

# Modelando um ambiente de aprendizagem na *Web*: a importância da formalização do processo de desenvolvimento<sup>1</sup>

Sabrina Marczak<sup>2</sup>, Lucia Giraffa, Gláucio Almeida, Marcelo Blois

Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS  
Faculdade de Informática – Programa de Pós-Graduação em Ciência da Computação  
Caixa Postal 1429 – 90.619-900 – Porto Alegre – RS – Brasil

{smarczak, giraffa, ac107026, blois}@inf.pucrs.br

**Resumo.** Com o aumento da complexidade dos softwares educacionais em relação aos sistemas produzidos anos atrás, da diversidade de tecnologias adotadas e do grande número de pessoas envolvidas, tornou-se inadequado projetar um programa educacional sem utilizar-se um processo bem definido para orientar o seu desenvolvimento. Neste contexto, este artigo tem como objetivo descrever o processo de desenvolvimento do ambiente PROOGRAMA, desde sua fase de concepção, modelagem e parte de sua implementação, baseado no Microsoft Solutions Framework Process Model. O projeto foi concebido para oferecer um conjunto de recursos, denominados ferramentas virtuais, e um grupo de agentes para monitorar informações oriundas da interação entre alunos e professores usuários do ambiente.

**Palavras-chave:** Ambiente de apoio a aprendizagem, Processo de desenvolvimento de software, Sistemas multiagentes, Tecnologias *Web*.

**Abstract.** With the increase of the educational software complexity in relation to the systems produced years ago, the diversity of adopted technologies and the great number of involved people, it became inadequate to design and develop educational software without using a well-defined process to guide its development. In this context, the aim of this paper is to describe the PROOGRAMA development process supported by a Software Engineering methodology, the Microsoft Solutions Framework Process Model. The PROOGRAMA environment intends to support teaching/learning activities. It is based on a set of resources and functionalities that allow students and teacher to exchange information about results from evaluating process.

**Keywords:** Multi-agent systems, Software process development, Supporting learning environment, Web technologies.

## 1. Introdução

Com o aumento da complexidade dos softwares educacionais em relação aos sistemas produzidos anos atrás, da diversidade de tecnologias adotadas e do grande número de pessoas envolvidas, tornou-se inadequado projetar um programa educacional sem utilizar-se um processo bem definido para orientar o seu desenvolvimento. As equipes interdisciplinares integrantes dos projetos de software educacional, agora necessitam de especialistas de Engenharia de Software (ES). Estes contribuem para organização e definição de todos aspectos relacionados à produção do software. Existem três aspectos fundamentais

---

<sup>1</sup> Pesquisa suportada pelo Centro de Tecnologia XML Microsoft/PUCRS e por MEC/SeSu.

<sup>2</sup> Pesquisa também parcialmente financiada pelo Convênio Dell/PUCRS, através de recursos da Lei de Informática (Lei Federal Brasileira nº 8.248/91).

envolvidos na ES: *métodos*, que proporcionam os detalhes de “como fazer” para construir o software através da definição de um conjunto de tarefas; *ferramentas*, que proporcionam apoio automatizado ou semi-automatizado aos métodos; e *processos*, que constituem o elo de ligação que mantém juntos as ferramentas e os métodos (Pressman, 2001). O último aspecto é considerado o mais importante da ES.

Existem diversos Processos de Desenvolvimento de Software (PDS) disponíveis no mercado. Entre os mais conhecidos atualmente, pode-se citar o *Rational Unified Process* (Kruchten, 2001), o *Extreme Programming* (Beck, 2000), o *Agile Software Development* (Cockburn, 2002) e o *MSF Process Model* (Microsoft, 2002). Este último faz parte de um *framework* para projetos de tecnologia proposto pela Microsoft, denominado *Microsoft Solutions Framework* (MSF). O MSF é composto por mais um modelo, o de equipes (*Team Model*), que trata como estruturar pessoas e suas atividades; e três disciplinas: (a) *gerência de projeto*, que orienta a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto buscando atender as especificações do mesmo; (b) *gerência de riscos*, que trata o gerenciamento de riscos, os quais são inerentes a qualquer tipo de projeto; e (c) *aprendizagem*, que propõe uma forma de identificar conhecimentos e habilidades a serem usufruídas nos projetos, e de usar um projeto como uma oportunidade de aprendizado (Microsoft, 2002). O modelo de PDS da Microsoft foi criado em 1994, baseado nas melhores práticas (*best practices*) coletadas no desenvolvimento de produtos da empresa. O *MSF Process Model* estabelece uma ordem para a execução das atividades de um projeto de software, as quais estão organizadas em fases finalizadas por marcos (*milestones*). Este modelo combina os melhores princípios do ciclo de vida espiral com o tradicional modelo em cascata. Em relação a ambientes virtuais de ensino-aprendizagem de uma forma geral, o PDS exige um tipo de atenção especial. Segundo Andrade (2001), é preciso levar-se em consideração que este tipo de software pressupõe uma rede de articulações de estratégias e táticas pedagógicas, as quais são definidas a partir dos objetivos e pressupostos pedagógicos do professor que irá utilizá-lo. Neste sentido, é necessário primeiramente definir uma arquitetura pedagógica para o ambiente para posteriormente se partir para as decisões computacionais (definição da estrutura, escolha das tecnologias, etc.) e para o desenvolvimento do ambiente como um todo.

Este trabalho tem por objetivo descrever o processo de desenvolvimento do ambiente PROOGRAMA, desde sua fase de concepção, modelagem e parte de sua implementação, baseado no *MSF Process Model*. O projeto foi concebido para oferecer um conjunto de recursos, denominados ferramentas virtuais, e um grupo de agentes para monitorar informações oriundas da interação entre alunos e professores usuários do ambiente. Nesta fase, encontra-se especificado apenas o agente AMIGO (Agente para *MonItorar* e *Gerenciar infORmações*). Por utilizar-se a abordagem de Sistemas Multiagentes (SMAs), o ambiente prevê a inserção de outros agentes que, agregados ao sistema, ampliarão a atuação do AMIGO no que tange a gestão das informações do ambiente, visando auxiliar no processo de avaliação da aprendizagem dos alunos suportada pelo ambiente virtual. Este artigo está organizado em 6 seções. Na Seção 2 apresenta-se o ambiente PROOGRAMA e na Seção 3 descreve-se o seu processo de desenvolvimento. As considerações finais e os agradecimentos aos colaboradores e patrocinadores são apresentados, respectivamente, nas Seções 4 e 5. As referências bibliográficas mencionadas neste artigo são apresentadas na Seção 6.

## 2. O ambiente PROOGRAMA

O ambiente PROOGRAMA oferece diferentes áreas de trabalho, de acordo com o perfil do usuário. Foram definidos perfis para aluno, professor, monitor e administrador do ambiente. Tanto o **aluno** quanto o **professor** são usuários finais do ambiente. Mas o **professor** é o responsável pela organização de todo o material e informações relacionadas ao repositório de conteúdos, a agenda da disciplina, o quadro de avisos, o sistema de ajuda e o fórum de discussão. O **monitor** auxilia os alunos, de forma *on-line*, no esclarecimento de dúvidas. Para tal, o mesmo utiliza a ferramenta *Moonline* (apresentada nesta seção). O **administrador** é o responsável por questões operacionais, ou seja, tem como responsabilidade cuidar do funcionamento do ambiente. Suas principais atividades são: prover acesso aos usuários, gerenciar suas contas, e as ferramentas e funcionalidades a serem disponibilizadas. Para acessar suas respectivas áreas de trabalho, todos os usuários, devem utilizar seu número de matrícula (fornecida pela Universidade), uma senha de acesso, atribuída a cada um dos mesmos, e a disciplina e turma que estão vinculados. Estes dados são definidos diferentemente para o administrador.

O professor pode estruturar os conteúdos do curso (materiais, exercícios, descrições de tarefas, etc.) através da ferramenta **Plano de Aulas**. Os documentos e materiais relacionados à disciplina, bem como as bibliografias e os *links* indicados pelo professor são organizados através das ferramentas **Material de Apoio** e **Biblioteca**, respectivamente. Existe um **Glossário de Termos** criado a partir do resultado das pesquisas do significado de termos solicitada pelo professor aos alunos. A criação deste glossário pelos próprios alunos tem como objetivo incentivar a leitura e a escrita, bem como a prática da pesquisa. Tem-se ainda a disponibilização de **Atividades** (exercícios, trabalhos ou avaliações) aos alunos e o recebimento dos arquivos correspondentes, os quais podem ser acessados pelo professor através da seleção de uma atividade específica ou de um determinado alunos. O agente AMIGO é responsável por monitorar e gerenciar as informações relativas às atividades. Sendo assim, este gerencia a divulgação das atividades, os prazos de entrega, e habilita/desabilita a opção de recebimento das soluções conforme as datas estabelecidas. O professor pode acompanhar o *status* da tarefa (em andamento, entregue para correção) e solicitar ao AMIGO que gere relatórios sobre o andamento das mesmas. Posteriormente, o AMIGO ainda pode comunicar os alunos por *e-mail* que o prazo de entrega de um trabalho se aproxima ou que existe alguma atividade pendente.

Buscando auxiliar o aluno a identificar e entender os passos relacionados à elaboração de uma solução através de um algoritmo, tem-se o **AMBAP** – AMBiente de Aprendizado de Programação (Almeida, 2002) e o **ILA** – Interpretador de Linguagem Algorítmica (Evaristo, 2000). Estas ferramentas permitem o teste de algoritmos em um português estruturado. Para a comunicação dos alunos com o professor e os demais colegas, são disponibilizados três serviços de comunicação: o **E-mail**, o qual permite o envio tanto para participantes interno quanto externos ao ambiente; o **Chat**, o qual proporciona uma comunicação em tempo real entre os alunos que estiverem utilizando o ambiente naquele momento; e o **Fórum**, o qual permite discussões sobre assuntos definidos pelo professor. A **Agenda de Compromissos** e o **Mural de Avisos** permitem a comunicação de datas, atividades e eventos relacionados à disciplina, bem como a publicação de avisos, recados e novidades. O Mural de Avisos apresenta os últimos anúncios (avisos, recados, mensagens, etc.) elaborados pelo professor. Para auxiliar no suporte ao esclarecimento das dúvidas dos alunos, pretende-se utilizar a ferramenta **Moonline** (Monitoria *On-line*). Seu principal objetivo é permitir que os indivíduos pertencentes a um mesmo ambiente de interação virtual possam interagir entre si, a fim de esclarecer dúvidas (Gava, 2000). Para identificar a quem deve destinar as perguntas, a ferramenta Moonline utiliza os serviços da ferramenta, a **QSabe**. Esta ferramenta é capaz de identificar o conteúdo de uma pergunta e, através de técnicas de aprendizagem automática, aprender o perfil de cada colaborador/respondedor, automatizando o envio de perguntas e respostas (Menezes, 1998). O ambiente oferece a oportunidade de um aluno consultar informações sobre seus colegas, através das **Informações Pessoais**. Também é através desta funcionalidade que os alunos podem alterar seus dados pessoais (senha, *e-mail*, etc). O aluno ainda pode conferir os graus atribuídos a cada uma das suas avaliações, através do **Expositor de Notas**.

### 3. O processo de desenvolvimento do PROOGRAMA

Para orientar o desenvolvimento do PROOGRAMA, optou-se pelo PDS da Microsoft, o *MSF Process Model*. Esta escolha foi baseada no conhecimento prévio da equipe e na facilidade de acesso à documentação relacionada, uma vez que a descrição do modelo e os *templates* para elaboração dos documentos de projeto propostos estão disponíveis gratuitamente na Internet<sup>3</sup>.

#### 3.1. Visão (*Envisioning*)

Tendo colocado em prática sua metodologia de ensino por seis semestres, o que permitiu ao professor adquirir uma certa experiência com o uso de espaços virtuais como elementos de extensão e mediação das atividades da sala de aula presencial, este identificou a necessidade de ter um ambiente para integrar todos os recursos computacionais utilizados individualmente. Esta necessidade surgiu em função da demanda de tempo do professor ao organizar e gerenciar as informações geradas durante o trabalho realizado com as ferramentas computacionais adotadas. Decidiu-se, então, por desenvolver o ambiente PROOGRAMA, o qual tem como objetivo centralizar as ferramentas adotadas e gerenciar as informações geradas, proporcionando ao professor a redução do tempo de organização das mesmas.

---

<sup>3</sup> Para maiores informações, consulte o *site* do MSF (<http://www.microsoft.com/msf>).

Neste contexto, a primeira etapa durante esta fase disse respeito à identificação dos aspectos que iriam compor o ambiente, ou seja, definir o escopo do projeto. Esta análise foi baseada na metodologia utilizada pelo professor (Detalhes sobre esta metodologia de ensino podem ser encontrado em Giraffa, 2003.) e no estudo realizado sobre alguns ambientes virtuais de ensino-aprendizagem. Por existir uma cultura e experiência no grupo de Informática na Educação (IE) ao qual se está vinculado e por saber-se que oferecem a possibilidade de se explorar aspectos que vão ao encontro da proposta construtivista de trabalhar a produção e aquisição de conhecimento, escolheu-se os seguintes ambientes para serem estudados: AmCorA, AulaNet, LearningSpace e WebCT.

O *AmCorA* é um ambiente constituído de um conjunto de sistemas que pretende prover ao aluno condições para que este resolva problemas de forma cooperativa, interagindo com agentes humanos e artificiais (Menezes, 1999). O *AulaNet* baseia-se nas relações de trabalho cooperativo que se manifestam nas interações dos aprendizes com seus instrutores, com outros aprendizes e com os conteúdos didáticos (Fuks, 2000). O ambiente oferece ao docente uma gama de serviços de comunicação, coordenação e cooperação que podem ser usados no curso de forma a complementá-lo. O *LearningSpace* é um ambiente para gerência de cursos a distância que utiliza um sistema de *groupware* para gerência de informações distribuídas. Este sistema é baseado em uma arquitetura cliente/servidor, onde os usuários podem fazer acesso às bases de dados através de estações cliente ou ainda através de um *Web browser* (Lotus, 1999). O *WebCT* é um gerenciador de cursos a distância, onde cada curso possui um conjunto de páginas HTML (*HyperText Markup Language*), as quais provêem acesso às ferramentas de comunicação e colaboração configuradas pelo professor para serem utilizadas durante o curso. Pode, também, ser utilizado simplesmente como repositório de material de apoio para cursos presenciais (WebCT, 2000).

Após a análise destes ambientes, identificou-se um conjunto de características comuns aos mesmos, as quais foram agrupadas por afinidade de suas funcionalidades. Um resumo das principais características é apresentado na Tabela 1, a qual indica a presença ou a ausência de uma característica através das palavras “Sim” e “Não”, respectivamente. Para aquelas características que não dizem respeito a um determinado ambiente é apresentado “N/A” (Não se aplica) e para aquelas que não se conseguiu identificar sua presença no ambiente, tem-se o caractere “?”. Maiores detalhes sobre os ambientes e suas características podem ser obtidos em Marczak (2002).

**Tabela 1 - Características dos ambientes estudados**

CARACTERÍSTICAS		AMCORA	AULANET	LEARNINGSPEACE	WEBCT	
CARACTERÍSTICAS GERAIS	Usuário	Administrador	Sim	Sim	Sim	Sim
		Professor	Não	Sim	Sim	Sim
		Instrutor/Monitor	Não	Sim	Sim	Sim
		Aluno	Sim	Sim	Sim	Sim
	Controle de acesso e diferentes visões conforme o perfil		Sim	Sim	Sim	Sim
	Ajuda/Help on-line		Sim	Sim	Não	?
	Glossário de termos		Não	Não	Não	Sim
FERRAMENTAS DE COMUNICAÇÃO	Sincr.	Mensagens instantâneas	Sim	Sim	Não	Não
		Sala para bate-papo ( <i>chat</i> )	Sim	Sim	Sim	Sim
		Registro das salas de bate-papo	?	Sim	Sim	Sim
	Assinc.	E-mail interno ao curso	Não	Sim	Sim	Sim
		Envio de e-mail a usuários externos ao curso	Sim	Não	Sim	Não
		Recebimento de e-mail de usuários externos ao curso	Sim	Não	Não	Não
		Lista e Fórum de discussão	Sim	Sim	Sim	Sim
	Publicação de avisos		Sim	Sim	Não	Sim
	Agenda/Cronograma de atividades		Não	Sim	Sim	Sim
	Dados sobre os participantes		Sim	Não	Sim	Sim
FERRAMENTAS DE APOIO AO PROFESSOR	Elaboração de plano de aula		N/A	Sim	Sim	Sim
	Publicação de material de apoio		Sim	Sim	Sim	Sim
	Envio de tarefas aos alunos		N/A	Sim	?	?
	Identificação de novas tarefas a serem corrigidas		N/A	Não	Não	Não
	Manutenção de notas		N/A	Sim	Sim	Sim
	Geração de relatórios de participação		Não	Sim	?	Sim
FERRAMENTAS DE APOIO AO ALUNO	Agendamento de compromissos		Não	Não	Não	Sim
	Download de material		Sim	Sim	Sim	Sim
	Entrega de material por <i>upload</i>		N/A	Sim	Sim	Sim
	Consulta a resultados de tarefas		N/A	Sim	Sim	Sim
	Atribuição de <i>status</i> a uma tarefa		N/A	Não	Sim	Não
	Esclarecimento de dúvidas <i>on-line</i>		Sim	Não	Não	Não

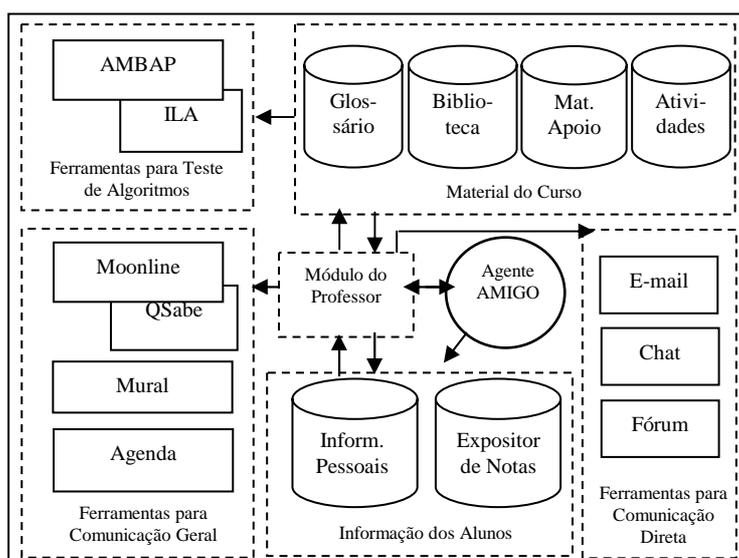
Identificou-se com a conclusão deste estudo e com a análise da metodologia do professor o escopo do projeto. Definiu-se que o ambiente deveria permitir a criação, organização e apresentação do cronograma da disciplina; criação e manutenção do acervo da mesma (banco de exercícios, banco de exemplos, biblioteca virtual, entre outros); a comunicação síncrona e assíncrona entre professor-aluno e aluno-aluno; a disponibilização de um sistema de ajuda *on-line*; a resolução de algoritmos em alto nível e teste dos mesmos; a organização/classificação dos materiais enviados pelos alunos; e o suporte ao processo de avaliação da disciplina. Além disto, definiu-se como premissa que o ambiente deveria suportar tanto atividades relacionadas ao professor quanto aos seus alunos e monitores.

Tendo-se definido o escopo do projeto, pode-se prever que haveria um determinado volume de trabalho e que, em função da restrição de tempo, havia o risco do cronograma de atividades não ser cumprido. Buscando evitar que este risco se tornasse real, decidiu-se pela estruturação de uma equipe de desenvolvimento. Selecionaram-se, então, dois bolsistas de Iniciação Científica (IC), um deles responsável pelo desenvolvimento e outro pela interface gráfica, realização de testes e apoio ao desenvolvimento da documentação do projeto, além de atender outras eventuais demandas, tais como auxiliar no desenvolvimento do *site* de divulgação do projeto. Outro risco identificado estava relacionado com as tecnologias que seriam adotadas, uma vez que se levou em consideração que o ambiente poderá ser utilizado fora do contexto da Universidade onde está sendo desenvolvido e é importante que funcione em diferentes sistemas operacionais. Para atender este aspecto, optou-se por utilizar a linguagem de desenvolvimento Java. Apesar da linguagem ser considerada portátil, não se tinha certeza se esta iria dar suporte a todas as necessidades identificadas. Havia o risco da linguagem não atender as expectativas e, desta forma, ter-se que optar por outra linguagem e investir tempo em seu estudo. Sendo assim, elaborou-se uma estratégia de estudo destes recursos para buscar garantir a opção tecnológica feita. Esta estratégia de viabilização das tecnologias foi colocada em prática na fase de Planejamento, conforme previsto pelo *MSF Process Model*.

### **3.2. Planejamento (*Planning*)**

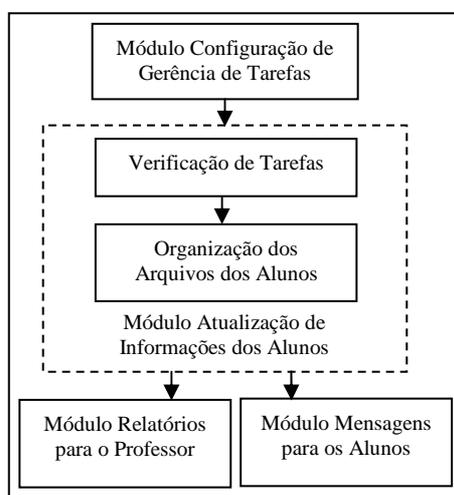
A primeira etapa desta fase constituiu-se em detalhar o escopo definido para o projeto e identificar os requisitos do sistema para poder-se propor uma solução. Estes requisitos são compostos por requisitos de negócio, de usuário e de sistema. Em relação ao primeiro, estes dizem respeito às necessidades pedagógicas e tecnológicas do professor, as quais já haviam sido identificadas com a análise de sua metodologia. Portanto, estes requisitos foram apenas revisados, uma vez que já haviam sido descritos. Quanto aos requisitos de usuário, estes dizem respeito aos aspectos não-funcionais do sistema, tais como questões de interface gráfica, aspectos de performance e segurança do ambiente. Pode-se citar uma das especificações feitas, a qual definiu que qualquer funcionalidade do sistema não deve ter uma resposta em um tempo superior a três segundos, pois este é o tempo médio que um usuário costuma aguardar uma resposta sem desviar sua atenção. Também, definiu-se que é necessário ter algum mecanismo que permita que as atividades do professor não sejam acessadas pelos alunos, ou seja, que haja restrições de acesso a determinadas funcionalidades conforme o perfil do usuário.

Encerrando a definição dos requisitos, detalharam-se os requisitos de negócio (necessidades pedagógicas e tecnológicas agregadas) e chegou-se aos requisitos do sistema, os quais estão sendo utilizados como base para a implementação do ambiente. Na verdade está se utilizando a abordagem do MSF para agregar o conhecimento oriundo de toda a experiência pedagógica formalizada através da metodologia descrita em Giraffa (2003). Seguindo a proposta de Sommerville (2003), elaborou-se uma versão preliminar da arquitetura do sistema (vide Figura 1), a qual identificou cinco diferentes módulos e definiu a presença de um agente de software, denominado AMIGO. A arquitetura interna do agente também foi definida neste momento.



**Figura 1 - Arquitetura do ambiente: visão modularizada**

Quanto à arquitetura do agente AMIGO (vide Figura 2), esta é composta de quatro módulos. O módulo *Configuração de Gerência de Tarefas*, que permite ao professor configurar o comportamento do agente, ou seja, definir quais informações este deve monitor e gerenciar. Por exemplo, o professor pode selecionar se o agente gerencia ou não a entrega de material dos estudantes. O módulo *Atualização de Informações dos Alunos* é composto por outros dois módulos: *Verificação de Tarefas* e *Organização dos Arquivos dos Alunos*. O primeiro módulo é responsável pelo controle do recebimento do material dos alunos. Para tal, o agente controla o calendário da disciplina e identifica os prazos de entrega definidos. O segundo módulo verifica os arquivos enviados pelos alunos, conferindo se o arquivo recebido é do mesmo tipo do solicitado (se possui a mesma extensão esperada) e se o arquivo não está vazio. O módulo *Atualização de Informações dos Alunos* também é responsável pela organização dos arquivos enviados pelos alunos quando da entrega de material relacionado a uma atividade. Isto ocorre baseado na configuração estabelecida pelo professor no módulo de *Configuração de Gerência de Tarefas*. O agente AMIGO cria, envia e gerencia relatórios para o professor, através do módulo *Relatórios para o Professor*. Estes relatórios têm como objetivo identificar o desempenho dos alunos durante a utilização do ambiente. Tem-se ainda o módulo *Mensagens para os Alunos*, responsável em organizar e comunicar aos estudantes informações relativas as atividades através de *e-mails*. Isto permite que o aluno receba informações sobre a disciplina mesmo que não esteja utilizando o ambiente PROOGRAMA.

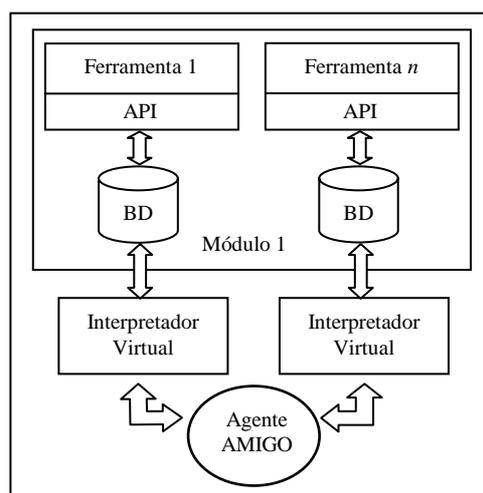


**Figura 2 - Arquitetura interna do agente AMIGO**

Em paralelo à etapa de conclusão da definição dos requisitos, parte da equipe estava colocando em prática o estudo planejado sobre tecnologias (linguagem de programação, ferramentas, técnicas, paradigma de desenvolvimento, entre outros). Foram estudadas diversas ferramentas que suportam o desenvolvimento em linguagem Java, bem como XML, uma vez que foi identificada a hipótese de adotar-se este padrão de desenvolvimento para atender questões envolvendo a troca de informações entre os módulos. Também se estudou característica geral de bancos de dados, como instalar e configurar servidores e metodologias de modelagem de SMAs, já que o monitoramento das informações do ambiente estariam sendo monitoradas por uma coleção de agentes de software. As vantagens e desvantagens de adotar cada uma das ferramentas estudadas foram descritas em um relatório técnico (Almeida, 2003), buscando divulgar aos interessados o estudo realizado. Sendo assim, optou-se por utilizar as seguintes ferramentas:

- *Rational Rose*: para modelar o ambiente (utilizando a UML – *Unified Modeling Language*);
- *FrontPage*: ferramenta de edição de HTML para editar a interface gráfica do ambiente;
- *Sun ONE Studio 4*: para desenvolver o código-fonte. Esta ferramenta está integrada ao componente Java Enterprise Edition (J2EE) versão 1.4, o qual permite a construção de softwares voltados para a Internet. Fornece, ainda, um servidor para disponibilizar as ferramentas J2EE;
- *MySQL Server*: para estruturar o banco de dados e armazenar os dados da aplicação.

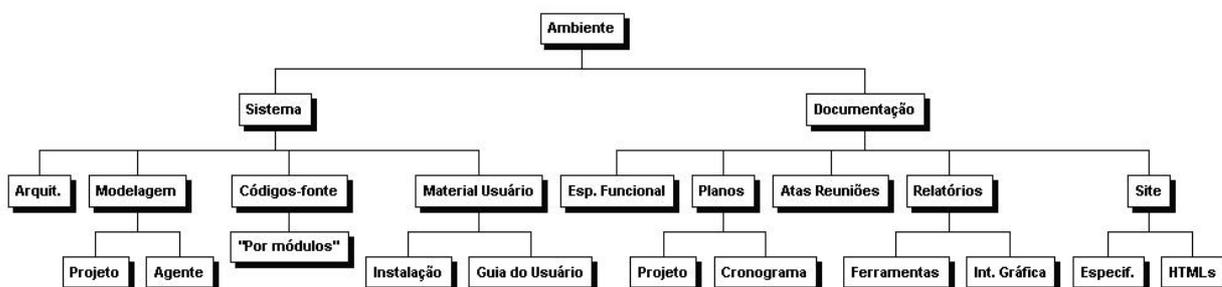
Cabe salientar que uma das razões que levaram tais ferramentas a serem adotadas foi a possibilidade destas serem disponibilizadas pela Faculdade de Informática da Universidade e poderem ser adquiridas sem custo algum. Este aspecto propicia que futuramente o ambiente possa ser disponibilizado aos demais parceiros de pesquisa sem que o fator custo inviabilize a sua utilização. A partir do estudo sobre as tecnologias elaborou-se uma versão mais detalhada da arquitetura do ambiente. Definiu-se que cada uma das ferramentas disponibilizadas nos módulos, denominadas ferramentas vertuais, deveria ser projetada de forma independente, visando facilitar a manutenção do ambiente e a padronização da comunicação do agente com os módulos identificados. Neste sentido, cada uma das ferramentas possui uma API para traduzir e codificar os dados que são trocados com o Banco de Dados (BD). O BD, por sua vez, comunica-se com um Interpretador Virtual, o qual funciona como um “interpretador” para a tomada de decisões do agente AMIGO. A Figura 3 apresenta uma visão simplificada do ambiente, uma vez que esta estrutura é idêntica para todas as ferramentas.



**Figura 3 - Arquitetura do ambiente: visão simplificada**

Uma segunda etapa desta fase tratou o planejamento das atividades do projeto. Identificou-se quais seriam os produtos a serem entregues ao final do projeto (vide esquema apresentado na Figura 4), e a partir desta lista identificou-se as atividades a serem realizadas. Após, estimou-se um prazo para realização de cada das mesmas e gerou-se o cronograma do projeto. Em seguida, definiu-se a forma de comunicação da equipe de desenvolvimento (*e-mail* com título padronizado, registro das decisões em atas

de reuniões semanais), a periodicidade de verificação da realização das tarefas e alguns padrões de desenvolvimento a serem seguidos, buscando facilitar a continuidade do projeto. Estas definições foram registradas em um plano geral do projeto, diferentemente do que propõe o *MSF Process Model*, em função de não se ter achado necessário redigir documentos distintos.



**Figura 4 – Produtos a serem produzidos pelo projeto**

Tendo-se identificado quais ferramentas seriam usadas, quais técnicas seriam adotadas e, uma vez planejado o desenvolvimento do projeto, partiu-se para o seu desenvolvimento.

### 3.3. Desenvolvimento (*Developing*)

Primeiramente, configurou-se o servidor onde o ambiente será disponibilizado. Este servidor foi configurado através da ferramenta Sun ONE Studio 4, mencionada anteriormente. Após, desenvolveu-se a interface gráfica do PROOGRAMA, usando HTML e JSP (*Java Server Pages*), uma tecnologia para a criação de conteúdo para a *Web* que permite a divulgação de componentes tanto estáticos quanto dinâmicos. Em seguida, partiu-se para o desenvolvimento das ferramentas em si e, posteriormente, do agente. Quanto à comunicação dos dados da interface do usuário com o BD, está se utilizando *servlets*, que são classes da linguagem de programação Java que estendem a capacidade dos servidores de aplicação acessados através de um modelo de programação do tipo “requisição-resposta”.

Até o momento foram desenvolvidas as ferramentas *Chat* e *E-mail*. Uma ferramenta de *Fórum*, desenvolvida em linguagem Java, foi cedida por um professor parceiro (vinculado a Faculdade de Informática desta Universidade), complementando o módulo *Comunicação Direta*. Além disto, também foram desenvolvidas as ferramentas *Informações Pessoais*, *Agenda* e *Atividades*, as quais fornecem subsídios para o funcionamento do agente AMIGO. Este se encontra em fase de finalização de sua implementação. Ainda, está-se estudando os códigos-fonte e uma forma de integração de duas ferramentas cedidas pelo grupo GAIA, a *Moonline* e o *Mural de Avisos*. Diferentemente do Fórum, estas ferramentas foram desenvolvidas em Delphi, fato este que pode inviabilizar a adoção das mesmas. As próximas atividades previstas envolvem o desenvolvimento do restante das ferramentas dos módulos *Informação dos Alunos*, *Comunicação Geral* e *Material do Curso*. Logo após, ter-se-á o estudo da viabilidade de integração do AMBAP e do ILA.

Ainda fará parte da fase de desenvolvimento redigir a documentação a ser disponibilizada ao usuário, que corresponde ao manual de instalação e configuração do ambiente e o guia do usuário. Também será elaborado o *help on-line* do ambiente. A última atividade desta fase corresponderá à configuração e preparação do ambiente de produção, ou seja, preparar o PROOGRAMA para ser disponibilizado ao professor em caráter experimental.

### 3.4. Estabilização (*Stabilizing*)

Foi planejado para esta fase um conjunto de testes para verificar se o software estará atendendo os requisitos especificados, especialmente os pedagógicos. Estes testes ocorrerão em dois momentos. Primeiramente, o professor irá usar o PROOGRAMA, em caráter experimental, buscando identificar se os requisitos pedagógicos e de usabilidade foram atendidos. Além disto, estes primeiros testes também terão a intenção de identificar se o manual de instalação e o guia do usuário foram bem redigidos. O segundo momento será realizado com a turma da disciplina de Algoritmos do professor, com os mesmos objetivos do teste com o professor. Os possíveis erros a serem identificados durante a realização destes testes serão

corrigidos e este processo de identificação e correção de erros se manterá até que se atenda todos os requisitos especificados para o ambiente.

### 3.5. Implantação (*Deploying*)

Ainda não foi planejado quando será disponibilizado o ambiente para ser adotado pelo professor como um recurso oficial da sua disciplina de Algoritmos, nem como se irá proceder para executar as eventuais sugestões apresentadas pelos alunos e professores que terão acesso ao ambiente. Provavelmente estas sugestões serão atendidas em novas versões do ambiente, as quais provavelmente irão incorporar outros requisitos, ainda não identificados neste momento. Isto virá ao encontro do objetivo deste projeto, o de desenvolver o ambiente de forma incremental.

## 4. Considerações finais

Não existe tradição na comunidade de IE de se documentar formalmente os softwares educacionais. Esta afirmação pode parecer um tanto contundente. Porém, ao se pensar na origem da área e dos seus projetos, encontra-se como responsáveis pela sua coordenação e gerência os parceiros da Ciência da Educação. Os softwares educacionais que se produzia há 15, 20 anos atrás eram de pequeno porte e a complexidade menor. Não se justificava a presença de um especialista em ES nas equipes interdisciplinares. Por sua vez, estes especialistas também não vislumbravam a área de IE como uma aplicação potencial para suas pesquisas e investigações. Hoje, este quadro mudou de forma significativa. Os ambientes estão cada vez mais complexos e de maior porte. Este crescimento é decorrente justamente da maturidade do trabalho interdisciplinar que a comunidade de IE atingiu através de muitas experiências oriundas das diversas parcerias que se estabeleceram nos projetos envolvendo a Ciência da Educação e a Ciência da Computação. O *design* de um ambiente educacional é resultado de uma escolha pedagógica. Ou seja, parte-se de uma proposta metodológica fundamentada nos pressupostos e crenças acerca do que se entende por aprendizagem, para então, buscar-se as tecnologias que viabilizarão o ambiente que se deseja construir. O que se consegue como resultado nos projetos é um protótipo ou no máximo um piloto do futuro sistema. Logo, evidencia-se a importância de seguir-se um PDS, especialmente quando se sabe *a priori* que o projeto será continuado. Não se encontra na literatura projetos de software da área educacional referenciando se um PDS foi seguido ou as vantagens advindas da adoção desta abordagem.

O grupo deste projeto vivenciou a dificuldade de interagir com outros pesquisadores na busca deste tipo de informação. Muitos dos grupos consultados não possuíam a documentação das estratégias elaboradas ou sequer sabiam explicar como a solução havia sido desenvolvida. Acredita-se que isto se deve ao fato de muitas pesquisas fazer parte de trabalhos de Conclusão de Curso ou IC. Geralmente estes trabalhos são desenvolvidos por um único aluno ou por pequenos grupos, o que não incentiva à prática da documentação. Neste sentido, buscou-se seguir um PDS para, além de ter-se uma orientação quanto às atividades a serem seguidas, também se ter uma documentação que relate as decisões do projeto e gere um histórico do mesmo. Acredita-se que isto irá auxiliar os eventuais estudantes que poderão vir a atuar nesta pesquisa. Após a conclusão da implementação, o PROOGRAMA será colocado à disposição aos alunos das disciplinas de Algoritmos e Programação de Computadores, em caráter experimental. A utilização do ambiente será monitorada buscando-se elaborar um estudo comparativo dos resultados de avaliação alcançados no rendimento desta turma piloto *versus* as turmas anteriores, as quais não trabalhavam com os recursos adotados em um ambiente centralizado. Para tal serão elaborados instrumentos específicos que permitam um estudo qualitativo e quantitativo, buscando-se indicadores que auxiliem a identificar o quanto o ambiente colaborou para a aprendizagem dos alunos. Como a aprendizagem é algo muito difícil de ser mensurado por um simples experimento e, os efeitos de uma metodologia só se manifestam a médio e longo prazo (quando o indivíduo realmente utilizar aquele conhecimento para resolver um problema), não se espera buscar elementos que forneçam indicadores de que o ambiente colaborou, de alguma forma, para auxiliar os alunos a criarem bons hábitos de estudo, autonomia e valorização do trabalho coletivo.

Dar-se-á continuidade a este projeto, buscando inserir novos agentes para auxiliar na definição de um modelo de aluno, analisar os arquivos enviados pelos usuários como resultado de suas atividades, agendar reuniões, entre outras funcionalidades. Buscando-se, com isto criar, tornar o PROOGRAMA um efetivo SMA. Novas funcionalidades e agentes serão definidas depois de avaliar-se o ambiente sob o

ponto de vista do seu desempenho. Uma vez que a inserção de novas funcionalidades e agentes deve ser justificada por uma melhoria associada a um ganho na exploração do potencial pedagógico do sistema.

## 5. Agradecimentos

Aos bolsistas de Iniciação Científica Gláucio Almeida (MEC/SeSu), e Luana Bernardino (FAPERGS), pelo apoio e auxílio prestados no contexto deste projeto. Agradecimentos especiais aos colegas da UFES e UNISINOS, e aos patrocinadores: Dell Inc. e Centro de Tecnologia XML Microsoft/PUCRS.

## 6. Referências bibliográficas

- Almeida, E. *et al*, *AMBAP: Um Ambiente de Apoio ao aprendizado de Programação*, In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 22, 2002, Florianópolis. **Anais...** Florianópolis: SBC, 2002.
- Almeida, G. *Um estudo sobre ferramentas para o desenvolvimento de um ambiente de ensino*, Relatório Técnico, PPGCC – Faculdade de Informática, PUCRS, Porto Alegre, 2003.
- Andrade, A. *et al*, *Requisitos para a modelagem de ambientes de aprendizagem a distância: Uma proposta da PUCRS Virtual*, In: INTERNATIONAL CONFERENCE ON NEW TECHNOLOGIES IN SCIENCE EDUCATION, 2001, Aveiro/Portugal, Disponível em [http://www.ead.pucrs.br/biblioteca/pesquisas\\_portal/index.php](http://www.ead.pucrs.br/biblioteca/pesquisas_portal/index.php), Consultado em Março de 2003.
- Beck, K., *Extreme Programming explained: Embrace change*, Boston, Addison-Wesley, 2000.
- Cockburn, A., *Agile Software Development*, Boston, Addison-Wesley, 2002.
- Evaristo, J, Crespo, S., *Aprendendo a Programar: Programando numa Linguagem Algorítmica Executável (ILA)*, Porto Alegre, Book Express, 2000.
- Fuks, H. *Aprendizagem e Trabalho Cooperativo no Ambiente AulaNet*. **Revista Brasileira de Informática na Educação**, [S.l.], n. 6, abril 2000, Disponível em <http://www.les.inf.puc-rio.br/groupware>, Consultado em Setembro de 2002.
- Gava, T., Menezes, C., *Moonline: Um Sistema Multiagentes Baseado na Web para Apoio a Aprendizagem*, In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 11, 2000, Maceió, AL. **Anais...** Maceió: UFAL, 2000.
- Giraffa, L; Marczak, S; Almeida, G., *O ensino de Algoritmos e Programação mediado por um ambiente na Web*, In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 23, 2003, Campinas. **Anais...** Campinas: SBC, 2003.
- Kruchten, P., *The Rational Unified Process: An introduction*, Boston, Addison-Wesley, 2001.
- Lotus Corporation, *Lotus Learning Space Forum: Student's Guide*, 3rd. ed, Cambridge: Lotus Press, 1999, Disponível em [http://www-12.lotus.com/ldd/doc/ufiles.nsf/docs/lsf36/\\$File/fstudent.pdf](http://www-12.lotus.com/ldd/doc/ufiles.nsf/docs/lsf36/$File/fstudent.pdf), Consultado em Outubro de 2002.
- Marczak, S., Giraffa, L., *A Gerência de Informação em Ambientes de Ensino a Distância: Um Estudo Comparativo*, Trabalho Individual II (Mestrado em Ciência da Computação) – PPGCC, Faculdade de Informática, PUCRS, Porto Alegre, 2002.
- Menezes, C; Cury, D. *AmCorA: Um Ambiente Cooperativo para a Aprendizagem Construtivista Utilizando a Internet*. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 10, 1999, Curitiba, PR **Anais...** Curitiba: UFPR, 1999.
- Menezes, C. *et al*, *QSave: Trocando Experiências sobre Informática Educativa em uma Rede de Educadores*, **Revista Brasileira de Informática na Educação**, [S.l.], n. 2, Abril 1998, Disponível em <http://www.inf.ufsc.br/sbc-ie/revista/nr2/Menezes02.htm>, Consultado em Janeiro de 2002.
- Microsoft. *MSF Process Model v. 3.1, White paper*, June 2002, Disponível em <http://www.microsoft.com/msf>, Consultado em Junho de 2003.
- Pressman, R., *Software Engineering: A Practioner's Approach*, New York, McGraw-Hill, 2001.
- Sommerville, I., *Engenharia de Software*, São Paulo, Addison Wesley, 2003.
- WebCT, *WebCT 3.0: Getting Started Tutorial*. 2000, Disponível em <http://www.webct.com>, Consultado em Setembro de 2002.