

---

## O uso de data warehousing no processo de aprendizagem de programação

Acompanhamento e avaliação de cursos de programação

**Daniel dos Santos Pêgas e Edgar Toshiro Yano**

Departamento de Computação - Instituto Tecnológico de Aeronáutica

Praça Marechal Eduardo Gomes, 50 - Vila das Acácias.

CEP 12228-900- São José dos Campos - SP – Brasil

{dspegas,yano}@ita.br

**Abstract.** *A trend in the learning process in general is the professor each time more to demand of the pupil the exploration capacity and creation of new knowledge and not simply memorization of the content of one disciplines. However this new boarding requires new instruments for accompaniment and evaluation of the learning process. A way to understand the process of programming learning is through data collected during the development of the project to carry through a process of Date warehousing on these data getting answers the questions that can direct the professors to redefine strategies to improve the performance of programming courses.*

**Resumo.** *Uma tendência no processo de aprendizagem em geral é o professor cada vez mais exigir do aluno a capacidade de exploração e criação de novos conhecimentos e não, simplesmente, memorização do conteúdo de uma disciplina. Entretanto, essa nova abordagem requer novos instrumentos para acompanhamento e avaliação do processo de aprendizagem. Uma maneira para entender o processo de aprendizagem de programação é, através de dados coletados durante o desenvolvimento do projeto, realizar um processo de Data warehousing sobre estes dados obtendo respostas a questões que podem direcionar os professores a redefinirem estratégias para melhorar o desempenho de cursos de programação.*

**Palavras-chave:** Educação à distância, programação, observação eletrônica, Data warehousing.

---

## 1. Introdução

Uma tendência no processo de aprendizagem em geral é o professor cada vez mais exigir do aluno a capacidade de exploração e criação de novos conhecimentos e não, simplesmente, memorização do conteúdo de uma disciplina. Entretanto, essa nova abordagem requer novos instrumentos para acompanhamento e avaliação do processo de aprendizagem.

Atualmente o processo de aprendizagem de programação requer que o aluno desenvolva, implemente e teste um conjunto de projetos para exercitar conceitos e técnicas vistas em aulas. Os projetos desenvolvidos, mesmo com um mesmo tema, apresentam características distintas, ficando difícil para o tutor fazer uma avaliação precisa do esforço despendido e aplicação correta de técnicas apresentadas em aula.

Uma maneira para entender o processo de aprendizagem é, através de dados coletados durante o desenvolvimento do projeto, realizar um processo de Data warehousing [Kimball 1998] sobre estes dados. Com um data warehousing poderemos obter respostas para várias questões, tais como: Como foi a evolução dos arquivos do projeto? Qual foi o tempo gasto com depuração? Quais trechos de programas foram mais avaliados pelo depurador? Quais foram os erros mais frequentes de compilação? Quais são os alunos que levam mais tempo para terminar um trabalho? Por que os alunos gastam tanto tempo para executar uma tarefa? Qual é o horário mais utilizado para a elaboração de projetos? O desempenho da turma desse ano foi melhor que a da turma do ano passado?

As respostas para essas questões poderão direcionar os professores a redefinirem estratégias para melhorar o desempenho de cursos de programação.

O artigo descreve um projeto de pesquisa em andamento que tem como propósito o desenvolvimento de um ambiente de ensino de programação que coleta, de forma transparente, dados sobre o desempenho de projetos de programação e a partir destas informações o professor tenha condição de, através de um processo de data warehousing, entender o processo de aprendizagem seguido pelos alunos durante o desenvolvimento de projetos de cursos de programação.

Descreveremos a seguir os seguintes itens: requisitos do sistema, descrição de cenários de uso, projeto de arquitetura de sistema, modelagem dimensional de data warehouse, data warehousing, vantagens e limitações do sistema e conclusões.

---

## 2. Requisitos do sistema

A primeira fase deste projeto de pesquisa foi o levantamento dos requisitos necessários, com ênfase no que deveria ser coletado no ambiente de desenvolvimento durante a realização da atividade pelo aluno [Pêgas e Yano 2004].

A partir dessas informações inicia-se o processo de data warehousing que deve ser capaz de:

1. Buscar ocorrências de padrões conhecidos;
2. Verificar a existência de relações entre os padrões encontrados;
3. Realizar comparações entre alunos, grupos e/ou turmas;
4. Apresentar o resultado obtido sob forma de relatórios.

Esta fase é conhecida como fase de exploração e transformação dos dados coletados [Kimball 1998]. A partir destes relatórios gerados é que o professor terá condições de redefinir estratégias que melhorem o desempenho de cursos de programação.

Os padrões mencionados são padrões que o módulo de dados irá entender como um padrão conhecido de como o aluno está se comportando ou aprendendo diante de uma atividade proposta de programação.

## 3. Cenários de uso

Com base nesses requisitos temos cenários que podem ilustrar melhor como o sistema pode colaborar para melhorar o processo de aprendizado de programação.

O professor configura o módulo de dados para um novo curso da linguagem C#. O professor registra os alunos matriculados, cria repositórios para os projetos e configura os coletores de dados a serem instalados em cada ambiente de desenvolvimento de aluno.

O aluno ao receber um projeto faz um esboço em papel, usando pseudo-código, dos algoritmos a serem utilizados. Para ter acesso ao ambiente de desenvolvimento ele ativa o módulo de aluno passando o nome de login e senha. Neste momento, coletores são instalados de forma transparente. Todos os arquivos fonte criados ou utilizados pelo aluno estarão em repositórios controlados pelo módulo de dados.

Para cada projeto passado para os alunos o professor obtém dados de desempenho individual ou coletivo, bem como todos os registros de informações que ele configurou para o processo de coleta de dados.

A partir dos dados coletados o professor poderá ter boa idéia da qualidade do código fonte em função do tempo que o aluno gastou para depurar um programa e estrutura dos arquivos desenvolvidos. Um analisador de código poderá verificar a presença de comentários, aplicação de princípios de programação modular, etc.

O professor terá a liberdade de fazer comparações entre alunos, grupos e turmas, tanto atuais como passadas.

Um supervisor de curso poderá analisar o andamento de um curso verificando quais projetos foram passados pelos professores, qual é o desempenho das diferentes classes ou o desempenho de um aluno ou grupo de alunos. Um supervisor poderá, por exemplo, avaliar o impacto do projeto de uma matéria sobre projetos de outras matérias.

---

## 4. Arquitetura do sistema

O sistema está dividido em três módulos conforme Figura 1.

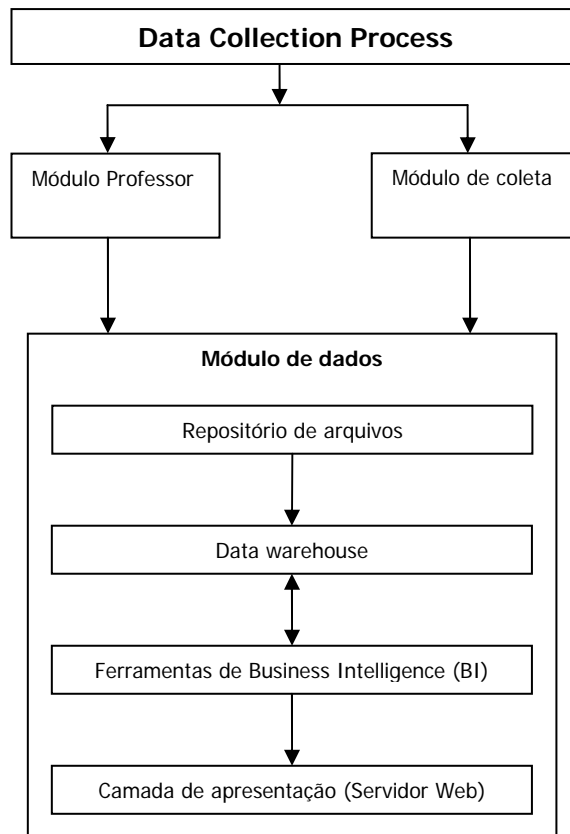


Figura 1: Arquitetura do sistema

O módulo de aluno consiste em um sistema instalado no ambiente de trabalho do aluno e de forma transparente, coleta e transmite ao módulo de dados, as informações sobre as atividades do aluno no ambiente de desenvolvimento. A única funcionalidade que é apresentada ao aluno é uma tela de identificação (login) para que o sistema identifique o aluno cujas informações estão sendo coletadas. O módulo de aluno é instalado como uma extensão do tipo plug-in em um ambiente de desenvolvimento como o MS-VisualStudio.net.

O módulo do Professor consiste em um sistema de administração, onde o professor irá configurar o processo de coleta, bem como ter acesso a relatórios referentes às informações coletadas de determinada atividade, de um aluno ou de uma turma. O módulo de professor é acessado através um navegador Web tal como IExplorer.

O módulo de dados é dividido em 4 camadas:

1. Repositório de arquivos: consiste em um ou mais repositórios de arquivos com acesso controlado;
2. Data warehouse: para armazenamento dos dados coletados;
3. Ferramentas de Business Intelligence [Kimball 1998]: conjunto de ferramentas para consultar e analisar os dados coletados e
4. Apresentação: consiste na camada onde há a apresentação das informações geradas pelas ferramentas de BI.

A comunicação entre os módulos de aluno e o módulo de dados será feita via Web Services, assim teremos a flexibilidade do aluno poder desenvolver a atividade em casa, de qualquer computador conectado à Internet, ou de um laboratório da própria instituição de ensino, via rede local.

A implementação do módulo de dados e de alunos está sendo feita através do uso das seguintes ferramentas: VisualStudio.net, MS-Sqlserver, Data Analyser e servidor web IIS. Entretanto, a arquitetura foi concebida de modo a permitir a criação de uma implementação com ferramentas de código aberto.

## 5. Modelo dimensional

O núcleo do sistema proposto é o módulo de armazenamento de dados que utiliza a tecnologia de data warehouse [Kimball 1998]. Para a elaboração do modelo de data warehouse foi feito um trabalho de modelagem dimensional, conforme representado na Figura 2.

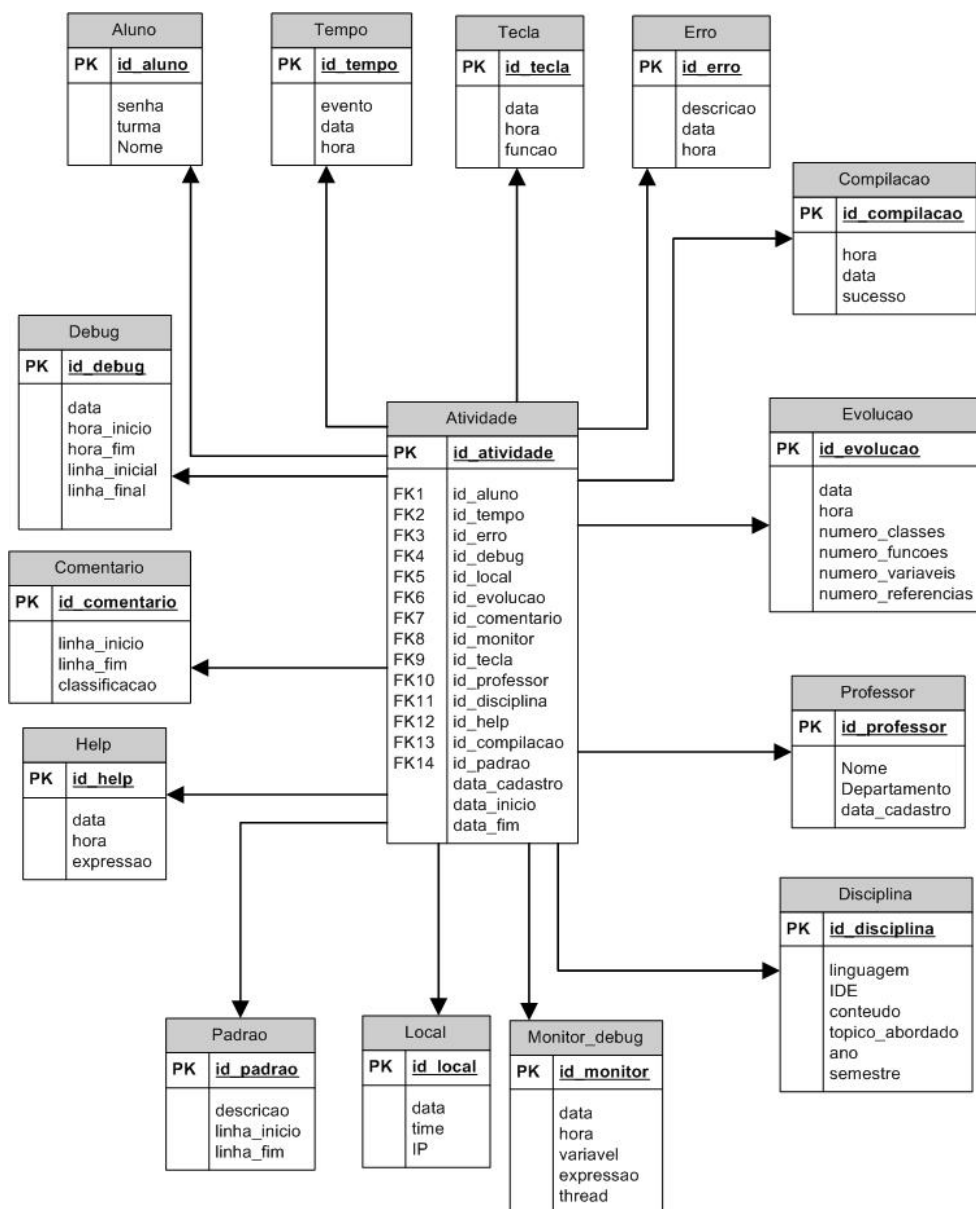


Figura 2: Modelo dimensional proposto para o sistema

---

Com este modelo conseguimos satisfazer todos os requisitos relacionados a armazenamento de dados para nosso sistema.

O uso de um Data warehouse se torna necessário, pois necessitamos combinar as necessidades de informações do professor com os dados coletados que serão armazenados no banco de dados.

Estas necessidades de informações que o professor precisa são transformadas em padrões conhecidos para o nosso sistema.

Um padrão conhecido pode ser de dois tipos:

1. Simples: indica a ocorrência de um único evento monitorado. Por exemplo: horário de início do desenvolvimento da atividade ou acusar a utilização do help durante o desenvolvimento.
2. Combinado: indica que dois ou mais eventos monitorados foram executados numa seqüência conhecida. Por exemplo: o início do uso do depurador implica em monitorar o horário de início e fim de uso bem como as linhas por onde o depurador passará.

Os padrões podem ser configurados pelo professor, que pode acrescentar ou retirar padrões que serão monitorados durante o desenvolvimento de determinada atividade. Atividades distintas, mas que ocorrem simultaneamente, podem ter configurações diferentes.

Passaremos a ilustrar alguns cenários resultantes do processo de Data warehousing.

## 6. Data Warehousing

Durante desenvolvimento da atividade o professor deseja saber apenas qual o horário mais utilizado pelos alunos para o desenvolvimento do projeto ou juntar a esta informação o local mais utilizado para desenvolvimento do projeto. Teríamos duas saídas como mostradas nas Figuras 3 e 4.

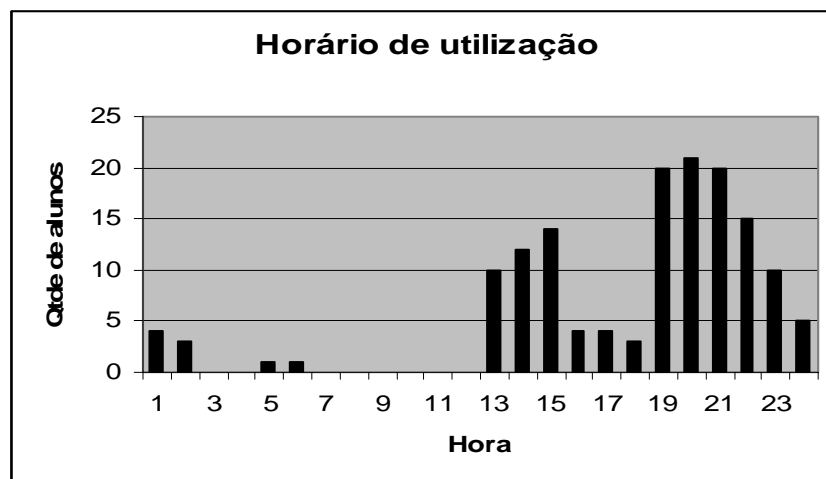


Figura 3. Gráfico que mostra os horários de desenvolvimento do projeto pelos alunos.

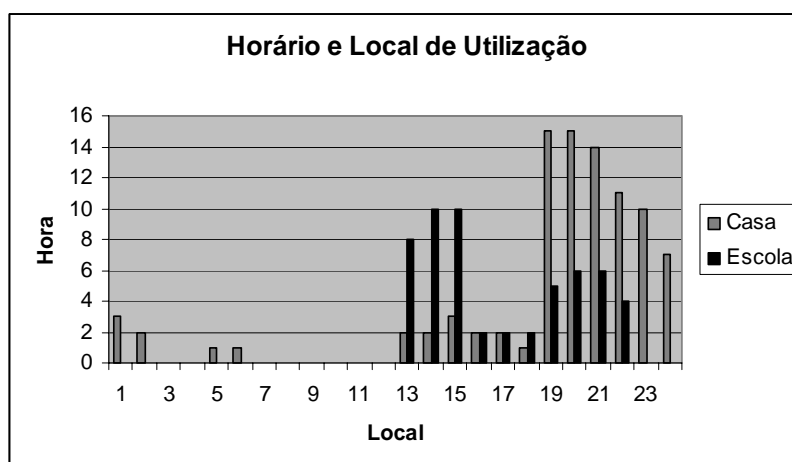


Figura 4. Gráfico que mostra os horários de desenvolvimento do projeto pelos alunos em relação ao local utilizado para o desenvolvimento.

Com base nestas informações o professor pode verificar se grupos de alunos estão reunindo-se para tentar desenvolver o projeto ou se os alunos estão tentando resolver isoladamente.

Outra informação importante que o professor pode ter acesso a qualquer momento é a evolução do arquivo fonte em função do tempo. Uma saída semelhante as já mostradas pode ser gerada. O gráfico poderia ser número de classes ou número de linhas de código ou ambas em função do tempo.

Podemos adicionar a estes dois cenários descritos acima a informação de evolução de ocorrência de erros. Teremos saídas como as Figuras 5 e 6 que ilustram o cenário ao longo de trinta dias do projeto de um aluno.

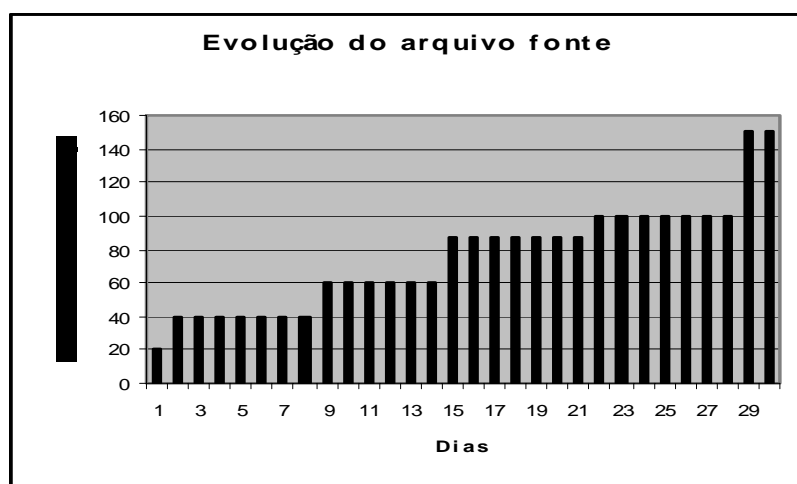


Figura 5. Evolução do arquivo fonte contando o número de linhas de código feitas pelo aluno.

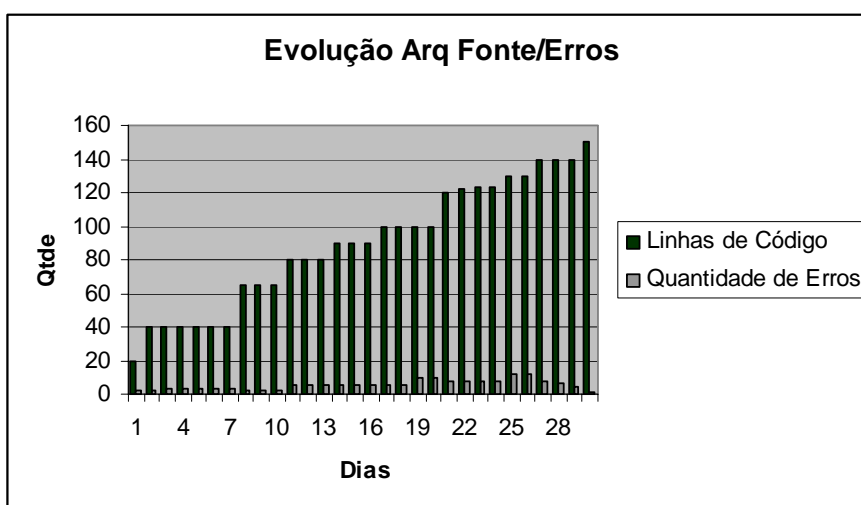


Figura 6. Evolução do arquivo fonte em relação à quantidade de erros apontados pelo compilador.

O professor através desta saída pode obter informações da dificuldade do aluno em lidar com muitas linhas de código, ou se ele teve uma evolução constante ou tentou resolver tudo de última hora. O professor pode ainda filtrar esta saída para apenas determinado erro que o compilador gera ou agregar um novo padrão, por exemplo, início do uso do depurador ou uso do help para verificar como o aluno esta tentando resolver o erro acusado.

Todos os padrões identificados, tanto os simples como os combinados, estão disponíveis para o professor poder mesclá-los da maneira que julgar conveniente.

Realizar comparações entre alunos agora se torna uma tarefa mais precisa. O professor terá condições de avaliar e gerar as mesmas saídas de um aluno para grupos ou turmas. A comparação entre atividades também é possível, visto que tudo está armazenado em um banco de dados à disposição do professor.

Com esta idéia de agregar padrões e obter saídas claras e precisas, o professor terá condições de responder a muitas questões sobre o aprendizado de linguagens de programação e assim redefinir estratégias que melhorem os cursos de programação.

## 7. Projetos similares

O departamento de computação da Open University possui em andamento um projeto de pesquisa denominado ASEOP, [Paine 1999], que consiste em coletar dados de um ambiente de aprendizagem (Smalltalk) de orientação a objeto para:

- Informar o educador qual foi o melhor efeito da aula;
- Prover um aparato para identificar problemas que estudantes podem experimentar enquanto aprendem;
- Prover evidências que melhorem o projeto do ambiente de programação.

Inicialmente o projeto contou com a coleta de informação de 30 alunos, em 2000 este número estava em torno de 300 estudantes.

Também na Open University há um outro enfoque que trabalha em conjunto com o projeto ASEOP que trata da gravação e análise do trabalho feito para entender as atividades práticas [Thomas 2002]. Como é um projeto em paralelo contou com os mesmos alunos do projeto ASEOP divididos em 8 grupos de atividades. Puderam ser constatadas diferenças significantes



---

entre a percepção do autor do curso e como os alunos se comportam realmente, principalmente do que diz respeito ao tempo de resolução das atividades.

A University of Glasgow possui um projeto de pesquisa (GRUMPS - Generic Remote Usage Measurement Production System) para desenvolver técnicas e software para automaticamente coletar, gerenciar e analisar ações do usuário, [Atkinson 2001], visando atingir as seguintes metas:

- Facilitar a organização de experiências que eficientemente coletam informações de usuários remotos;
- Facilitar a análise de repositórios que contenham estas informações a fim de testar hipóteses sobre as atividades dos usuários e facilitar o suporte a eles;
- Descobrir se esta abordagem pode melhorar a qualidade de sistemas de informação distribuídos e
- Descobrir se esta abordagem é válida para áreas de atuação específicas como educação, avaliação da usabilidade e ciência.

A Monash University está realizando um estudo para entender melhor o esforço do estudante na tarefa de desenvolver programas. Para isso estão desenvolvendo um plug-in para o Visual Studio .NET 2003 Academic para capturar conteúdo dos projetos, progresso de compilação e estatísticas de execução, [Redmond 2004].

A University of Hull também está realizando um estudo para entender os hábitos dos estudantes de programação. Para isto estão desenvolvendo um plug-in para o Visual Studio.Net 2003 Academic para capturar quantas vezes o aluno compila um projeto, para entender como o aluno trabalha nele e por quanto tempo, [Grey 2004].

Numa comparação com os sistemas citados, o nosso sistema de coleta e análise de dados de padrões de programação é o único que utiliza a tecnologia de data warehouse para o processo de armazenamento e análise de dados coletados.

## **8. Vantagens e limitações do sistema proposto.**

O sistema em desenvolvimento apresenta as seguintes vantagens:

- Transparente;
- Extensível;
- Arquitetura aberta;
- Permite o acompanhamento de projetos sendo executados em um laboratório ou em casa em qualquer hora.

Limitações da primeira versão:

- A análise de documentos se restringe ao código fonte dos programas.

## **9. Conclusões e comentários finais**

O aprendizado de linguagens de programação é uma tarefa ainda não muito compreendida e de difícil acompanhamento. O sistema em desenvolvimento utiliza, de forma integrada, as tecnologias da Internet, data warehouse e ambientes de desenvolvimento visuais para coletar e analisar dados obtidos do processo de aprendizagem de programação.

A arquitetura utilizada para o sistema pode ser facilmente adaptada para outros processos de aprendizagem. Como extensão futura desse projeto está a construção de um sistema de monitoramento de uso de objetos de aprendizagem (learning objects). A criação e uso de objetos de aprendizagem apresentam características similares ao processo de

---

aprendizagem de linguagens de programação. Assim como um professor de programação deseja ter métricas sobre a evolução de arquivos de programas, um autor de um objeto de aprendizagem deseja também ter métricas para a avaliação da efetividade do uso de um determinado objeto e acompanhamento do aprendizado adquirido pelo aluno.

A coleta e análise de métricas de aprendizagem é sem dúvida um campo a ser explorado. O uso adequado das tecnologias providas pela Internet e banco de dados relacionais pode oferecer respostas para muitas questões ainda não resolvidas sobre processos de aprendizagem.

## 10. Referências Bibliográficas

KIMBALL, Ralph. *Data Warehouse Toolkit*. São Paulo: Makron Books, 1998

KIMBALL, Ralph et al. *The Data Warehouse Lifecycle Toolkit: experts methods for designing, developing and deploying data warehouses*. New York: Wiley, 1998.

KIMBALL, Ralph; MERZ, Richard. *Data Warehouse: construindo o Data Warehouse para a WEB*. Rio de Janeiro: Campus, 2000.

THOMAS, Pete; PAINE, Carina. *Monitoring distance education students' practical programming activities*. 2002. Disponível em: <[http://ifets.ieee.org/periodical/vol\\_3\\_2002/thomas.html](http://ifets.ieee.org/periodical/vol_3_2002/thomas.html)>. Acesso em: 17 abril 2004

PAINE, Carina; THOMAS, Pete. *AESOP-An Electronic Student Observatory Project*. 1999. Disponível em: <<http://mcs.open.ac.uk/aesop/>>. Acesso em: 17 abril 2004.

ATKINSON, Malcolm et al. *GRUMPS - Generic Remote Usage Measurement Production System*. 2001. Disponível em : <<http://grumps.dcs.gla.ac.uk/>>. Acesso em: 18 abril 2004

GREY, David J. *Student Profiling & Course Management in Assignment Manager*. 2004. Disponível em : <<http://www2.dcs.hull.ac.uk/people/cssdjg/Projects/AM/>>. Acesso em: 05 abril 2004.

REDMOND, Wash. *Microsoft Selects Five Universities To Enhance Visual Studio .NET 2003 Academic*. 2003. Disponível em : <<http://www.microsoft.com/presspass/press/2003/nov03/11-04UniversitiesVSTOPR.asp>>. Acesso em: 05 abril 2004.

PUC, *Data Mining - Conceitos, Técnicas, Ferramentas e Aplicações*. 2004. Disponível em : <<http://www.cce.puc-rio.br/informatica/dataminingcentro.htm>>. Acesso em: 15 de junho 2004.

Pêgas, Daniel; Yano, Edgar. *Coleta e análise de métricas do processo de aprendizagem de programação*. Aceito no Congresso da Sociedade Brasileira de Computação (SBC'04), Salvador (BA), Agosto, 2004.