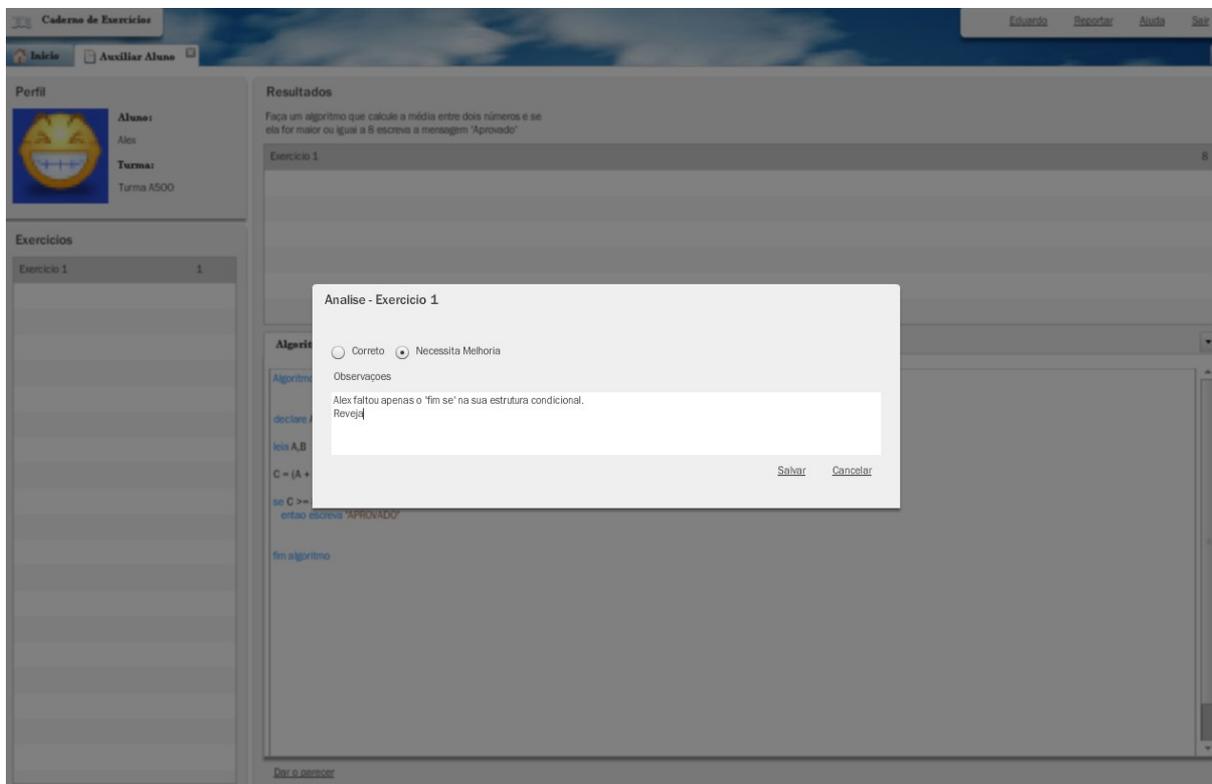
### **Acompanhar Aprendizado**

Um usuário professor quando autenticado no sistema possui acesso aos seus alunos via submenu contido na parte esquerda da tela. No menu alunos o professor visualiza todos os alunos participantes da turma pela qual ele é responsável. Com um duplo clique no nome do aluno o ambiente apresenta a tela de acompanhamento de aprendizado. Nessa tela o professor pode visualizar os exercícios propostos, quando selecionado um dos exercícios o ambiente apresenta os algoritmos pendentes de análise do professor. O professor ao selecionar o algoritmo pode visualizá-lo, ver seus pareceres anteriores, as análises que o aluno realizou utilizando o SIATP e as anotações do aluno. Com base nessas informações o professor pode dar o seu parecer quanto a solução desenvolvida pelo aluno para o exercício proposto. Após dar o parecer a solução volta para o aluno com o *feedback* do professor.

A Ilustração 46 e 47 demonstram respectivamente a interface do acompanhamento de aprendizado e a tela para atribuir o parecer.



**Ilustração 46: Tela de Acompanhamento de Aprendizado**



**Ilustração 47: Tela de Atribuição de Parecer**

## 5 TESTES E AVALIAÇÃO

Com o intuito de validar e avaliar o ambiente desenvolvido no dia vinte e cinco (25) de novembro do ano de 2009 na sala 400 do Bloco 71 da Universidade de Caxias do Sul realizou-se um teste alfa, teste de sistema em um ambiente controlado pelo desenvolvedor, do ambiente apresentado nesse trabalho. O teste contou com a participação de sete(7) alunos do curso de Licenciatura em Computação. Escolheu-se alunos de licenciatura por possuírem o conhecimento para realizar uma avaliação pedagógica e técnica. Não foram envolvidos alunos de disciplinas de algoritmos por se tratar de um momento inicial de testes muitos problemas poderiam ocorrer e, realizar uma avaliação de cunho pedagógico seria de grande interesse nesse momento a fim de detectar melhorias e ajustes de interface.

Foram avaliados além do ambiente descrito nesse trabalho a ferramenta para depuração de algoritmos desenvolvido no trabalho de Tiago Arrosi [ARROSI, 2009] que vem sendo construído paralelamente a esse trabalho a qual se agregará ao ambiente como uma nova funcionalidade.

O instrumento de avaliação utilizado foi um questionário desenvolvido por WEBBER,2009 o qual era composto de vinte e quatro questões onde cinco(5) buscavam avaliar aspectos pedagógicos e dezenove (19) aspectos técnicos. Ao final do instrumento reservou-se um espaço para comentários e sugestões de melhoria.

O teste foi iniciado com uma apresentação breve do ambiente demonstrando suas funcionalidades. Durante a apresentação os avaliadores interagiam com o ambiente livremente testando as funcionalidades que estavam sendo apresentadas e até mesmo testando funcionalidades ainda não explicadas. Percebeu-se nesse momento um aspecto positivo visto que os avaliadores conseguiam explorar funcionalidades ainda não explicadas a eles e utilizá-las. Ao final da demonstração foi entregue a cada avaliador o questionário de avaliação. Os resultados gerados por esse instrumento seguem representados nas Ilustrações 48 49, 50, 51 e 52. O questionário utilizado pode ser encontrado vide Anexo B.

<b>Avaliador</b>	<b>I – Classificação</b>	<b>II – Atividade Pedagógica</b>
Avaliador 1	Exercício e Prática Ambiente de Aprendizagem Interativo(micromundos) Ambiente de Aprendizagem Cooperativo	Construtivista
Avaliador 2	Exercício e Prática	Instrucionista/Comportamentalista
Avaliador 3	Exercício e Prática	Instrucionista/Comportamentalista
Avaliador 4	Ambiente de Aprendizagem Cooperativo	Construtivista
Avaliador 5	Jogo Educativo Exercício e Prática Tutorial Ambiente de Aprendizagem Interativo(micromundos)	Construtivista Sócio-interacionista
Avaliador 6	Sistema Tutor Inteligente Ambiente de Aprendizagem Interativo(micromundos) Ambiente de Aprendizagem Cooperativo	Construtivista
Avaliador 7	Exercício e Prática Tutorial	Sócio-interacionista

**Ilustração 48: Respostas do Questionário Referentes a Classificação**

<b>Avaliador</b>	<b>Questão 01</b>	<b>Questão 02</b>	<b>Questão 03</b>	<b>Questão 04</b>	<b>Questão 05</b>
Avaliador 1	Parcialmente	Sim	Parcialmente	Parcialmente	Não
Avaliador 2	Parcialmente	Não se aplica	Não	Não	Não
Avaliador 3	Parcialmente	Não se aplica	Não	Não	Não
Avaliador 4	Sim	Sim	Não	Não	Parcialmente
Avaliador 5	Sim	Sim	Sim	Sim	Sim
Avaliador 6	Parcialmente	Parcialmente	Não	Não	Não
Avaliador 7	Parcialmente	Parcialmente	Sim	Não	Sim

**Ilustração 49: Respostas das Questões referentes à Aspectos Pedagógicos**

<b>Avaliador</b>	<b>Questão 06</b>	<b>Questão 07</b>	<b>Questão 08</b>	<b>Questão 09</b>	<b>Questão 10</b>
Avaliador 1	Sim	Sim	Não se aplica	Sim	Não
Avaliador 2	Sim	Sim	Não se aplica	Parcialmente	Não
Avaliador 3	Sim	Sim	Não se aplica	Parcialmente	Não
Avaliador 4	Sim	Sim	Sim	Sim	Não
Avaliador 5	Sim	Parcialmente	Parcialmente	Parcialmente	Não
Avaliador 6	Parcialmente	Parcialmente	Sim	Sim	Não
Avaliador 7	Sim	Não	Não	Parcialmente	Não

<b>Avaliador</b>	<b>Questão 11</b>	<b>Questão 12</b>	<b>Questão 13</b>	<b>Questão 14</b>	<b>Questão 15</b>
Avaliador 1	Não se aplica	Parcialmente	Não se Aplica	Sim	Sim
Avaliador 2	Não se aplica	Sim	Não se Aplica	Não se aplica	Parcialmente
Avaliador 3	Não se aplica	Sim	Não se Aplica	Não se aplica	Parcialmente
Avaliador 4	Sim	Sim	Não se Aplica	Sim	Sim
Avaliador 5	Não se aplica	Parcialmente	Não se Aplica	Parcialmente	Parcialmente
Avaliador 6	Sim	Parcialmente	Parcialmente	Sim	Sim
Avaliador 7	Não se aplica	Parcialmente	Não se Aplica	Sim	Parcialmente

<b>Avaliador</b>	<b>Questão 16</b>	<b>Questão 17</b>	<b>Questão 18</b>	<b>Questão 19</b>	<b>Questão 20</b>
Avaliador 1	Sim	Não	Sim	Sim	Não
Avaliador 2	Sim	Não	Não se aplica	Não se aplica	Sim
Avaliador 3	Sim	Não	Não se aplica	Não se aplica	Sim
Avaliador 4	Sim	Parcialmente	Sim	Sim	Não se aplica
Avaliador 5	Sim	Parcialmente	Sim	Sim	Não
Avaliador 6	Sim	Não	Sim	Sim	Não
Avaliador 7	Não se Aplica	Não	Sim	Sim	Não

**Ilustração 50: Respostas das Questões referentes à Aspectos Técnicos**

<b>Avaliador</b>	<b>Questão 21</b>	<b>Questão 22</b>	<b>Questão 23</b>	<b>Questão 24</b>
Avaliador 1	Não	Não	Não	Sim
Avaliador 2	Não se aplica	Não se aplica	Não se aplica	Parcialmente
Avaliador 3	Não se aplica	Não se aplica	Não se aplica	Parcialmente
Avaliador 4	Não	Não se aplica	Não	Sim
Avaliador 5	Não	Não se aplica	Não	Parcialmente
Avaliador 6	Não	Não	Não	Parcialmente
Avaliador 7	Não se aplica	Não se aplica	Não	Não

**Ilustração 51: Respostas das Questões referentes à Aspectos Técnicos**

<b>Avaliador</b>	<b>Comentários / Sugestões</b>
Avaliador 1	Habilitar a tecla Enter para telas de Login e Entrada de Dados
Avaliador 2	
Avaliador 3	
Avaliador 4	
Avaliador 5	
Avaliador 6	
Avaliador 7	Exibir um menu de ajuda com a sintaxe utilizada pelo editor Possibilitar acessibilidade visual, exemplo aumentar fonte O layout do perfil está confuso. Poderia ser um menu suspenso ou uma página inteira.

**Ilustração 52: Respostas referente ao espaço para Comentários e Sugestões**

Durante o andamento dos testes, o Ambiente teve um resultado aceitável em termos de performance, ou seja, as pesquisas de busca de dados e a navegação tiveram um tempo de resposta de dois a cinco segundos. O único agravante foi a realização da análise utilizando o sistema SIATP, o qual obteve tempos de resposta variando de dez a trinta segundos. Ocorreram erros quando solicitada a análise no sentido de não retornar o resultado ao usuário e comprometer a performance do ambiente a cada execução. No geral, obteve-se resultados positivos referentes a interface e a proposta do trabalho como a facilidade de uso e ser efetivo, útil e produtivo. Diversos avaliadores comentaram que a ferramenta pode vir a ser um bom meio de apoio a educação a distância e como reforço as atividades realizadas em sala de aula.

Alguns avaliadores ressaltaram sobre a importância de desenvolver uma ferramenta de ajuda para que o aluno tenha conhecimento das ações que ele pode realizar no ambiente assim como a própria sintaxe utilizada pelo editor e pelo analisador SIATP. Disseram ainda que ficaram satisfeitos

com o aspecto visual apenas chamaram atenção para as tooltips, elemento de interface utilizado para fornecer algum tipo de explicação sobre o item o qual o cursor do mouse se encontra , que não estão com uma cor muito visível.

Após o término dos testes foi possível analisar os logs gerados no servidor de aplicação e com isso detectar as causas do problema entre a comunicação do ambiente com o sistema SIATP e assim realizar as devidas correções.

## **6 CONCLUSÃO**

### **6.1 SÍNTESE**

O presente trabalho buscou apresentar uma interface para um ambiente virtual de aprendizagem em Programação de Computadores a fim de proporcionar a disciplina uma ferramenta adicional que forneça um acompanhamento individualizado da evolução do aluno por parte do professor. O ambiente consiste num espaço interativo onde o aluno pode acessar os conteúdos didáticos e através da resolução de problemas elaborados pelo professor formar o seu conhecimento.

A interface desenvolvida busca integrar-se ao sistema SIA-TP com o intuito de identificar os pontos os quais o aluno deve reforçar os seus estudos, pois são pontos que podem não estar bem absorvido pelo mesmo.

Com a análise de alguns ambientes virtuais existentes e que são aplicados em diversas atividades desde específicas a genéricas foi possível eliciar os recursos e modelar suas características visuais. O estudo sobre usabilidade serviu como base para trabalhar com o design da aplicação. Buscou-se uma interface simples, evitando menus e funcionalidades de difícil acesso além de se transmitir um ar de seriedade e de comodidade ao utilizador.

Ressalta ainda, que em nenhum momento o sistema proposto buscou substituir qualquer ferramenta já existente que auxilie na aprendizagem ou ainda substituir um compilador. O trabalho procurou construir uma ferramenta de auxílio à aprendizagem que detecte os possíveis erros cometidos pelo aluno e alerte o professor sobre os possíveis pontos fracos do aluno. Proporcionando assim, um acompanhamento individualizado do aluno pelo professor.

### **6.2 CONTRIBUIÇÕES DO TRABALHO**

Esse trabalho propôs uma interface para um ambiente de aprendizagem de algoritmos o qual integra conceitos de ajuda colaborativa, análise e diagnóstico de algoritmos num ambiente web de fácil acesso, permitindo que o aluno construa a sua aprendizagem por conta própria e que o professor passe a ter um papel de orientador nesse processo. Com isso, a disciplina de ensino de

algoritmos pode contar com mais uma ferramenta de apoio para auxiliar na tarefa de ensinar algoritmos.

Espera-se que o ambiente possa ser utilizado nas aulas e que também possa contribuir para trabalhos futuros que agreguem novas funcionalidades ao mesmo.

### **6.3 PERSPECTIVA DE TRABALHOS FUTUROS**

O ambiente, desenvolvido nesse trabalho, focou-se na comunicação entre usuários e ao desenvolvimento de algoritmos. Existem aspectos importantes que podem ser explorados como a customização da interface pelo usuário, criação de assistentes virtuais ou ainda modificações no ambiente para que se possa não só atender disciplinas algoritmos e sim, outras disciplinas do curso de Ciência da Computação.

## 7 REFERÊNCIAS

- AMADEUS (2009) - Projeto Amadeus LMS. Projeto de gestão de aprendizagem. Disponível em <<http://amadeus.cin.ufpe.br/site/>>. Acessado em 26 de Junho de 2009
- AMBAP - Ambiente de Aprendizado de Programação. Disponível em <<http://www.ufal.br/tci/ambap/>>. Acessado em 26 de Junho de 2009.
- AMPLIA. Ambiente Multiagente Probabilístico Inteligente de Aprendizagem. Disponível em <<http://www.inf.ufrgs.br/~dflores/AMPLIA/index.html>>. Acessado em 26 de Junho de 2009.
- APLUSIX. Assistente para o aprendizado de álgebra. Disponível em: <<http://aplusix.imag.fr/>> Acessado em: 26 de Junho de 2009.
- BOVET, J. Visual Automata Simulator: a tool for simulating automata and turing machines. Universidade de San Francisco. 2004. Disponibilidade em <http://www.cs.usfca.edu/~jbovet/vas.html>. Acesso em Junho de 2009.
- BRUSILOVSKY, P., CALABRESE, E., HVORECKY, J., KOUCHNIRENKO, A. and MILLER, P. (1997), “Minilanguages: a way to learn programming principles”, International Journal of Education and Information Technologies, Vol. 2 No. 1, pp. 65-83.
- CASAS, Luis Alberto Alfaro CONTRIBUIÇÕES PARA A MODELAGEM DE UM AMBIENTE INTELIGENTE DE EDUCAÇÃO BASEADO EM REALIDADE VIRTUAL , UFSC, Florianópolis, 1999
- CHARNIAK, E.; MCDERMOTT , D. Introduction to artificial intelligence, Addison-Wesley, Reading, Mass., 1985
- COSTA, E. B. Um modelo de Ambiente Interativo de Ensino-Aprendizagem baseado numa Arquitetura Multi-Agentes. Campina Grande, 1997. Exame de Qualificação (Doutorado CPGEE), UFPA.
- CYBIS, W. De Abreu, ENGENHARIA DE USABILIDADE: UMA ABORDAGEM ERGONÔMICA, Laboratório de Utilizabilidade de Informática, 2003
- CYRINO, Eliana Goldfarb and TORALLES-PEREIRA, Maria Lúcia. Trabalhando com estratégias de ensino-aprendizado por descoberta na área da saúde: a problematização e a aprendizagem baseada em problemas. *Cad. Saúde Pública* [online]. 2004, v. 20, n. 3, pp. 780-788. ISSN 0102-311X.
- ELP, Environment for Learning to Program. Disponível em <<http://elp.fit.qut.edu.au/>>. Acessado em 26 de Junho de 2009.
- FUKS, H Aprendizagem e Trabalho Cooperativo no Ambiente AulaNet, Revista Brasileira de Informática na Educação. N6, Abril 2000, ISSN 1414-5685, Sociedade Brasileira de Computação, pp.57-73
- GIRAFFA, L.M.M.; NUNES, M. A.; VICCARI, R.M. *Multi-Ecological: an Intelligent Learning Environment using Multi-Agent architecture*. MASTA'97: Multi-Agent System: Theory and Applications. Proceedings... Coimbra: DE-Universidade de Coimbra, 1997.
- KOLIKANT, Y.B.-D. and BEN-ARI, M. (2008), “Fertile zones of cultural encounter in computer science education”, Journal of the Learning Sciences, Vol. 17 No. 1, pp. 1-32.
- KOMIS, V. (2001), “Didactics of informatics: from the formation of the scientific field to the

conjunction among research and school practice”, in Manolopoulos, Y. and Evripidou, S. (Eds), Proceedings of 8th Panhellenic Conference on Informatics with International Participation, Greek Computer Society, University of Cyprus, November 2001, pp. 463-71.

KOMIS, V. (2005), Introduction to Didactics of Informatics, Kleidarithmos Press, Athens (in Greek).

MALTEMPI, M.V.; VALENTE, J.A. (2000). Melhorando e Diversificando a Aprendizagem via Programação de Computadores. In: International Conference on Engineering and Computing Education (ICECE), São Paulo, SP. 27 a 30 de agosto. Anais em CD.

MCCARTHY, John; HAYES, P.J. "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in D. Michie (ed), Machine Intelligence 4, American Elsevier, New York, NY, 1969.

MOODLE - MODULAR OBJECT-ORIENTED DYNAMIC LEARNING ENVIRONMENT Disponível em <<http://moodle.org/>>. Acessado em 26 de Junho de 2009.

NARDI, B. (1993), A Small Matter of Programming: Perspectives on End – User Computing, MIT Press, Cambridge, MA.

NIELSEN, J. (25 Agosto 2003) Usability 101: Fundamentals and Definitions – What, Why, How. Jakob Nielsen's Alertbox. <http://www.useit.com/alertbox/20030825.html>

PREECE, J., ROGERS, Y., e SHARP, H. (2002) Interaction Design: beyond human-computer interaction. John Wiley & Sons, Inc.

RICH, Elaine; KNIGHT, Kevin: Inteligência Artificial. Makron Books. 2ª. Edição. São Paulo, 1994. 722p.

RINALDI, G , Educação Especial Deficiência Auditiva, Série Atualidades Pedagógicas, Secretaria de Educação Especial Brasília: SEESP, 1997, Disponível em [http://www.ines.gov.br/ines\\_livros/livro.html](http://www.ines.gov.br/ines_livros/livro.html)

ROSATELLI, M.C. Novas Tendências da Pesquisa em Inteligência Artificial na Educação. In: ESCOLA DE INFORMÁTICA DA SBC-SUL, 2000, Santa Maria:Anais...Santa Maria:UFSM, 2000.

SAJANIEMI, J. and KUITTINEN, M. (2005), “An experiment on using roles of variables in teaching introductory programming”, Computer Science Education, Vol. 15 No. 1, pp. 59-82.

SANTAROSA, L. M. C.; PASSERINO, L.; BASSO, L. de Oliveira; DIAS, C. de Oliveira Acessibilidade em Ambientes de Aprendizagem por Projetos I: construção de espaços virtuais para inclusão digital e social de PNEEs , Novas Tecnologias na Educação CINTED-UFRGS V. 5 N° 1, Julho, 2007

SILVEIRA, S. R. . Formação de Grupos Colaborativos num Ambiente Multiagente Interativo de Aprendizagem: um estudo de caso utilizando sistemas multiagentes e algoritmos genéticos. Porto Alegre: PPGC/UFRGS, 2006

SIOZOU, S ; TSELIOS, N. ; KOMIS, V. Effect of algorithms' multiple representations in the context of programming education, Interactive Technology and Smart Education Vol. 5 No. 4, 2008 pp. 230-243

SOLAR. Ambiente On-line de Aprendizagem. Disponível em <<http://www.virtual.ufc.br/solar/>>. Acessado em 26 de Junho de 2009.

TELEDUC. Ambiente de Educação à Distância. Disponível em < <http://www.teleduc.org.br/> >. Acessado em 26 de Junho de 2009.

TRUONG N., BANCROFT P. and ROE P. Learning to Program Through the Web, Faculty of Information Technology Queensland University of Technology GPO Box 2434, Brisbane QLD 4001, Australia , 2005

WEBBER, C., Boff, E., Bono, F. Ferramenta Especialista para Avaliação de Software Educacional. In: Anais do Simpósio Brasileiro de Informática na Educação (SBIE 2009), Florianópolis, SC, 2009.

WOOLF, (B. 1992. AI in Education. Encyclopedia of Artificial Intelligence, Shapiro, S., ed., John Wiley & Sons, Inc., New York, pp. 434-444.

WU, Helen Binghui. A framework for intelligente tutoring systems . 1995

ZAMBERLAM, Alexandre O.; GOULART, Rodrigo R. V. AS-MCOE: Tutor inteligente modelado em AgentSpeak(L), Grahl, E. A; Hübner, J. F. (Eds.). Anais do XV Seminário de Computação, Blumenau, 20-22 de Novembro, 2006. p 143-153

ARROSI, Tiago Proposta de Ferramenta de Auxílio à Aprendizagem de Algoritmos – Estudo baseado em Ambientes de Aprendizagem para Áreas Complexas. Trabalho de Conclusão de Curso (Graduação em Bacharelado em Ciências da Computação), Universidade de Caxias do Sul, 2009

## 8 ANEXOS

### 8.1 ANEXO A

**Tabela 11: Relação Utilizada pelo SIA-TP que relaciona Área de Conhecimento com o Conteúdo de aprendizagem**

Área de Conhecimento	Conteúdo de aprendizagem
1ª - Algoritmos sequenciais e condicionais	Declaração de variáveis; Uso de expressões; Uso de comandos condicionais (se, caso); Uso do comando de atribuição; Uso dos comandos de entrada e saída de dados.
2ª - Algoritmos com estruturas de repetição	Uso dos comandos de repetição; ○ Condição de parada no início (enquanto); ○ Condição de parada no final (repita); ○ Laço iterativo para; Controle de Loops; Condições de parada;
3ª - Algoritmos com vetores e matrizes	Declaração de vetores e matrizes; Uso de Índices de vetores e matrizes;

**Tabela 12: Classificação dos sinais de perigo em cada uma das três áreas de aprendizagem da disciplina de algoritmo de acordo com a área de conhecimento que o aluno se encontra.**

Área do Conhecimento do Aluno	Área 1		Área 2		Área 3	
	$< \alpha$	$\geq \alpha$	$< \alpha$	$\geq \alpha$	$< \alpha$	$\geq \alpha$
1	Pfr	Pfo	-----		-----	
2	NP	Pfr	Pfr	Pfo	-----	
3	NP	Pfr	NP	Pfr	Pfr	Pfo

#### Legenda

$\alpha$ , um valor para o limiar;

NP, uma situação onde não é identificado perigo;

Pfo, um sinal **forte** de perigo;

Pfr, um sinal **fraco** de perigo; e

Tracejado (-----), não se aplica.

## 8.2 ANEXO B



### Universidade de Caxias do Sul Centro de Computação e Tecnologia da Informação

#### Avaliação de Software Educativo

Este instrumento foi aplicado para avaliar o *Software Caderno de Exercícios* desenvolvido pelos alunos Tiago Arrozi e Matheus Zenato na disciplina de TCC do curso de Ciência da Computação.

Nome do Avaliador:

Formação:

I - Classificação do Software:

- Jogo educativo
- Simulador
- Exercício e prática
- Tutorial
- Hipermídia
- Sistema Tutor Inteligente
- Ambiente de aprendizagem interativo (micromundos)
- Sistema de Autoria
- Ambiente de aprendizagem cooperativo

II - Atividade pedagógica apresentada pelo software:

- Instrucionista/Comportamentalista
- Construtivista
- Sócio-interacionista

<b>ETAPA A - ASPECTOS PEDAGÓGICOS</b>	
01	O software propõe situações-problema que envolvam a formulação de hipóteses, a investigação e/ou a comparação? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
02	É adequado ao nível do aprendiz (público-alvo)? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
03	Instiga a procura de outras informações em diferentes fontes de pesquisa? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
04	Ele favorece a utilização interdisciplinar? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
05	O software apresenta atividades variadas, variando os níveis de complexidade das tarefas? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica

<b>ETAPA B - ASPECTOS TÉCNICOS</b>	
06	O usuário tem controle do software, podendo interromper as tarefas, retomá-las, corrigir erros? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
07	A apresentação de textos é adequada à leitura (tamanho da fonte, cor das fontes utilizadas)? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
08	As imagens auxiliam na compreensão dos conteúdos, sendo mencionadas nos textos explicativos? Elas não são apenas utilizadas de maneira decorativa? Por exemplo, ao tratar o assunto de funcionamento do motor, o software apresenta uma imagem de um tipo de motor. ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
09	Os grafismos e layout de tela são bonitos, estimulando a utilização do software? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica

10	O software dispõe de helps (ajuda) ou dicas? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
11	Recursos de animação são bem empregados (complementam o conteúdo, e não apenas distraem o usuário e decoram o sistema)? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
12	As instruções do software são claras, indicando de maneira precisa o que deve ser realizado? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
13	O programa opera de acordo com as instruções fornecidas na documentação? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
14	A linguagem apresentada é adequada? (Adequada ao público a que o software se destina, adequada ao tópico/conteúdo, características regionais) ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
15	As convenções e símbolos utilizados para guiar o usuário na interface são usuais? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
16	Quando um erro ocorre, o sistema consegue contorná-lo, permitindo que o usuário cancele a operação ou retorne a um ponto anterior? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
17	Utiliza recursos multimídia (textos, imagens, sons, filmes) para apresentar os assuntos? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
18	O vocabulário é adequado ao público-alvo? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
19	O conteúdo é apropriado ao nível do aluno? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
20	O software observa a correção da ortografia? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
21	O software apresenta manual do usuário? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
22	O software apresenta manual de instalação? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica

23	O software apresenta manual pedagógico ao professor? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica
24	O aluno consegue utilizar o software sem a interferência do professor? ( ) Sim ( ) Parcialmente ( ) Não ( ) Não se aplica

*Você tem comentários ou sugestões para o aprimoramento do software? Use o espaço abaixo.*