

# Suporte à Cooperação em um Ambiente de Aprendizagem para Programação (SambA)

Isaura Alcina Martins Nobre<sup>1,2</sup>, Crediné Silva de Menezes<sup>2</sup>

<sup>1</sup> Coordenadoria de Informática – CEFET/ES – UnED/Serra  
Rodovia Es 010, Km 6,5, Manguinhos – Serra – ES  
CEP 29160-000 – Tel 9963-5101  
imnobre@terra.com.br

<sup>2</sup> Mestrado em Informática - Centro Tecnológico – UFES  
Av. Fernando Ferrari, S/N, Campus Goiabeiras – Vitória –ES  
CEP 29060-970 – Tel (27) 3335-2654  
credine@inf.ufes.br

**Resumo.** Este trabalho apresenta *um ambiente para apoio a prática do ensino de Introdução a Programação*. Tem como objetivo auxiliar os alunos na solução de problemas, provendo interação e utilizando de estratégias de grupo para reforçar a colaboração e cooperação entre os participantes do curso. Foram definidas e modeladas várias ferramentas de auxílio a professores e alunos. Destacamos o *selecionador de soluções* que a partir de uma base de conhecimento e das soluções submetidas pelos alunos para uma determinada questão, deverá ser capaz de selecionar um conjunto de soluções de interesse para o professor. Essa seleção tem como objetivo ajudar o professor no reconhecimento das dificuldades encontradas pelos alunos ao resolver problemas possibilitando com isso uma realimentação do processo de aprendizagem mais eficaz.

**Palavras- Chave:** Internet e Educação, Ambientes Cooperativos baseados na Web, Introdução a Programação.

## 1. Introdução

O curso de Introdução a Programação é oferecido nos cursos voltados para a formação de profissionais de informática e também em cursos como: matemática, física, engenharia, entre outros. Estes cursos são ministrados de forma expositiva baseado na solução de problemas. O professor apresenta o conteúdo através do chamado *Portugol* (pseudo-linguagem), apresenta alguns exemplos e propõe exercícios para a turma. Destacamos algumas das dificuldades vivenciadas pelo professor no processo: reconhecer as habilidades inatas de seus alunos; apresentar técnicas para soluções de problemas; trabalhar a capacidade de abstração do aluno, tanto na busca das possíveis soluções como na escolha

da estrutura de dados a ser utilizada; promover a cooperação e colaboração entre os alunos. Isto parece ser um padrão em cursos voltados para formar pessoas na construção de artefatos complexos.

Na busca por contribuir para o equacionamento dessas dificuldades estamos desenvolvendo um projeto onde se busca a construção de ambientes telemáticos para apoio à realização de cursos orientados à resolução de problemas. Embora entenda-se que esses cursos guardam bastante similaridades, nosso foco de observação se ateve aos cursos introdutórios de programação.

Para o ambiente cooperativo definimos algumas ferramentas para auxiliar os alunos que privilegiam as estratégias de grupo no ambiente,

por exemplo, a viabilidade da comparação e análise das soluções obtidas, a possibilidade de ter uma base de casos para auxiliar na busca de uma solução específica, entre outras. Dentre as ferramentas voltadas para auxiliar o professor temos: um gerador de grupos automático, podendo este ser formado baseando-se nos perfis dos alunos, seleção de questões que atendam a um ponto determinado do andamento do curso, seleção de soluções que comparem as emitidas pelos alunos com os padrões de soluções que sejam de interesse do professor comentar, entre outras.

Neste artigo descrevemos os resultados obtidos no projeto. Na seção 2 apresenta-se o levantamento de ambientes voltados para o apoio ao ensino de Programação, na seção 3 descreve-se o ambiente Cooperativo de Aprendizagem (AmCorA), na seção 4 temos a proposta para o ambiente cooperativo (Samba), na seção 5 a modelagem do sistema e, finalmente, na seção 6 descreve-se o protótipo para apoio aos cursos de Introdução a Programação.

## **2. Ambientes para Apoio ao Ensino de Programação**

Nesta seção apresentamos alguns projetos relacionados ao ensino de Introdução a Programação, desatacando as principais contribuições para a melhoria do processo ensino aprendizagem de programação, contribuições estas que tomamos como base para concebermos nossa solução.

### **RoboProf e um curso de Introdução a Programação**

Segundo [Daly 1999], RoboProf é um sistema de ensino on-line utilizando o serviço WWW da Internet capaz de prover informação aos alunos, através de um conjunto de tópicos fechados e exercícios. Utilizado para dar suporte a um curso introdutório de programação, Roboprof pode ser efetivamente usado para ensinar a sintaxe e a semântica de uma linguagem, bem como a elaboração do projeto de um programa.

RoboProf vem favorecer a explanação da disciplina tópico a tópico apresentando vários exercícios com diferentes graus de dificuldade sendo um suporte ao ensino da sintaxe e

semântica da linguagem de programação. Além disso, este favorece um trabalho de dependência paralela, motivando o aluno com dificuldades para a aprendizagem.

### **AIDE: Uma Ferramenta Automatizada para o Ensino de um Curso de Introdução a Programação**

AIDE (Automated Interactive Design Editor) [Schweitzer 1989], é um sistema baseado em menu que permite aos estudantes entender um novo projeto, modificar um existente, ou automaticamente gerar a estrutura básica de um sistema. O formato do projeto é o mesmo gerado manualmente: descrição do problema, especificação da estrutura, descrição dos módulos, esqueleto do programa, porém todas as informações são editadas eletronicamente. Tendo um projeto sido criado e apropriadamente modificado, AIDE produz um programa Pascal compilável com blocos de comentários, cabeçalhos de procedimentos, declarações de parâmetros, pares BEGIN-END necessários, com endentação e formatação apropriada.

A ferramenta AIDE visa orientar as etapas de desenvolvimento de um software. Especificando a estrutura e descrevendo cada um dos módulos do sistema, o aluno observa a necessidade de cada etapa (a começar pelo entendimento do problema). Vale ressaltar a possibilidade de refinar a solução (alterando a descrição do projeto) e de através do ambiente poder gerar uma estrutura básica do programa em Pascal.

### **PORTUGOL/PLUS: Uma Ferramenta de Apoio ao Ensino de Lógica de programação**

Segundo [Esmin 1998], o ensino de lógica de programação é facilitado pelo uso do Portugol, pseudo-linguagem algorítmica utilizada para descrição das tarefas a serem executadas no computador visando a solução de problemas. Divide-se em duas partes principais: editor de algoritmos, que permite a digitação e a manipulação dos arquivos que contém os algoritmos utilizando uma interface dirigida por menus; um compilador para o ambiente Portugol/Plus capaz de verificar a sintaxe das instruções do algoritmo, caso não sejam encontrados erros, um programa objeto é gerado na linguagem de programação Pascal.

A utilização de uma ferramenta computacional para a prática do mesmo, ao invés do desenvolvimento dos algoritmos no papel, torna-se um referencial importante para o processo de ensino-aprendizagem.

### **Construção de Abstrações em Lógica de Programação**

Pela abordagem abstrata no ensino de lógica de programação [Mattos 2000] justifica o desenvolvimento de ferramentas e/ou aplicações tecnológicas que venham a auxiliar o processo de ensino-aprendizagem. Observa-se com frequência que a partir da solução de um determinado número de exercícios alguns alunos “descobrem” o formalismo para construção de soluções para lógica de programação. Já o grupo de alunos com dificuldades, permanece estagnado, e assume uma postura de tentativa-e-erro, levando-o ao desânimo e muitas vezes à reprovação na disciplina. O protótipo desenvolvido consiste em um sistema especialista capaz de reter o conhecimento acerca da solução de problemas de lógica de programação.

Sua contribuição está na apresentação da chamada metodologia dos 8 passos, ou seja, na definição de um método voltado para solução de problemas de lógica de programação – seqüência de perguntas que auxiliam os alunos na busca de uma solução para os problemas apresentados na disciplina, quebrando a primeira barreira para busca da solução que consiste em conseguir abstrair (entender o problema). Visando auxiliar os alunos em seu raciocínio (indução) o protótipo utiliza-se também do método de raciocínio baseado em casos (RBC).

### **Aprendizagem Baseada em Casos - um Ambiente para Ensino de Lógica de Programação**

[Koslosky 1999] propõe um modelo de ambiente de aprendizagem de lógica de programação utilizando a metodologia de raciocínio baseado em casos (RBC). A cada solução apresentada pelo aluno para um determinado problema, o algoritmo é gravado e o ambiente gera uma representação gráfica na tela.

A partir de um conjunto de projetos básicos e a avaliação dos processos lógicos desenvolvidos por um aluno, o ambiente atuando como

mediador deve sugerir os seguintes, mantendo ou aumentando a complexidade conforme os objetivos propostos. Este deve comparar os índices levantados pelo aluno na sua solução e os armazenados em cada caso da base para um mesmo tipo de problema, determinando sua similaridade.

Através do ambiente o aluno encontra facilidade para retornar e rever facilmente as etapas anteriores de seu raciocínio para solução de um problema, minimizando a ocorrência de erros cometidos anteriormente. Destaca-se também o conjunto de projetos com diferentes graus de dificuldades apoiando a busca sistemática de soluções para um problema e outros correlatos.

### **Um Ambiente Integrado para Solução de Problemas e Desenvolvimento de Programas**

Fadi [Fadi 1997] avalia a interdependência existente entre o processo de solução de problemas e o desenvolvimento de programas, objetivando desenvolver um ambiente integrado capaz de tornar o processo de solução de problemas e escrita de programas mais simples para estudantes iniciantes. A partir da análise de vários métodos de solução de problemas clássicos, Fadi sintetiza um método comum, incorporando aspectos essenciais dos métodos clássicos, o denominado modelo dual para solução de problemas e desenvolvimento de programas. Neste modelo baseia-se o desenvolvimento do ambiente integrado chamado SOLVEIT para apoio aos estudantes visando a solução de problemas e desenvolvimento de programas incluindo formulação de problema, planejamento, projeto, tradução, teste e resultados.

A grande contribuição está na explanação do processo de busca de uma solução por parte do aluno e as considerações levantadas por Fadi. Esta exposição reforça o fato de considerarmos relevante o auxílio aos alunos durante este processo de busca de uma solução e a necessidade de apresentar outras metodologias capazes de fazê-los analisar e criticar as soluções obtidas, buscando o refinamento da mesma e/ou outras soluções para o mesmo problema.

Estes trabalhos nos apontam aspectos relevantes a serem trabalhados em um curso introdutório de programação: capacidade de abstração

(entendi-mento do problema); decomposição do problema em subproblemas; planejamento e projeto de possíveis soluções; organização e refinamento da solução; especificação da solução lógica sem aprofundar em detalhes de implementação; e conhecimento de sintaxe, semântica e prática de uma linguagem de programação.

### **3. O Ambiente Cooperativo de Aprendizagem (AmCorA)**

O equacionamento dos problemas identificados requerem um ambiente com ferramentas apropriadas visando apoiar tanto professores como alunos no desenvolvimento de um curso baseado na solução de problemas. Para isto estamos desenvolvendo o ambiente SAmBA (Suporte à Cooperação em um Ambiente de Aprendizagem para Programação) através da implementação de ferramentas inteligentes apoiadas por um ambiente que favoreça a colaboração e a cooperação. Nesse sentido o SAmBA se constitui em uma facilidade adicional ao ambiente AmCorA.

O AmCorA (Ambiente Cooperativo de Aprendizagem) [Menezes e Cury 1999], em desenvolvimento no departamento de Informática da UFES (Universidade Federal do Espírito Santo), tem na sua concepção fornecer suporte inteligente a ambientes voltados para aprendizagem cooperativa.

Atuando como um provedor de grupos e subgrupos este possui várias ferramentas incorporadas ao ambiente facilitando o desenvolvimento de tarefas voltadas para educação. Dentre estas podemos destacar: o Moonline [Salles 1998], ferramenta de apoio ao esclarecimento, pode funcionar stand alone, na medida em que um estudante precisando obter esclarecimentos pode solicitá-lo ao seu grupo, ou ancorado em um documento; o Sábila [Goulart 2001], ferramenta de apoio à revisão bibliográfica, permite o destaque de trechos, a interligação entre textos, a inserção de comentários e a geração de resumos; o Timoneiro [Santos 2001], sistema composto de duas ferramentas, uma de autoria para apoio à construção de interfaces baseadas em mapas conceituais e de um navegador de mapas conceituais; e o Qsabe [Pessoa 1997], ambiente capaz de a partir de um serviço de perguntas e respostas, este é capaz de catalogar dinamicamente o perfil de especialistas do

conhecimento (colaboradores) e atuar como um distribuidor inteligente de mensagens. O projeto AmCorA incorpora também em sua especificação ferramentas de comunicação como: email e salas de bate-papo.

Além disso, as informações contidas no sistema são utilizadas para o desenvolvimento de cursos, sendo portanto controlada quanto ao acesso e utilização. Com isto, o sistema também oferecerá relatórios para acompanhamento do usuário e grupo de usuários. Informações como: número de acessos, quantidade de documentos armazenados, entre outras, devem estar disponíveis ao usuário e ao grupo em forma de relatórios.

#### **Proposta de um Ambiente Cooperativo para o Ensino de Programação**

A partir de um levantamento das metodologias utilizadas pelos professores e da classificação feita pelos alunos dos aspectos considerados necessários para o bom desenvolvimento de um curso, elaboramos a especificação de requisitos para o ambiente e, em seguida, detalhamos o projeto lógico através da modelagem do mesmo. Na sua concepção consideramos algumas estratégias de aprendizagem que privilegiam o trabalho individual, colaborativo e cooperativo. Dentre as estratégias de grupo destacamos: revisão por pares, distribuição das soluções obtidas pelos alunos para que outros não autores analisem; comparação de soluções, gerar soluções visando discutir e comparar, buscando em conjunto a melhor solução; e escolha e discussão de uma solução específica, que consiste em tecer comentários sobre uma solução específica. [Nobre e Menezes 2002]

Propomos um ambiente com suporte a ferramentas telemáticas de apoio ao processo de ensino-aprendizagem corrente, visando contornar as dificuldades encontradas por alunos e professores.

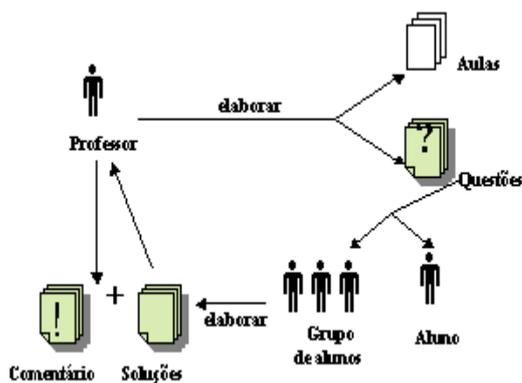


Figura 4.1. Visão das tarefas

Na Figura 4.1, temos a visão das tarefas dentro do ambiente: o professor responsável por elaborar aulas e questões, as questões sendo submetidas para os alunos ou grupo de alunos, os alunos elaborando uma solução e o professor visando identificar soluções para uma dada questão.

#### 4.1. Ambiente do Aluno

Descrição das tarefas voltadas para o aluno que poderão ser desempenhadas com o apoio do ambiente:

**Disponibilização de material bibliográfico:** O aluno poderá visualizar a aula correspondente ao conteúdo trabalhado no curso diretamente no ambiente. Deverão estar disponibilizadas também atividades relacionadas com a aula apresentada.

**Resolução de questões individuais ou em grupo:** Os exercícios disponibilizados poderão ser resolvidos individualmente ou em grupo e as soluções obtidas poderão ser inseridas no banco de Questões. O ambiente deverá permitir a reutilização, acesso e inclusão de novas soluções em repositórios de soluções de questões, possibilitando também armazenar comentários sobre as soluções obtidas.

**Comparação e análise de soluções:** A troca de informações entre os colegas da turma e, em particular, entre os membros de um grupo visando a análise e comparação de soluções deverá ser viabilizada. Esta análise e comparação servirá de suporte para elaboração de uma solução para a questão em conjunto entre todos os grupos, contemplando as principais características levantadas nas soluções encontradas.

**Comentários sobre soluções:** O ambiente deverá permitir a reutilização, acesso e inclusão de novos comentários em repositórios de soluções de questões. Esta ferramenta poderá auxiliar tanto a professores como alunos, favorecendo a troca de informações sobre uma determinada solução. O ambiente deverá favorecer o agrupamento dos comentários sobre a solução apresentada, de acordo com as principais características levantadas até que se obtenha um refinamento da mesma.

**Esclarecimento de dúvidas:** O aluno poderá ao tentar resolver um exercício, ou ao ler o conteúdo de uma aula específica se deparar com dúvidas sobre a aplicação de determinado conceito. O ambiente deverá dar suporte para que o aluno possa esclarecer suas dúvidas, permitindo que se estabeleça comunicação com outros colegas ou diretamente com o professor.

**Revisão de soluções por não autores:** O ambiente deverá permitir a troca entre os grupos das soluções encontradas para que grupos não autores possam analisar e discutir sobre a solução encontrada pelo outro grupo. Esta ferramenta estará favorecendo a utilização de estratégia de grupo, revisão por pares.

**Discussão de uma solução:** O ambiente deverá permitir que seja colocada para discussão uma solução inicial apresentada pelo professor ou por um aluno a uma questão, dando suporte a troca de informações para obtenção de conclusões sobre a solução. Os alunos ou grupos poderão através desta discussão chegar a elaboração de uma nova solução, ou seja, uma melhor solução para a questão.

#### 4.2. Ambiente do Professor

Descrição das tarefas voltadas para o professor que poderão ser desempenhadas com o apoio do ambiente:

**Elaboração de Aulas e Questões:** O ambiente dará suporte a construção de planos de unidades, planos de aulas, cadastro de aulas e suporte a elaboração de questões, cabendo ao professor determinar o grau de dificuldade da questão e a unidade a que esta se refere.

**Coordenação de atividades em grupo:** O professor deverá contar com recursos para apoiar a coordenação das atividades desenvolvidas por seus alunos. Dentre estes recursos temos o gerador de grupos que permitirá ao professor automatizar a criação de

grupos, segundo vários critérios, por exemplo, número de alunos ou perfil de aprendizagem do aluno.

**Seleção de questões:** O ambiente dará suporte a seleção de questões, através de consulta, para que o professor possa indicar como atividade referente a uma aula ou conteúdo.

**Seleção de soluções:** O ambiente dará suporte a seleção de soluções, para análise e comparação por parte do professor. Ao percorrer a sala de aula e observar o processo de solução dos alunos, a seleção de algumas soluções em detrimento de outras a serem apresentadas obedece alguns critérios como: solução comum, solução diferenciada das demais, outras soluções para o mesmo problema e solução que apresenta erro freqüentemente cometido pelos alunos.

### 4.3. Cenários de Uso

Este ambiente poderá ser utilizado em diversas situações, por exemplo:

**Situação 1:** o aluno poderá se utilizar do ambiente para através do serviço WWW para acessar conteúdo e exercícios, tornando-se assim, agente ativo no processo de ensino-aprendizagem. O ambiente estará favorecendo o ritmo individualizado de aprendizagem. O aluno com dificuldades na aprendizagem da disciplina poderá também estar se utilizando do ambiente como um trabalho de dependência paralela, já que o ambiente dará suporte ao esclarecimento de dúvidas. (não presencial)

**Situação 2:** o aluno poderá fazer uso do banco de soluções e comentários sobre soluções, favorecendo a sua análise crítica sobre a sua solução e as obtidas pelos colegas, auxiliando na compreensão e na descoberta de outras soluções para o mesmo problema, ou otimização das soluções propostas. (presencial ou não)

**Situação 3:** o professor poderá fazer uso do selecionador de questões ou do selecionador de soluções de acordo com o propósito de sua dinâmica de sala de aula, visando a organização e adequação das mesmas ao conteúdo ministrado. (presencial ou não)

**Situação 4:** favorecer as estratégias de grupo, a colaboração e a cooperação entre aluno-aluno e aluno-professor, no desenvolvimento de

soluções, comparação das mesmas, entre outras. (presencial ou não)

## 5. Modelagem do Sistema

Através da análise e projeto orientado a agentes identificando requisitos como: ter comportamento inteligente, ter conhecimento, interação com o ambiente e comunicação, obtivemos um modelo de agentes, cuja arquitetura podemos visualizar na Figura 5.1.

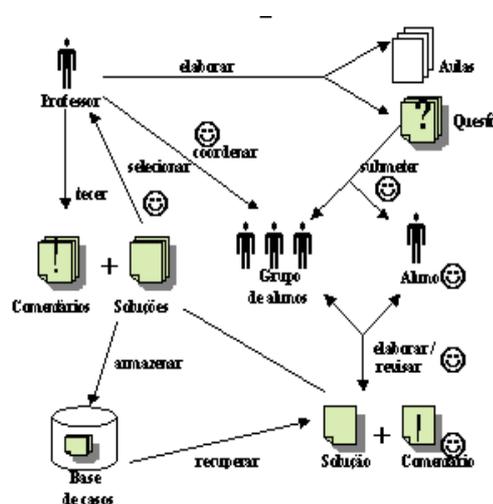


Figura 5.1. Visão do ambiente e os agentes

Os agentes identificados são:

a) **Agente Gerador de Grupos:** Agente deverá ser capaz de gerar grupos de alunos segundo critérios pré-estabelecidos pelo professor, através de uma linguagem para expressões. **Atributos:** Captura quesitos para formação do grupo e armazena numa base de conhecimento.

**Serviços:** Apresentar sugestão quanto a formação dos grupos de alunos para um determinado grupo de questões e/ou projeto a ser desenvolvido pelos alunos.

b) **Agente Ajustador de Perfil:** Tem como principal função traçar o perfil do aluno através de quesitos previamente estabelecidos pelo professor, oferecendo estes quesitos uma base de conhecimento para o agente sobre o aluno. **Atributos:** Captura quesitos e armazena numa base de conhecimento a medida que o aluno e/ou professor interage com o ambiente. **Serviços:** Orientar o professor e/ou o Agente Gerador de Grupos quanto ao perfil

dos alunos, visando a formação de grupos para solução de exercícios e/ou projeto.

- c) **Agente Seleccionador de Questões:** Visa selecionar dentre as informações sobre os exercícios catalogados aqueles que atendem ao objetivo de aprendizagem pretendido pelo professor. **Atributos:** Este agente infere através de alguns campos, se existem exercícios que possam ser de interesse para a turma, ou grupo específico de alunos de acordo com a abordagem do professor e desempenho das atividades das aulas associados a um banco de conhecimento. **Serviços:** Sugerir uma lista de exercícios.
- d) **Agente Seleccionador de Soluções:** Ser capaz de “ler” e “entender” uma solução com o auxílio de uma base de conhecimento composta por um conjunto de palavras-chaves e meta-informações sobre as soluções apresentadas.

**Atributos:** Este agente terá autonomia para selecionar soluções, através de um mecanismo de inferência buscando na base de dados informações que sejam de interesse do aluno, avaliando as variações sobre soluções apresentadas. **Serviços:** Sugere um conjunto de soluções para discussão por parte do professor.

- e) **Agente Revisão por pares:** Distribuir as soluções encontradas pelos alunos ou grupos de alunos para outros não autores analisarem e criticarem a solução apresentada. **Atributos:** De posse das soluções apresentadas para uma determinada questão este agente terá autonomia para distribuir estas soluções para alunos ou grupo de alunos não autores. **Serviços:** Levantar os pares para análise e crítica às soluções apresentadas a uma determinada questão.
- f) **Agente Agrupador de Comentários:** Capturar dentre os comentários aqueles mais consistentes. “ler” e “entender” um comentário com o auxílio de uma base de conhecimento composta por um conjunto de palavras-chaves e meta-informações. **Atributos:** A partir da análise e crítica de uma solução, poderão surgir vários comentários que precisam ser agrupados através de uma reestruturação. **Serviços:** Agrupar comentários, retirando redundâncias e ressaltando comentários reincidentes e validados.

Destacamos o papel do agente seccionador de soluções:

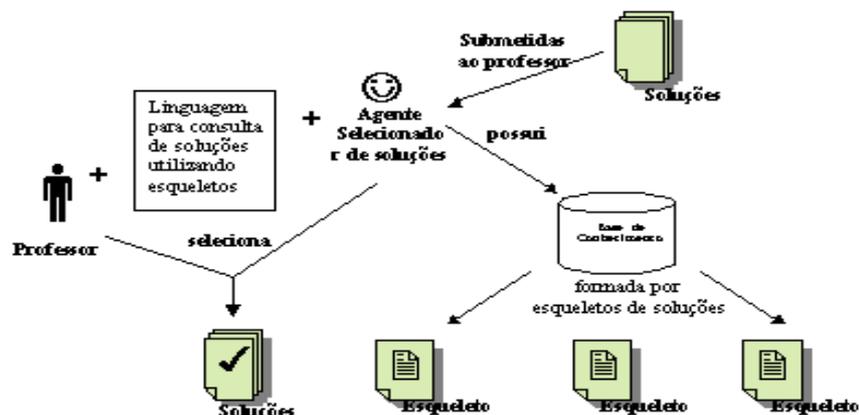


Figura 5.2. O seccionador de soluções

Na Figura 5.2, damos destaque ao agente Seleccionador de Soluções. Este agente utiliza e uma linguagem para descrição de consultas, que deverá ser utilizada para determinar o interesse do professor. Para cada problema proposto o professor deve associar um conjunto de padrões de solução que lhe interessa observar, formando assim a base de conhecimento. Os padrões serão guardados em forma de esquemas (palavras-

chaves e/ou meta-informações capazes de expressar o esqueleto da solução). Além disso, faz-se necessária a definição de uma linguagem de expressão voltada para a criação de esquemas, que possibilite ao agente analisar e comparar as soluções submetidas e ao professor dar manutenção na base de conhecimento, podendo também descrever esqueletos de seu interesse.

Para materializar a idéia do agente selecionador de soluções, definimos uma linguagem chamada LDE (Linguagem para Definição de Esquemas) a partir da linguagem Pascal, onde o professor poderá descrever os programas que pretende selecionar.

O professor poderá descrever seu interesse a partir do conjunto de soluções dos alunos, por exemplo:

#### ( 1 ) Verificar a ocorrência de uma variável do tipo inteiro e do comando if

Em LDE temos:

**var (integer)**

**if**

Comandos utilizados:

**var ( tipoS)**: Função responsável por verificar a ocorrência da declaração de uma variável de um tipoS qualquer. VAR (INTEGER)

**if** : Função responsável por verificar a ocorrência do comando if no programa.

#### ( 2 ) Verificar o aninhamento do comando if em else, ou seja, verificar se existe um comando if no bloco de comandos de else.

Em LDE temos:

**com ( else , if )**

Comando utilizado:

**com ( comando [ , comando ] )**: Função responsável por verificar o aninhamento de comandos, a ocorrência de um comando dentro de um bloco de comandos.

#### ( 3 ) Verificar a presença de uma variável do tipo inteiro na expressão lógica do comando IF.

Em LDE temos:

**expl (if, integer)**

Comando utilizado:

**expl ( comando , tipoS )**: Função responsável por verificar a presença de um identificador do tipoS na expressão lógica de comando.

### 6. SAmBA - Um Protótipo

Para materializar as idéias aqui apresentadas construímos um protótipo para o SAmBA, integrando-o como uma ferramenta para o trabalho em grupo do AmCorA. Dessa forma um curso que utilize SAmBA é modelado como um grupo do AmCorA. A título de ilustração criamos um grupo no AmCorA chamado IntProg, voltado para apoiar o ambiente SAmBA.

Para o grupo IntProg temos disponibilizados: cadastro de links e contatos de interesse para o curso; suporte a troca de mensagens (e-mail) entre os membros do grupo; uma área denominada estante (área pública de documentos) onde o professor pode disponibilizar aulas, exercícios e ferramentas para utilização dos alunos no curso, tais documentos podem ser organizados em pastas e subpastas; uma área denominada mesa redonda viabilizando a comunicação com os alunos através do esclarecimento de dúvidas, disponibilização de questões e desafios para os alunos, e/ou outros esclarecimentos quanto ao andamento do curso; escaninho (área privada de documentos) voltado para atender professores e monitores, área onde o professor pode estar recebendo os trabalhos desenvolvidos pelo alunos.

Na Figura 6.1, vemos que o ambiente oferece suporte para cadastro das unidades do curso de Introdução a Computação. Vemos que o ambiente oferece suporte para cadastro das aulas e manutenção das aulas objetivando a atualização de um dado curso. Neste protótipo será dada ênfase ao agente selecionador de soluções (Figura 6.2). Este agente visa identificar e selecionar de uma lista de soluções submetidas pelo aluno a uma determinada questão, aquelas que satisfaçam as características (sintáticas e semânticas) de interesse do professor. Através de uma linguagem, o professor poderá definir as características semânticas que espera encontrar nas soluções que lhe interessam ser comentadas.

Documento	Descrição	Data	Tam <sup>k</sup>
Ip.zip	Turbo Pascal - zipado	25/01/02 11:52:0	3524
winzip70.exe	winzip	07/02/02 11:53:3	922
Unidade1	Introdução	25/03/02 19:32:2	
Unidade2	Definição de Constantes e Variáveis	25/03/02 19:33:4	
Unidade3	Comandos Básicos	25/03/02 19:35:0	
Questoes	Exercícios para os alunos	27/03/02 11:09:3	

Figura 6.1. Cadastro das Unidades

Documento	Descrição	Data	Tam <sup>k</sup>
esquema1.txt		28/03/02 10:29:0	1
Prog1.pas		28/03/02 10:29:3	1
esquema2.txt		28/03/02 10:39:3	1
esquema3.txt		28/03/02 10:39:4	1
esquema4.txt		28/03/02 10:39:5	1

**Selecionador de soluções**

**Resultado da Seleção de Soluções**

Prog1.pas	<a href="#">Visualizar</a>	<a href="#">Enviar para área do grupo</a>
Prog2.pas	<a href="#">Visualizar</a>	<a href="#">Enviar para área do grupo</a>
Prog3.pas	<a href="#">Visualizar</a>	<a href="#">Enviar para área do grupo</a>

Figura 6.2. Selecionador de Soluções

Na Figura 6.3, vemos que o Turbo LDE foi construído com o auxílio das ferramentas TP lex e TP yacc, ferramentas utilizadas para desenvolver compiladores e programas similares como utilitários e interpretadores de linguagem de comandos com a linguagem de programação Turbo Pascal, construímos a gramática e a funcionalidade da linguagem LDE.

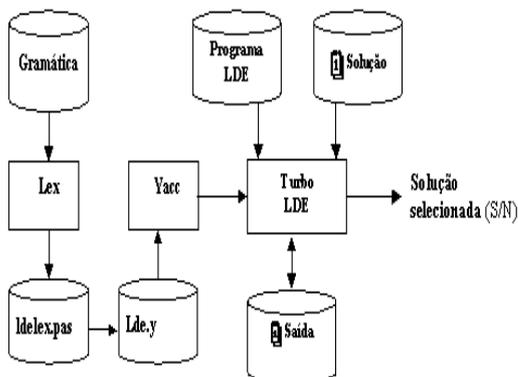


Figura 6.3. Estrutura do Turbo LDE

O Turbo LDE é um programa gerado para definir se dada uma solução de um aluno (programa PASCAL) esta atende ou não a estrutura especificada em um programa LDE.

Este utiliza um arquivo de saída para armazenar uma tradução feita pelo Turbo do programa do aluno em LDE, possibilitando pesquisar a estrutura do programa LDE submetido pelo professor dentro da estrutura do aluno.

## 7. Conclusão

Este trabalho nos permitiu a partir do levantamento da prática corrente do ensino de um curso de Introdução a Programação, propor um ambiente telemático de aprendizagem visando apoiar estes cursos introdutórios de programação.

Vários aspectos quanto a metodologia adotada pelo professor em sala de aula foram determinantes para que a proposta pude-se estar adequada a realidade da prática docente fundamentando o desenvolvimento de algumas ferramentas específicas para o professor. Além disso, nos aspectos levantados junto aos alunos, constatamos a relevância da prática da disciplina em laboratórios e a dificuldade encontrada para sanar dúvidas quando com atividades extra-classe. Isto reforçou a necessidade de propor um ambiente voltado para a colaboração e cooperação entre os participantes aluno-aluno e professor-aluno.

Como tecnologia de desenvolvimento temos o uso de agentes e a ênfase dada ao trabalho cooperativo. O trabalho vem propor várias ferramentas específicas para os ambientes alunos e professores. Destacamos a implementação do selecionador de soluções, ferramenta específica para o professor, que objetiva a seleção de soluções de interesse do professor. E o esforço no desenvolvimento de uma linguagem voltada para expressar esqueletos de solução.

Como indicação para trabalhos futuros, temos: a definição do perfil de um aluno de um curso baseado na solução de problemas; outro ponto a ser explorado ainda sobre agentes, é a comunicação entre o agente gerador de grupos e o ajustador de perfil, visando gerar grupos que melhor atendam as necessidades do professor; a animação do rastreamento de exemplos e soluções de questões, uma forma de ilustrar o que o professor faz em sala de aula; explorar o uso do RBC (raciocínio baseado em casos) para auxiliar o aluno no entendimento do problema; e traçar trilha de progresso vertical (unidade a unidade) e horizontal (dentro da unidade, aula a aula), visando um acompanhamento melhor por parte do professor.

## 8. Referências Bibliográficas

- [Daly 1999] Daly, Charlie, RoboProf and an Introductory Computer Programming Course, Department of Computer Applications, Dublin City University, ACM, 1999.
- [Deek 1997] F.P. Deek, An Integrated Environment for Problem Solving and Program Development, Ph.D. Dissertation, New Jersey Institute of Technology, 1997.
- [Esmim 1998] Esmim, Ahmed, PORTUGOL /PLUS: Uma Ferramenta de Apoio ao Ensino de Lógica de Programação Baseado no Português, IX congresso RIBIE – Congresso Ibero-americano de Informática na Educação, Brasília 1998.
- [Goulart 2001] Goulart, A. M. A., Pessoa, J. M., Menezes, C. S., Um Ambiente Cooperativo para Apoio à Revisão Bibliográfica, Simpósio Brasileiro de Informática Educativa, SBIE'2001, Vitória - ES, Brasil, 2001.
- [Koslosky 1999] Koslosky, Marco Antônio N., Aprendizagem Baseada em Casos em um Ambiente para Ensino de Lógica de Programação, Dissertação de Mestrado submetida ao Programa de Pós-graduação em Engenharia de Produção, UFSC, 1999.
- [Mattos 2000] Mattos, Mauro M., Construção de Abstrações em Lógica de Programação, VIII Workshop de Educação em Computação WEI 2000, Curitiba – PR.
- [Menezes e Cury 1999] Menezes, C; Cury, D. AmCorA: Um Ambiente Cooperativo para a Aprendizagem Construtivista Utilizando a Internet .SBIE, 1999. [www.gaia.ufes.br/amcora](http://www.gaia.ufes.br/amcora)
- [Nobre e Menezes 2002] Menezes, C., Nobre, I. Um Ambiente Cooperativo para Apoio a Cursos de Introdução a Programação Workshop de Educação em Computação WEI 2002, Florianópolis – SC.
- [Pessoa 1997] Pessoa, J.M. Desenvolvimento Orientado a Agentes: Uma Experiência com Agentes de Interface. Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática, UFES, 1997.
- [Salles 1998] Salles, T.; Monitoria OnLine – Um Ambiente para Apoio ao Atendimento Extra-Classe. Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática, UFES, 1998.
- [Santos 2001] Santos, M. C., Pessoa, J. M., Menezes, C. S., Aplicando Mapas Conceituais para Recuperar Informação em um Ambiente Virtual de Aprendizagem, Simpósio Brasileiro de Informática Educativa, SBIE'2001, Vitória - ES, Brasil, 2001.

[Schweitzer 1998] Schweitzer, Dino, Teel, Scott C., AIDE: An Automated Tool for Teaching Design in an Introductory Programming Course. Communications of ACM, 1998.

[Torrissen 1996] Torrissen, Christian B. Intelligent Agents and Conceptual Modelling, Norwegian University of Science and Technology, Trondheim Faculty of Physics, Informatics and Mathematics Department of Computer and Information Science, May, 1996.

[Wooldridge 1995] Wooldridge, M., Jennings, N. R., Intelligent Agents: Theory and Practice, 1995.